SLAM-aware, Self-Supervised Perception in Mobile Robots

by

Sudeep Pillai

B.S.E in Mechanical Engineering
University of Michigan (2008)

S.M. in Electrical Engineering and Computer Science
Massachusetts Institute of Technology (2014)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in
Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2017

Author . . . . . .
Department of Electrical Engineering and Computer Science
August 31, 2017

Certified by . . . . . .
John J. Leonard
Samuel C. Collins Professor of Mechanical and Ocean Engineering
Thesis Supervisor

Accepted by . . . . . .
Leslie Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Committee for Graduate Students

To my loved ones

SLAM-aware, Self-Supervised Perception
in Mobile Robots

by

Sudeep Pillai

Submitted to the Department of Electrical Engineering and Computer Science
on August 31, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in
Electrical Engineering and Computer Science

Abstract

Simultaneous Localization and Mapping (SLAM) is a fundamental capability in mobile robots, and has been typically considered in the context of aiding mapping and navigation tasks. In this thesis, we advocate for the use of SLAM as a supervisory signal to further the perceptual capabilities in robots. Through the concept of *SLAM-supported object recognition*, we develop the ability for robots equipped with a single camera to be able to leverage their SLAM-awareness (via Monocular Visual-SLAM) to better inform object recognition within its immediate environment. Additionally, by maintaining a spatially-cognizant view of the world, we find our SLAM-aware approach to be particularly amenable to few-shot object learning. We show that a SLAM-aware, few-shot object learning strategy can be especially advantageous to mobile robots, and is able to learn object detectors from a reduced set of training examples.

Implicit to realizing modern visual-SLAM systems is its choice of map representation. It is imperative that the map representation is crucially utilized by multiple components in the robot's decision-making stack, while it is constantly optimized as more measurements are available. Motivated by the need for a unified map representation in vision-based mapping, navigation and planning, we develop an *iterative and high-performance mesh-reconstruction algorithm* for stereo imagery. We envision that in the future, these tunable mesh representations can potentially enable robots to quickly reconstruct their immediate surroundings while being able to directly plan in them and maneuver at high-speeds.

While most visual-SLAM front-ends explicitly encode application-specific constraints for accurate and robust operation, we advocate for an automated solution to developing these systems. By bootstrapping the robot's ability to perform GPS-aided SLAM, we develop a *self-supervised visual-SLAM front-end* capable of performing visual ego-motion, and vision-based loop-closure recognition in mobile robots. We propose a novel, generative model solution that it is able to predict ego-motion estimates from optical flow, while also allowing for the prediction of induced scene

flow conditioned on the ego-motion. Following a similar bootstrapped learning strategy, we explore the ability to self-supervise place recognition in mobile robots and cast it as a metric learning problem, with a GPS-aided SLAM solution providing the relevant supervision. Furthermore, we show that the newly learned embedding can be particularly powerful in discriminating visual scene instances from each other for the purpose of loop-closure detection. We envision that such self-supervised solutions to vision-based task learning will have far-reaching implications in several domains, especially facilitating life-long learning in autonomous systems.

Thesis Supervisor: John J. Leonard
Title: Samuel C. Collins Professor of Mechanical and Ocean Engineering

Acknowledgments

First and foremost, I would like to thank my advisor, John Leonard. It has been an absolute pleasure working with him, and I am extremely appreciative of the freedom and open-ended research that he has allowed me to pursue over the years. I strongly believe that his long-term vision for life-long autonomy has considerably impacted the motivation for this thesis.

I would like to thank Nick Roy, for all the insightful and intellectual discussions we have had about my research vision, and am extremely grateful to have had the opportunity to work with him. Research discussions with him have always been thought-provoking, and I am thankful that he was on my thesis committee. I am extremely grateful to Leslie Kaelbling and Antonio Torralba, who were very grounded and constructive in their feedback. I really appreciate the general guidance and valuable input that has helped shaped this thesis. I am also thankful for the Machine Learning (6.867) and Advances in Computer Vision (6.869) courses they taught. Both these classes got me excited about their impact in mobile robots, and has ever since been part of my growing research toolkit. I would also like to thank Srikumar Ramalingam, who was my mentor at Mitsubishi Electric Research Labs. He has kept me motivated about computer vision and geometric problems throughout my term as a graduate student, and I hope that we can continue to work on interesting problems for the years to come. I would also like to acknowledge the late Seth Teller. I would not be here today writing this thesis if it weren't for both Seth and John. In a lot of instances, before I brainstormed with John about research problems and ideas, I would constantly ask myself "What would Seth say?". Having him in spirits has made me think twice about my ideas and solutions. I constantly find myself at awe that I have had such insightful and knowledgeable mentors to guide me throughout the years. Their excitement and attitude towards research continue to inspire me.

I would like to especially thank my colleagues and friends in the Marine Robotics Group, Dehann, Pedro, Liam, Ross, Julian, Peter, David Rosen, Francesco and Mei. It has been wonderful working amidst you all. Special thanks to Dehann, Pedro, David and Liam who engaged in insightful and lengthy discussions about each others' work to keep our research interesting and relevant. I would also like to thank my colleagues in the Robotics, Vision and Sensor Networks Group. Big thanks to Matt Walter, Sachi Hemachandra, Mike Fleder, David Hayden, William Li, Jon Brookshire, Nick Wang and many others who made those first few transition years easier, and got me excited about robotics research.

I would like to thank my friends here at MIT; being surrounded by their compassion and laughter has made my experience here at MIT all the more enjoyable. Abhishek, Shreya, Kat, Carl, Oscar, and many others, it has been a wonderful journey here in Cambridge, and I owe it you all for making this journey a memorable one. I would also like to thank Area 4, Darwin and Flour for their cappuccinos; this thesis would not have been possible without their valuable contribution.

I would like to especially thank my parents for the unconditional love and endless support they have provided throughout my life. I cannot imagine being here today if it weren't for them, and I am forever indebted to their love, support and sacrifice. My father, who kept believing in me that I could pursue whatever I wanted as long as I put my heart into it. His courage to pursue his own dreams has driven me to this very day. His attitude towards life keeps inspiring me, and I hope I continue to learn from him for the many years to come. My mother, being a primary school teacher, continues to see me as her student and had words of advice for me especially before my thesis defense. Her unconditional love and support for me and my well-being throughout these student years is something that I will always remember and cherish; To both of you *Amma* and *Acha*, I want to thank you from the bottom of my heart for everything you have done for me. My parents will forever be my first and most influential mentor(s).

I want to thank my brothers, and their wives for being a constant source of inspiration, and support. My brothers have always mentored me throughout the years. I especially remember that so-claimed "watershed" moment, where they convinced me to pursue my education in the US. If the fact that I am mentioning it here isn't telling enough, I want to take this opportunity to thank you both for being there for me; for having my back, for pushing me to my limits, for standing up for me, for challenging me, for believing in me. I have grown to the person I am because of the two of you. To my nephews and niece, you bring a huge smile to my face every time I think of you; I couldn't ask for more.

Finally, I would like to thank my fiancé, Sruthi. She has been extraordinarily supportive and loving throughout my unforgiving graduate student life, and I am extremely grateful to her for this. Sruthi, this thesis would not have been possible without your support — and as you are here sitting beside me, as I write this final section, I want to say that I am excited to spend the rest of my life with a wonderful person like you. Thank you, again.

Funding

This work was partially supported by the Office of Naval Research under grants N00014-11-1-0688 and N00014-16-1-2628 and by the National Science Foundation under grant IIS-1318392

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomous mobile robots have enjoyed wide attention recently, particularly in the form of self-driving cars, unmanned aerial drones and autonomous underwater vehicles. One of the reasons for their wide scale adoption has been attributed to the success of powerful computer-vision algorithms that are able to reliably build a geometric, and semantic description of the world it perceives.

While geometric and semantic scene understanding are two core competencies that can mutually benefit each other, they have been predominantly treated as two independent problems for the past few decades. Geometric scene understanding in mobile robots, more popularly referred to as *Simultaneous Localization and Mapping* (SLAM) (Bailey and Durrant-Whyte 2006; Durrant-Whyte and Bailey 2006; Thrun and Leonard 2008), is a long-studied, fundamental capability that equip robots to simultaneously build a *geometric representation* of its environment and localize itself within that representation. While most autonomous systems could benefit from the full-SLAM solution, most systems however only perform either localization or mapping, assuming the other to be known or pre-specified. This is primarily due to the challenges in developing long-term and robust SLAM solutions, both from an algorithmic complexity and semantic data-association standpoint. Nevertheless, we are interested in studying the limits of semantic scene understanding in mobile robots that are *spatially-cognizant* of its immediate environment.

Semantic scene understanding algorithms in the computer vision literature are evolving at an unprecedented pace today. Challenging datasets in object recognition, semantic segmentation, pose estimation from a few years back are being solved with considerably stronger accuracy year after year. This trend has predominantly been attributed to recent developments in Convolutional Neural Net-

works (CNNs), more generally Deep Neural Networks, and their implementation on modern computing architectures such as GPGPUs. Their success have especially been remarkable with continued performance gains year-over-year, consuming ever-increasing datasets readily available on the internet for training purposes. These methods have considerably altered the landscape of computer-vision today, bringing contextual and semantically-rich scene understanding to seemingly daunting perceptual tasks.

Spatially-cognizant Scene Understanding This thesis focuses on endowing robots with spatially-cognizant, otherwise referred to in this thesis as *SLAM-aware*, perceptual models for improved scene understanding. More specifically, we investigate the task of object recognition in mobile robots while simultaneously maintaining a strong geometric understanding of its environment via a full visual-SLAM solution. This further leads us to consider the converse problem, "Can rich spatio-temporally consistent semantic cues extracted from images, provide sufficiently reliable measurements to improve SLAM?". We investigate this further in the context of vision-based localization, where the task of identifying previously visited scenes is recovered purely by extracting and matching relevant semantic cues from sequential imagery captured on a mobile robot.

Following the need for strong geometric understanding in mobile robots, we reconsider the map representation problem in autonomous mobile systems. Most systems today typically convert the map reconstructions provided by the SLAM component into more convenient, intermediate representations to afford tasks such as motion planning, obstacle avoidance or high-level decision making. We motivate the need for a unified representation for vision-based mapping in mobile robots, that is potentially amenable for joint inference and control in a resource-aware manner.

Life-long Learning With the unreasonable effectiveness of data in the deep-learning era (Sun et al. 2017), most state-of-the-art solutions to semantic scene understanding from images require large amounts of training data and ever-increasing training periods. Furthermore, amassing large amounts of labeled data for task-specific solutions becomes increasingly tedious and expensive. Robots, on the other hand, collect a rich set of cross-modal information across their various sensor streams. This brings us to yet another interesting question: "What if we can take advantage of the natural time-based synchronization to learn capabilities in one sensor by transferring that knowledge from another domain or sensor?". If this were feasible, then we would be required to collect data in an unsupervised manner, and be

able to bootstrap task knowledge using a known, calibrated sensor and transfer the task capability onto a newer un-calibrated sensor.

Strongly rooted to this thesis is the fundamental ability in robots to perform vision-based SLAM. While model-based visual-SLAM algorithms have enabled significant advances in mobile robot navigation, however, they are limited in their ability to learn from new experiences and adapt to newer environments. We envision robots to be able to learn from their previous experiences and continuously tune their internal model representations in order to achieve improved task-performance and model efficiency. We investigate the concept of self-supervised learning of visual SLAM front-ends in mobile robots, by bootstrapping an existing GPS-aided SLAM solution as a supervisory signal.

1.1 Thesis Objective

The objective of this thesis is to extend the perceptual capabilities in autonomous robots by making them *spatially-cognizant* of their environment via Simultaneous Localization and Mapping (SLAM). This thesis addresses how mobile robots could potentially leverage their inherent SLAM capabilities to better inform tasks such as object recognition, and bootstrap the learning of vision-based localization tasks.

- As localization and mapping techniques become more powerful, we envision that the spatial-awareness that SLAM solutions yield can be used effectively for various robot-specific tasks. These methods can be particularly useful, as they act as *correspondence-engines*, providing strong spatial data association even across the robot's long-term operation. By viewing SLAM *as a sensor*, we consider the problem of semantic scene understanding in robots and investigate the benefits of developing a spatially-cognizant recognition system. Conversely, as scene-semantics are spatially grounded within physical environments that robots observe, we consider the potential of richer semantic scene understanding and the role it may have in bolstering SLAM systems of tomorrow.

- As robots leverage their SLAM capabilities in other tasks, it is critical to re-evaluate the underlying map representation that is maintained for the purpose of vision-based navigation. While many mapping systems are tailored towards constructing high-fidelity maps, their utility in other sub-tasks such

as motion planning and obstacle avoidance is questionable. We seek a *unified and flexible map representation* that can be directly estimated in SLAM subcomponents, while being readily usable in the context of motion planning in mobile robots with little modification. Furthermore, we expect this common representation to be especially amenable to planning feedback so that these systems can perform in a resource-constrained and plan-aware setting.

- Autonomous systems today are typically configured with a growing set of sensors that make calibration, monitoring and model maintenance of these independent sensors especially tedious. With the rich set of cross-modal information that these sensors typically collect, we envision robots to be able to *self-supervise* themselves in certain tasks by transferring or bootstrapping these capabilities that leverage other complementary sensors. Additionally, we show that this bootstrap mechanism can also take advantage of the representations that are implicitly maintained in robots such as localization and mapping (SLAM). In the future, we expect robots to able to build redundancy in tasks using sensors that are newly introduced by bootstrapping existing knowledge from previously modeled sensors.

We describe our contributions to these objectives in the following sections.

1.2 Contributions

- **Monocular SLAM-Supported Object Recognition** In **Chapter 3**, we develop novel and powerful models for simultaneous semantic and geometric scene understanding that considerably improve upon existing state-of-the-art techniques. We introduce the notion of "*SLAM-aware*" object recognition (Pillai and Leonard 2015) — a robot, spatially cognizant of its environment and location, can outperform traditional frame-by-frame detection and recognition techniques, by incorporating its knowledge of the object from various viewpoints. Additionally, by maintaining a spatially-cognizant view of the world, we find our SLAM-aware approach to be particularly amenable to few-shot object learning. We show that a SLAM-aware, few-shot object learning strategy can be especially advantageous to mobile robots, and is able to learn object detectors from a reduced set of training examples.

- **Map Representations for Vision-Based Navigation** Inherent to the spatial and semantic understanding is the choice of a map representation. We moti-

vate the need for a unified representation for vision-based mapping and planning, and introduce a mesh-based stereo reconstruction algorithm that has compelling properties geared towards mobile robots. In **Chapter 4**, we develop a *High-performance and Tunable Stereo Reconstruction* algorithm that enable robots to quickly reconstruct their immediate surroundings and thereby maneuver at high-speeds (Pillai et al. 2016). Our key contribution is an iterative refinement step that approximates and refines the scene reconstruction via a piece-wise planar mesh representation, while being dynamically tunable in order to enable resource-aware computation in fast-maneuvering vehicles. Furthermore, we emphasize that our approach can also be readily extended to incorporate navigation plans for context-aware and robust obstacle avoidance.

- **Self-Supervised Ego-motion Learning in Robots** Fundamental to any autonomous system is its ability to leverage its multitude of sensors in order infer, and act in its immediate environment. We address the concern of growing sensor modalities in autonomous systems today, and the need for a general-purpose framework that enables self-supervision in vision-based navigation tasks such as visual ego-motion estimation and loop-closure identification. In **Chapter 5**, we propose a self-supervised solution to visual ego-motion estimation for varied camera optics, including *pinhole, fisheye, and catadioptric lenses* (Pillai and Leonard 2017a). We develop a generative model for optical flow prediction that can be utilized to perform outlier-rejection and scene flow reasoning. Our proposed model is especially amenable to *bootstrapped ego-motion learning in robots* where the supervision in ego-motion estimation for a particular camera sensor can be obtained from the fusion of measurements from other robot sensors such as GPS/INS, wheel encoders etc. We expect our approach to be fairly general-purpose and transferable, while its hyper-parameters to be easily fine-tuned for high performance accuracy and robustness during operation.

- **Self-Supervised Visual Place Recognition in Robots** We envision the capability of robots to self-supervise computer vision tasks such as visual ego-motion to be especially beneficial in the context of life-long perceptual learning in autonomous systems. In **Chapter 6**, we extend this capability to yet another critical component of vision-based SLAM, loop-closure detection. The task of visual loop-closure identification is cast as a similarity metric learning problem, where the labels for positive and negative examples of loop-closures

can be self-supervised by an existing navigation solution (GPS/SLAM) that the robot uses. By leveraging the synchronization between sensors, we show that we are able to transfer and learn a metric for image-image similarity in an embedded space by sampling corresponding information from the navigation solution space (Pillai and Leonard 2017b). Furthermore, we show that the newly learned embedding can be particularly powerful in disambiguating visual scenes from each other.

We envision that self-supervised solutions to task learning will have far-reaching implications in several domains, especially in the context of life-long learning in autonomous systems. Furthermore, we expect these techniques to seamlessly operate under resource-constrained situations in the near future by leveraging existing solutions in model reduction and dynamic model architecture tuning. With the availability of multiple sensors on these autonomous systems, we also foresee bootstrapped task learning to potentially enable robots to learn from experience, and use the new models learned from these experiences to encode redundancy and fault-tolerance all within the same framework. In **Chapter 7**, we foresee some of the implications of this thesis, and discuss some future research directions to realize this vision (Fourie et al. 2017; Moll et al. 2017).

Chapter 2

Background

Simultaneous Localization and Mapping, widely known as SLAM, is a fundamental capability in robots that allow them to map their immediate environment while simultaneously being able to localize themselves within it. We briefly describe the different variants of SLAM relevant to this thesis.

2.1 SLAM Landscape

SLAM has traditionally been considered as the back-end optimization routine for various robot navigation tasks. The SLAM problem (Bailey and Durrant-Whyte 2006; Durrant-Whyte and Bailey 2006; Thrun and Leonard 2008; Thrun et al. 2005) is formulated in terms of a Bayes Net shown in Figure 2-1. We define $\mathbf{X} = \{\mathbf{x}_i\}, i \in 0, \ldots, M$ as the robot's trajectory through time, with $\mathbf{x}_i$ as the vector representation of the state or pose of the robot. The state of a robot is typically parameterized in $SE(2)$ or $SE(3)$, and may also include other parameters to be estimated such as instantaneous velocity and acceleration. The latent positions of landmarks are maintained as $\mathbf{L} = \{\mathbf{l}_j\}, j \in \{1, \ldots, N\}$, with $\mathbf{Z} = \{\mathbf{z}_k\}, k \in \{1, \ldots, K\}$ denoting the noisy measurements to the landmarks detected. The robot's motion estimate, typically recovered from its wheel-encoders or Inertial-Measurement Unit (IMU), are also incorporated as noisy measurements denoted by $\mathbf{U} = \{\mathbf{u}_i\}, i = \{1, \ldots, M\}$.

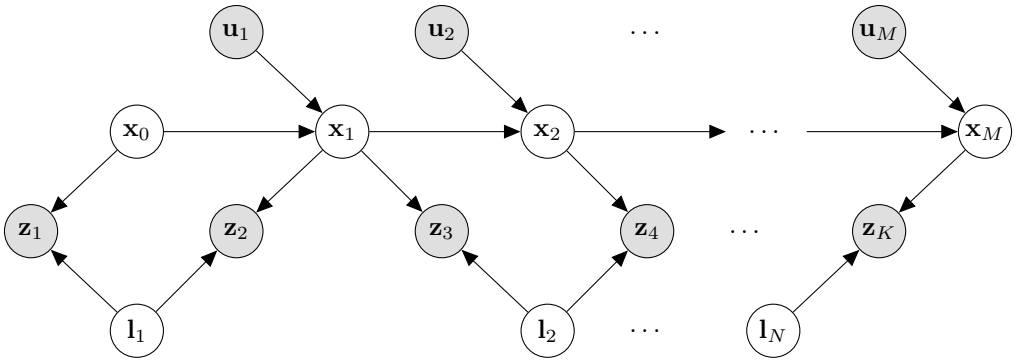


Figure 2-1: **SLAM as a Bayes Net** ▶ Bayes network representation of the SLAM problem. $\mathbf{x}_i$ represents the state of the robot's trajectory at time i, $\mathbf{l}_j$ is the location of the landmark j, $\mathbf{u}_i$ is the robot's odometry measurement at time i, and $\mathbf{z}_k$ represents the k^{th} landmark measurement. The figure illustrates the conditional independence between variables, whose joint probability distribution is given by Equation 2.1. The observed variables $\mathbf{u}$, and $\mathbf{z}$ are drawn in a lighter gray shade, while the latent variables $\mathbf{x}$ and $\mathbf{l}$ are drawn in white.

2.1.1 Full SLAM

In the full SLAM formulation, both the robot's trajectory $\mathbf{X}$ and the landmarks $\mathbf{L}$ are simultaneously estimated, given the robot's odometry measurements $\mathbf{U}$ and the set of landmark sightings $\mathbf{L}$. The joint probability of all the latent variables given all the associated measurements can be written as,

$$p(\mathbf{X}, \mathbf{L} \mid \mathbf{U}, \mathbf{Z}) \propto p(\mathbf{x}_0) \prod_{i=1}^{M} p(\mathbf{x}_i \mid \mathbf{x}_{i-1}, \mathbf{u}_i) \prod_{k=1}^{K} p(\mathbf{z}_k \mid \mathbf{x}_{ik}, \mathbf{l}_{jk}) \tag{2.1}$$

$$\propto \underbrace{\prod_{i=1}^{M} \exp\left(-\frac{1}{2} \left\| f_u(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i \right\|^2_{\sum_u}\right)}_{\text{Influence of odometry measurements}} \underbrace{\prod_{k=1}^{K} \exp\left(-\frac{1}{2} \left\| h_k(\mathbf{x}_{ik}, \mathbf{l}_{jk}) - \mathbf{z}_k \right\|^2_{\sum_k}\right)}_{\text{Influence of landmark measurements}} \tag{2.2}$$

where $p(\mathbf{x}_0)$ is the prior on the initial state of the robot, $p(\mathbf{x}_i \mid \mathbf{x}_{i-1}, \mathbf{u}_i)$ is the influence of the motion model on the state of the system, $p(\mathbf{z}_k \mid \mathbf{x}_{ik}, \mathbf{l}_{jk})$ is the influence of landmark measurements, while assuming appropriate data association (i_k, j_k) between landmark sightings $\mathbf{z}_k$. The measurements in the system $\mathbf{U}, \mathbf{Z}$ are corrupted by noise that is assumed to be Gaussian, with zero-mean and covariance $\sum_u$ and $\sum_k$ respectively.

In order to recover an optimal estimate of the variables defined in the system, we re-formulate the above equation 2.1 in an equivalent least-squares form. The *maximum a posteriori* (MAP) position estimate of the robot's trajectory and the land-

marks detected can be then be defined as,

$$\mathbf{X}^*, \mathbf{L}^* = \arg\max_{\mathbf{X},\mathbf{L}} p(\mathbf{X}, \mathbf{L} \mid \mathbf{U}, \mathbf{Z}) \tag{2.3}$$

$$= \arg\max_{\mathbf{X},\mathbf{L}} \left\{ \log p(\mathbf{X}, \mathbf{L} \mid \mathbf{U}, \mathbf{Z}) \right\} \tag{2.4}$$

$$= \arg\min_{\mathbf{X},\mathbf{L}} \left\{ -\log p(\mathbf{X}, \mathbf{L} \mid \mathbf{U}, \mathbf{Z}) \right\} \tag{2.5}$$

$$= \arg\min_{\mathbf{X},\mathbf{L}} \left\{ \underbrace{\sum_{i=1}^{M} \| f_u(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i \|^2_{\sum_u}}_{\text{Odometry Measurement Factors}} + \underbrace{\sum_{k=1}^{K} \| h_k(\mathbf{x}_{ik}, \mathbf{l}_{jk}) - \mathbf{z}_k \|^2_{\sum_k}}_{\text{Landmark Measurement Factors}} \right\} \tag{2.6}$$

where $\|\mathbf{v}\|_{\sum} = \mathbf{v}^T \sum^{-1} \mathbf{v}$ is the squared Mahalonbis distance with covariance matrix $\sum$. For a detailed introduction to SLAM, we refer the reader to (Kaess et al. 2008; Thrun et al. 2005).

2.1.2 Pose SLAM

In some applications such as localization, it may not be necessary to maintain and recover the set of landmarks and their associations into the future. This particular application leads to Pose SLAM (Konolige et al. 2010b;c; Lu and Milios 1997), where two subsequent set of landmarks are registered between each other to identify a relative pose constraint between them. This effectively marginalizes out all the landmarks in the scene, and instead considers the uncertainty indirectly in the relative pose estimated. As the robot explores its immediate environment, it may encounter previously visited locations that it can incorporate as additional constraints into the overall SLAM objective. These *loop-closure constraints* are introduced as relative pose constraints by registering the previously visited location measurement with the current view of it. Figure 2-3 shows the graphical interpretation of Pose SLAM obtained from connecting subsequent nodes in the odometry chain (via odometry or control input measurements), and establishing edges between temporally distant nodes (via loop-closure detection). The MAP estimate of the robot's trajectory then reduces to,

$$\mathbf{X}^* = \arg\max_{\mathbf{X}} p(\mathbf{X} \mid \mathbf{U}, \mathbf{Z_c}) \tag{2.7}$$

$$= \arg\min_{\mathbf{X}} \left\{ \underbrace{\sum_{i=1}^{M} \|f_u(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i\|^2_{\sum_u}}_{\text{Odometry Measurement Factors}} + \underbrace{\sum_{(j,k)\in\mathcal{C}} \|h_c(\mathbf{x}_j, \mathbf{x}_k) - \mathbf{z}_{jk}\|^2_{\sum_c}}_{\text{Loop-Closure Constraint Factors}} \right\} \tag{2.8}$$

2.1.3 Data Association

Data association is one of the key components in a SLAM system (Bar-Shalom et al. 1990). While a lot of care is taken in setting up the optimization objective, it is critical to ensure that the measurements fed into the back-end optimization is not erroneous. Data association can be evaluated in the same way as classical recognition related tasks: they need to achieve high-precision in the set of measurements associated, while ensuring high-recall of the relevant measurements that can be associated (Neira and Tardós 2001). We elaborate on the necessity of robust data association in Section 2.3.

2.2 Factor Graphs for SLAM

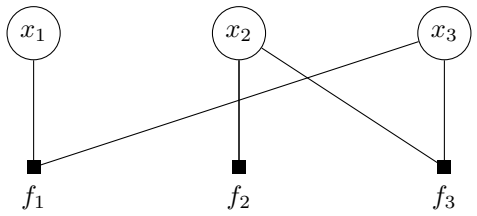


Figure 2-2: **Factor graph example** ▶ A factor graph is a bipartite graph that describes the factorization of a joint probability distribution over latent random variables. The figure illustrates the conditional independence constraints between variables, whose joint probability distribution can be written as the product of the m factors, given by $f(x_1, x_2, x_3) = \prod_{i=1}^{m} f_i(\mathcal{X}_i) = f_1(x_1, x_3) f_2(x_2) f_3(x_2, x_3)$. $\mathcal{X}_i$ refers to the subset of variables that f_i depends on.

A *factor graph* (Kschischang et al. 2001) is a bipartite graph that encodes how a function of several variables factorizes into its a product of local functions. A factor graph typically consists of nodes representing latent variables considered in the estimation problem, and factors that represent the information between or on

these variables.

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{m} f_i(\mathcal{X}_i) \tag{2.9}$$

Intuitively, a factor graph encodes the conditional independence inherent in the joint distribution over the set of variables considered. We consider the factor graph representation to be especially elegant for formalizing and intuitively describing the different applications of SLAM in this thesis.

Factor graphs were introduced by Kschischang et al. (2001) as a modern probabilistic tool for factorization of and inference over arbitrary functions and probability distributions. These have recently been applied to SLAM (Dellaert 2012; Dellaert et al. 2017; Kaess et al. 2011), where the joint probability distribution over the relevant variables is factored as a product over measurement factors. Figure 2-3 illustrates the Pose-SLAM problem, reformulated in the form of a factor graph.

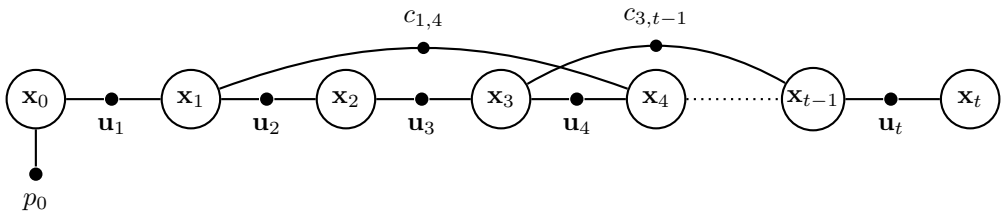


Figure 2-3: **Pose-Graph SLAM: Pose SLAM as a factor graph ►** A typical factor-graph formulation of **Pose SLAM**, where the odometry factors are represented as $\mathbf{u}_i$ and loop-closure factors are represented as $c_{j,k}$. Factors are filled-in black nodes, and the latent variables are represented in white circles. The prior p_0 is also incorporated as a measurement factor in the far left.

2.2.1 Bundle Adjustment

In Bundle Adjustment (BA) (Hartley and Zisserman 2003; Triggs et al. 1999), the variables $\mathbf{x}_i$ represent the camera poses, while the factors represent the multi-view constraints that are derived from multiple 2D projections of the same 3D landmark point $\mathbf{l}_j$. Bundle Adjustment applications in robotics however, can leverage other sensory measurements such as IMU, or wheel odometry to further improve the overall estimate of the robot's trajectory and the map. Figure 2-4 illustrates this application of Bundle Adjustment in a factor graph while simultaneously including odometry information typically measured in mobile robots. The 3D landmarks $\mathbf{l}_j$ may be sighted from various views along the robot's trajectory, with 2D image-based measurements referred to as m_k. Equation 2.6 refers to the visual-SLAM

formulation described earlier, that incorporates both odometry measurements and landmark sightings as factors in the overall state-estimation. The classical BA problem can be written as follows:

$$\mathbf{X}^*, \mathbf{L}^* = \arg\max_{\mathbf{X},\mathbf{L}} p(\mathbf{X}, \mathbf{L} \mid \mathbf{Z}_\mathbf{l}) \tag{2.10}$$

$$= \arg\min_{\mathbf{X}} \sum_{k=1}^{K} \| h_k(\mathbf{x}_{ik}, \mathbf{l}_{jk}) - \mathbf{z}_k \|^2_{\sum_k} \tag{2.11}$$

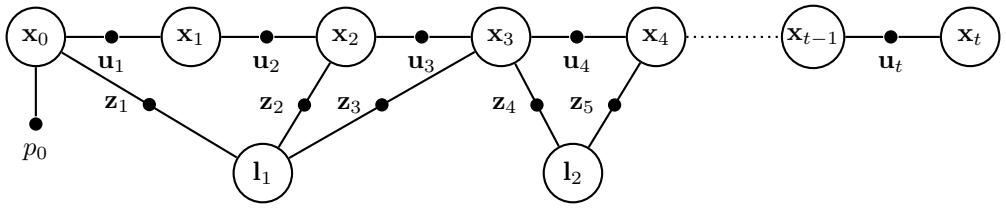


Figure 2-4: **Visual-SLAM: Bundle Adjustment (BA) contained in a factor graph** ▶ A typical factor-graph formulation of **Bundle-Adjustment**, where the odometry factors are represented as $\mathbf{u}_i$ and landmarks are represented as $\mathbf{l}$. The measurements from the robot to the various landmarks at different timesteps are indicated as $\mathbf{z}$.

One of the major difficulties in classical Bundle Adjustment is the scale ambiguity problem. In the objective function 2.11, the camera pose and landmarks are estimated up to scale, implying that the system can be scaled down or up without affecting the overall residual term. However, in most robotic applications where odometry measurements are available, we are able to introduce an over-complete set of measurements to recover the scale of the system, while being able to simultaneously incorporate the Bundle Adjustment objective within the same factor graph (Equation 2.13). Figure 2-4 graphically illustrates how these measurements are incorporated to recover the robot's trajectory, while simultaneously performing Bundle Adjustment.

$$\mathbf{X}^*, \mathbf{L}^* = \arg\max_{\mathbf{X},\mathbf{L}} p(\mathbf{X}, \mathbf{L} \mid \mathbf{U}, \mathbf{Z}_\mathbf{l}) \tag{2.12}$$

$$= \arg\min_{\mathbf{X},\mathbf{L}} \Bigg\{ \underbrace{\sum_{i=1}^{M} \| f_u(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i \|^2_{\sum_u}}_{\text{Odometry Measurement Factors}} + \underbrace{\sum_{k=1}^{K} \| h_k(\mathbf{x}_{ik}, \mathbf{l}_{jk}) - \mathbf{z}_k \|^2_{\sum_k}}_{\text{Bundle Adjustment Problem}} \Bigg\} \tag{2.13}$$

2.2.2 GPS-aided Localization

Another application of localization-only SLAM that we shall refer to in later chapters is GPS-aided localization (Indelman et al. 2013). This is typically considered in standard navigation-related tasks where the goal is to fuse mutually uncorrelated sensor measurements from wheel odometry or IMUs and GPS. While GPS is known to provide precise global positioning on a coarser timescale, IMUs and wheel odometry operate at much higher frequencies providing accurate and fine-grained relative pose estimates on a shorter time-scale. The fusion of both these complementary measurements allow us to recover globally-consistent, and accurate, long-term trajectories that the robot has observed. This is formalized as,

$$\mathbf{X}^* = \arg\max_{\mathbf{X}} p(\mathbf{X} \mid \mathbf{U}, \mathbf{Z_g}) \tag{2.14}$$

$$= \arg\min_{\mathbf{X}} \left\{ \underbrace{\sum_{i=1}^{M} \|f_u(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i\|^2_{\sum_u}}_{\text{Odometry Measurement Factors}} + \underbrace{\sum_{j=1}^{G} \|h_g(\mathbf{x}_j) - \mathbf{z}_j\|^2_{\sum_g}}_{\text{GPS Measurement Priors}} \right\} \tag{2.15}$$

Figure 2-5 illustrates the equivalent factor graph representation of this specialized SLAM problem.

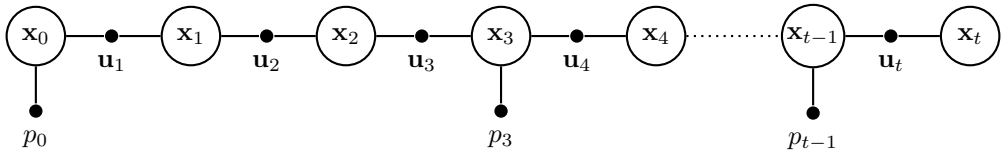


Figure 2-5: **GPS-Aided Localization as a factor graph** ▶ The factor graph illustration of the GPS-Aided Localization problem that we solve to enable self-supervision in Chapters 5 and 6. The odometry factors are represented as $\mathbf{u}_i$ and GPS measurement prior factors are represented as p_j.

2.3 Vision-based SLAM Front-Ends

So far we have described the critical optimization objective that is responsible for accurate recovery of the robot's trajectory and the landmarks that it has observed. We refer to this component as the "*back-end*" to the SLAM implementation, as it sits behind an abstraction layer that is agnostic to the different sensors and measurement modalities available to the robot. A typical mobile robot may be equipped with a multitude of sensors including cameras, laser range-finders, wheel-encoders, IMUs (Inertial Measurement Units), GPS modules etc. We expect

robots to maximize the utility of these sensors they are equipped with, leveraging all the cross-modal information to ensure accurate, robust and fault-tolerant operation. As we observe the several variants of the SLAM objective described in Section 2.1, a key observation is that these measurements are assumed to be associated to begin with. While the back-end may be able to provide some level of robustness to false associations, it is still imperative that the measurements from the variety of sensors are well-calibrated, and associated before they are incorporated into the overall optimization. This task is done by what we refer to as the "*front-end*". Thus, the front-end varies depending on the application-specific requirements of the robot and the sensors it is equipped with.

In this thesis, we focus on Visual-SLAM front-ends that primarily use either a single camera (also referred to as monocular SLAM (Davison et al. 2007)) or a combination of monocular camera with robot sensors such as wheel encoders and GPS. We shall now describe two key components of a vision-based SLAM pipeline: (i) *Visual Odometry* (VO), and (ii) Vision-based *Loop-Closure Recognition*.

2.3.1 Visual Odometry

Visual Odometry (VO) (Nistér et al. 2004), otherwise known as visual ego-motion estimation, is the process of determining the camera's trajectory, typically parameterized in $SE(3)$ from sequential images captured by a single or a set of cameras.

In VO, the main task is to recover the 6-DOF pose of the camera as it traverses through a scene. The pose of the camera is typically represented as a rigid-body transformation in $SE(3)$, given by $T_{t,t-1} \in \mathbb{R}^{4\times4}$:

$$T_{t,t-1} = \begin{bmatrix} R_{t,t-1} & t_{t,t-1} \\ 0 & 1 \end{bmatrix} \tag{2.16}$$

where $R_{t,t-1} \in SO(3)$ is the orthonormal rotation matrix, and $t_{t,t-1} \in \mathbb{R}^{3\times1}$ is the translation vector. The camera's pose at any point t can be recovered by compounding the set of transformations, $T_{t,0} = T_{t,t-1} * T_{t-1,t-2} * \cdots * T_{1,0}$ with the initial state T_0 arbitrarily defined based on the application. Following the notation from (Cheeseman et al. 1987), we use an equivalent notation to define compounding pose transformations.

$$\mathbf{z}_{t,0} = \mathbf{z}_{t,t-1} \oplus \mathbf{z}_{t-1,t-2} \oplus \cdots \oplus \mathbf{z}_{1,0} \tag{2.17}$$

First, the relative transformations $\mathbf{z}_{t,t-1}$ (i.e. the pose at time t relative to the previous timestep $t-1$) are computed from subsequent images $\mathcal{I}_t$, $\mathcal{I}_{t-1}$, and then compounded in order to recover the full trajectory $\mathbf{z}_{t,0}$ (i.e. *dead-reckoned*). We shall now discuss how these relative pose terms are recovered from subsequent images.

Recovering Pose Instead of associating individual landmark measurements across multiple frames as done in Bundle Adjustment, the relative-pose measurements are estimated from subsequent camera frames. This effectively marginalizes out the landmarks, and indirectly encodes their uncertainty in the relative-pose estimated between subsequent views (Nistér et al. 2004).

Feature Detection, Matching and Pose Estimation Visual Odometry estimation can be broadly be classified in to two categories: feature-based and direct methods. Feature-based methods, rely on recovering relative pose information by detecting and tracking salient and repeatable features across the images. Direct-methods, otherwise referred to as *global* methods, consider the image intensities across all pixels in the image sequence to inform incremental pose recovery. While both variants have their advantages, feature-based methods are known to be more robust primarily because they have been well-studied in the past two to three decades. In this thesis, we focus on the former and provide a brief introduction to feature-based visual SLAM.

VO implementations can be broken down in to a few key steps: (i) *Feature Detection*, (ii) *Feature Matching*, (iii) *Pose Estimation*, and (iv) *Pose Optimization (Optional)*. In the feature detection stage, salient and repeatable features are detected in every new image $\mathcal{I}_t$ and potentially matched with those detected in previous frames $\mathcal{I}_{t-1}$. The feature matching step involves describing these locally detected features with feature descriptors such as SIFT, SURF, ORB, BRIEF etc (Bay et al. 2006; Calonder et al. 2010; Lowe 1999; Rublee et al. 2011) and efficiently matching them with appropriate distance metrics and indexing strategies. In some situations where the relative motion is sufficiently small between subsequent image frames (i.e. high camera frame-rate or slow camera motion), certain methods leverage sparse optical flow techniques such as multi-scale Lucas-Kanade Optical Flow (Birchfield 2007; Lucas et al. 1981) to enable feature tracking. Once the feature correspondences are established, the relative pose $\mathbf{z}_{t-1,t}$ is recovered by two-view motion estimation algorithms depending on the application and motion constraints the camera may have (Fraundorfer et al. 2010; Longuet-Higgins 1987; Nistér et al. 2004; 2006; Scaramuzza et al. 2009b; Wang et al. 2005). Once the frame-to-frame pose estimate is recovered, they are compounded (Equation 2.17) to recover the full camera trajec-

tory. Optionally, a pose optimization step can be performed to reduce the overall error incurred in the pose-compounding procedure, and simultaneously reduce the overall uncertainty in the camera's trajectory. This optional step is referred to as Bundle Adjustment, and generally varies based on the application-specific requirements such as computational-constraints, or pose accuracy-requirements.

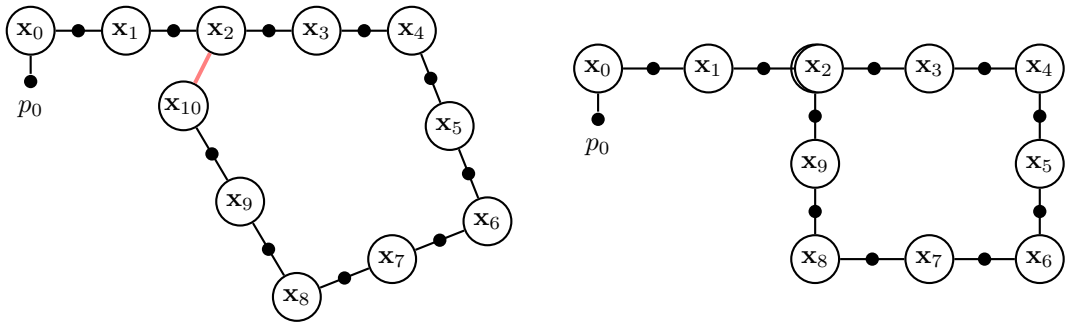
At various stages including feature matching, pose estimation, and pose optimization, it is imperative that the feature correspondences established are accurate and free of outliers. This is ensured via a model-guided *consensus* step (via RANSAC (Fischler and Bolles 1981), MLESAC (Torr and Zisserman 2000), M-estimation (Torr and Murray 1997) or other variants), where a constrained-model is repeatedly sampled, hypothesized and tested to minimize an appropriate geometric objective (such as the Sampson distance (Hartley and Zisserman 2003) in BA). Other solutions incorporate constraints such as structural scene priors (Guerrero et al. 2005; Wang et al. 2005), camera/vehicle motion models (Nistér et al. 2006; Scaramuzza et al. 2009a;b), or external sensor measurements (such as GPS, IMUs etc) (Jones and Soatto 2011; Konolige et al. 2010a; Mourikis and Roumeliotis 2007) to further improve outlier-rejection, and bolster pose estimation.

For a more thorough introduction to Visual Odometry estimation, we refer the reader to a two-part tutorial by Scaramuzza and Fraundorfer (2011) and Fraundorfer and Scaramuzza (2012). For a detailed literature review in visual odometry estimation, we refer the reader to Section 5.2 in Chapter 5

2.3.2 Vision-based Loop-Closure Recognition

Loop-closure recognition, also referred to as place recognition, uses visual cues contained in images to identify previously visited scenes. This information is used to incorporate constraints in the overall optimization in order to correct for the drift incurred in the overall dead-reckoned visual odometry estimate. As previously described in Section 2.1.2, the odometry estimates are maintained as edges within the pose-graph, connecting subsequent nodes. As the pose-graph chain grows, the uncertainty grows unbounded. Thus, intuitively, it can be valuable if we are able to establish loop-constraints between temporally distant nodes in graph in order to reduce the overall uncertainty propagated in the pose-graph chain (as illustrated in Figure 2-6). As the robot re-observes a previously observed landmark or scene, we are able to incorporate an additional constraint between the current node $\mathbf{x}_k$, and the matched previously observed node $\mathbf{x}_j$. This step in vision-based pose-graph

SLAM provides a sufficiently large improvement in overall trajectory estimation accuracy, and is an essential component in recovering globally consistent, and metrically accurate trajectory estimates for localization purposes. We illustrate this behavior in Figure 2-6.



(a) Open-loop odometry with loop-closure proposal **(b) Optimized trajectory with loop-closure constraints**

Figure 2-6: **Loop Closure Example** ▶ Dead-reckoned (open-loop) odometry chain typically drifts over time, with the uncertainty growing unbounded. Adding loop-closure constraints to dead-reckoned pose-graph chain provides useful information to recover the correct vehicle trajectory. **(a)** In this illustration, the robot at $\mathbf{x}_{10}$ has identified that it has previously visited the same location at $\mathbf{x}_2$, and establishes a loop-closure constraint (in red) between the corresponding nodes in the pose-graph. **(b)** The resulting optimized pose-graph corrects for the drift incurred in the dead-reckoned estimate, and shifts the pose-graph nodes to recover a more accurate robot trajectory. $\mathbf{x}_2$ and $\mathbf{x}_{10}$ physically overlap each other in this example, and only one of the nodes is shown.

Since we are interested in determining these loop-closure constraints from vision, identifying visual similarity in the form of semantic and geometry cues becomes extremely crucial. Visual similarity has been shown to be reliably computed using global image descriptors (Oliva and Torralba 2006; Sünderhauf and Protzel 2011; Ulrich and Nourbakhsh 2000), via bag-of-visual-words-based descriptions from local-descriptors (Cummins and Newman 2010; Fraundorfer et al. 2007; Newman et al. 2006), or more recently via Convolutional Neural Networks-based (CNN) descriptors (Chen et al. 2015b; 2017; Sunderhauf et al. 2015). We refer the reader to Section 6.2 in Chapter 6 for a more detailed overview and literature review in vision-based loop-closure recognition.

2.4 SLAM in this Thesis

In this thesis, we leverage two different variants of vision-based SLAM. In Chapter 3, we consider the classical *Visual-SLAM*, where the full Bundle Adjustment problem (Section 2.2.1) is solved (up to an unknown scale factor) to recover the camera's trajectory $\mathbf{X}^*$, along with the scale-ambiguous map $\mathbf{L}^*$. We take advan-

tage of this optimized solution to better inform object-recognition in mobile robots. In Chapters 5 and 6, we re-visit the capabilities of a *vision-based pose-graph SLAM* front-end. We consider a camera rigidly mounted on a mobile robot, and recover its trajectory as it traverses its environment for an extended period of time. In this specialized case, we are only concerned with recovering the optimal robot trajectory $\mathbf{X}^*$ over its entire session lifetime. By leveraging known navigation-based sensor fusion strategies such as GPS-aided SLAM (See Section 2.2.2), we are able to self-supervise mobile robots in tasks such as visual odometry estimation (Chapter 5) and vision-based loop-closure recognition (Chapter 6). The resulting models learned from these core SLAM-aided navigational tasks enables us self-supervise a vision-based pose-graph SLAM front-end that adapts to its operating environment, subsequently providing reliable measurements to a pose-graph SLAM back-end (Section 2.1.2). For a thorough introduction to factor-graph SLAM and its variants typically used in robot perception, please refer to (Dellaert et al. 2017).

Chapter 3

Monocular SLAM-Supported Object Recognition

Geometric and semantic scene understanding have been long-studied in the computer vision literature, however, they are still predominantly considered as two separate problems. While these two problems can mutually benefit each other, it still remains a fairly unexplored research problem to effectively utilize the strengths from both these capabilities. Robots, on the other hand, need to be able to contextualize all the relevant information available to them to make critical decisions they are tasked with. This necessitates tight integration between spatial-awareness, or its realization in mobile robots more commonly referred to as SLAM, and semantic-understanding in the form of object and scene recognition. We envision that such spatially-cognizant, scene-understanding capabilities can especially enable richer, and contextual perception in mobile robots making them far more capable than before.

3.1 Introduction

Object recognition is a vital component in a robot's repertoire of skills. Traditional object recognition methods have focused on improving recognition performance (Precision-Recall, or mean Average-Precision) on specific datasets (Everingham et al. 2010; Russakovsky et al. 2015). While these datasets provide sufficient variability in object categories and instances, the training data mostly consists of images of arbitrarily picked scenes and/or objects. Robots, on the other hand, perceive their environment as a continuous image stream, observing the same object

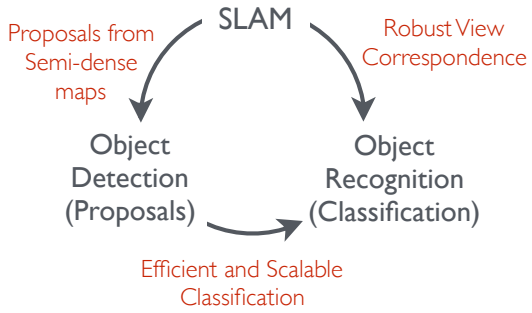


Figure 3-1: **Motivation for SLAM-Aware Object Recognition ►** One of the key motivations for this work is the need for *SLAM-awareness* in perceptual tasks such as object detection and classification. Robots that are spatially-cognizant of the world (with the ability to implicitly perform SLAM) can simultaneously take advantage of the maps they maintain to recover temporally consistent object proposals. Furthermore, by leveraging their localization capabilities, they are able to use robust view correspondence that SLAM provides for improved multi-view object classification and occlusion-aware spatial reasoning.

several times, and from multiple viewpoints, as it constantly moves around in its immediate environment. As a result, object detection and recognition can be further bolstered if the robot were capable of simultaneously localizing itself and mapping its immediate environment - by integrating object detection evidences across multiple views.

We refer to a "*SLAM-aware*" system as one that has access to the map of its observable surroundings as it builds it incrementally and the location of its camera at any point in time. This is in contrast to classical recognition systems that are "*SLAM-oblivious*" - those that detect and recognize objects on a frame-by-frame basis without being cognizant of the map of its environment, the location of its camera, or the objects that may be situated within these maps. In this work, we develop the ability for a SLAM-aware system to robustly recognize objects in its environment, using an RGB camera as its only sensory input (Figure 3-2).

We make the following contributions towards this end: Using state-of-the-art semi-dense map reconstruction techniques in monocular visual SLAM, we develop the capability to propose spatially consistent, scale-ambiguous object candidates within a 3D scene. Leveraging this object proposal method, we introduce the concept of *SLAM-aware object recognition*, where we bolster classical, frame-based object recognition in mobile robots by aggregating object evidences across multiple viewpoints, facilitated by a SLAM-aware representation. We incorporate some of the recent advancements in object classification methods, including Bag-of-Visual-Words-based (BoVW) (Arandjelovic and Zisserman 2013; Delhumeau et al. 2013; Jégou et al. 2010) and Convolutional Neural Network-based feature encoding (Gir-

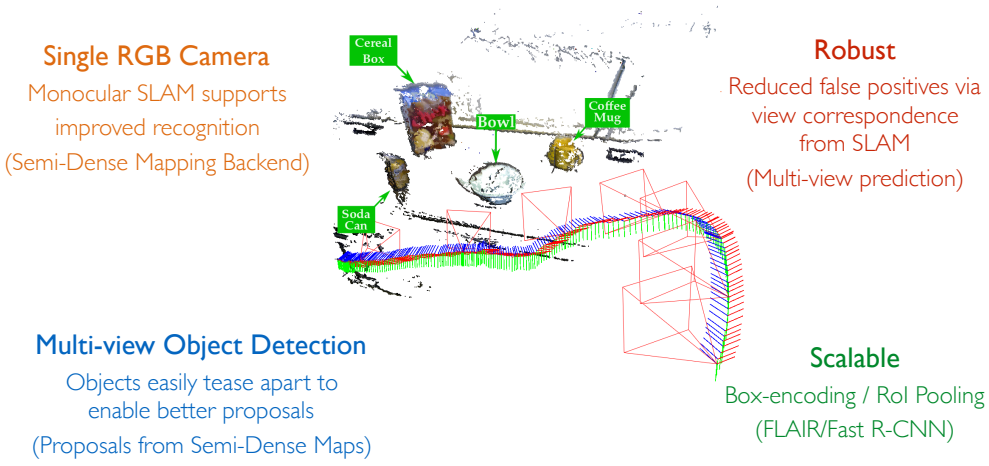


Figure 3-2: **SLAM-aware Object Recognition** ▶ The proposed SLAM-aware object recognition system is able to robustly localize and recognize several objects in the scene, aggregating detection evidence across multiple views. Annotations in white are provided for clarity and are actual predictions proposed by our system. Keyframe poses are shown with red camera frustums, while the 3-D triads correspond to the camera poses tracked on a frame-by-frame basis. *The labels in **green** are for illustrative purposes only.*

shick 2015; He et al. 2017; Redmon et al. 2016; Ren et al.; 2015), to enable strong recognition performance in monocular mobile systems. Additionally, we show that maintaining a SLAM-aware representation makes our system particularly amenable to few-shot object learning. Thus, the integration with a monocular visual-SLAM (vSLAM) back-end enables our SLAM-aware approach to take advantage of both the reconstructed map and camera location to significantly bolster object recognition, both during its training and deployment phases.

We present several experimental results validating the improved recognition performance of our proposed system: (i) The system is compared against the current state-of-the-art (Lai et al. 2012; 2014) on the UW-RGBD Scene (Lai et al. 2011; 2014) Dataset. We compare the improved recognition performance of being SLAM-aware to being SLAM-oblivious (i.e. classical frame-based techniques); (ii) We show that our approach easily extends to newer feature encoding techniques utilized in state-of-the-art CNN-based methods, further improving the recognition performance in single-camera equipped mobile robots; and (iii) By leveraging the underlying semi-dense reconstruction and optimized keyframes that our approach provides, we show that a SLAM-aware, few-shot object learning strategy can be especially advantageous to mobile robots that can learn quickly from a minimal set of experiences.

3.2 Related Work

We discuss some of the recent developments in object proposals, recognition, and the semi-dense monocular visual SLAM literature that has sparked the ideas described in this work.

Sliding window techniques and DPM In traditional state-of-the-art object detection, Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005) and Deformable-Part-based-Models (DPM) proposed by Felzenszwalb et al. (2010) have become the norm due to their success in recognition performance. These methods explicitly model the shape of each object and its parts via oriented-edge templates, across several scales. Despite its reduced dimensionality, the template model is scanned over the entire image in a sliding-window fashion across multiple scales for each object type that needs to be identified. This is a highly limiting factor in scalability, as the run-time performance of the system is directly dependent on the number of categories identifiable. While techniques have been proposed to scale such schemes to larger object categories (Dean et al. 2013), they incur a drop in recognition performance to trade-off for speed.

Dense sampling and feature encoding methods Recently, many of the state-of-the-art techniques (Lazebnik et al. 2006; van de Sande et al. 2014) for generic object classification have resorted to dense feature extraction. Features are densely sampled on an image grid (Bosch et al. 2007), described, encoded and aggregated over the image or a region to provide a rich description of the object contained in it. The aggregated feature encodings lie as feature vectors in a high-dimensional space, on which linear or kernel-based classification methods perform remarkably well. Among the most popular encoding schemes include Bag-of-Visual-Words (BoVW) (Csurka et al. 2004; Sivic and Zisserman 2003), and more recently Super-Vectors (Zhou et al. 2010), VLAD (Jégou et al. 2010), and Fisher Vectors (Perronnin et al. 2010b). In the case of BoVW, a histogram of occurrences of codes are built using a vocabulary of finite size $V \in \mathbb{R}^{K \times D}$. VLAD and Fisher Vectors, in contrast, aggregate residuals using the vocabulary to estimate the first and second order moment statistics in an attempt to reduce the loss of information introduced in the vector-quantization (VQ) step in BoVW. Both VLAD and Fisher Vectors have been shown to outperform traditional BoVW approaches (Chatfield et al. 2011; Jégou et al. 2010; Perronnin et al. 2010b), and are used as a drop-in replacement to BoVW; we do the same utilizing VLAD as it provides a good trade-off between descriptiveness and computation time.

Object Proposals Recently, many of the state-of-the-art techniques in large-scale object recognition systems have argued the need for a category-independent object proposal method that provides candidate regions in images that may likely contain objects. Variants of these include Constrained-Parametric Min-cuts (CPMC) (Carreira and Sminchisescu 2010), Selective Search (Uijlings et al. 2013), Edge Boxes (Zitnick and Dollár 2014), Binarized Normed Gradients (BING) (Cheng et al. 2014). The object candidates proposed are category-independent, and achieve detection rates (DR) of 95-99% at 0.7 intersection-over-union (IoU[1]) threshold, by generating about 1000-5000 candidate proposal windows (Hosang et al. 2014; Zitnick and Dollár 2014). This dramatically reduces the search space for existing sliding-window approaches that scan templates over the entire image, and across multiple scales; however, it still bodes a challenge to accurately classify irrelevant proposal windows as background. For a thorough evaluation of the state-of-the-art object proposal methods, and their performance, we refer the reader to Hosang et al. (2014).

Scalable Encoding with Object Proposals As previously addressed, sliding-window techniques inherently suffer from the scalability issue, despite recent schemes to speed-up such an approach. The BoVW approach handles this scalability issue rather nicely since the histograms do not particularly encode spatial relations as strongly. This however, inhibits BoVW approaches from localizing objects in an image. The advent of category-independent object proposal methods has subsequently opened the door to bag-of-words-driven architectures, where object proposal windows can now be described via existing feature encoding methods. van de Sande et al. (2014) employ a novel box-encoding technique using integral histograms to describe object proposal windows with a run-time independent of the window size of object proposals supplied. They report results with an 18x speedup over brute-force BoVW encoding (for 30,000 object proposals), enabling a new state-of-the-art on the challenging 2010 PASCAL VOC detection task at that time.

Convolutional Neural Networks Recently, Convolutional Neural Network (CNN) architectures have considerably changed the landscape of classical vision-based tasks such as image classification (Chatfield et al. 2014; Krizhevsky et al. 2012; Russakovsky et al. 2015; Szegedy et al. 2016; 2017), object recognition (Girshick 2015; Girshick et al. 2014a; Gupta et al. 2014; He et al. 2017; Redmon et al. 2016; Ren et al.; 2015), semantic segmentation (Badrinarayanan et al. 2015; Long et al. 2015; Yu and Koltun 2015) etc. Their adoption in a wide variety of image-based

[1]Intersection-over-Union (IoU) is a common technique to evaluate the quality of candidate object proposals with respect to ground truth. The intersection area of the ground truth bounding box and that of the candidate is divided by the union of their areas.

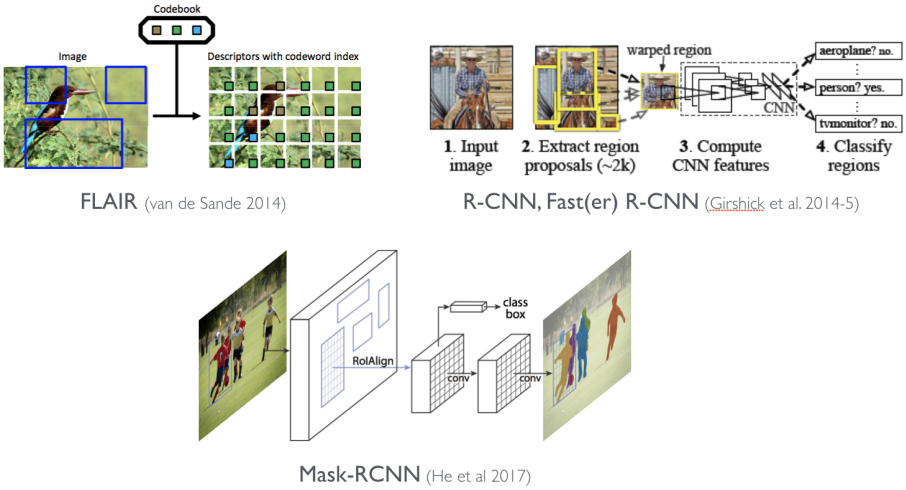


Figure 3-3: **Object Recognition Landscape** ▶ The accelerated evolution of object recognition in the recent years (Girshick 2015; He et al. 2017; Ren et al. 2015; van de Sande et al. 2014) has made it practical for robots to deploy these state-of-the-art systems with real-time performance capabilities.

applications have clearly justified their powerful representational capacity and rich spatial-semantic descriptive capability (Cao et al. 2016; Dosovitskiy et al. 2015; Ronneberger et al. 2015). Strictly within the object recognition landscape, there has been a sudden surge of advancements, leveraging these state-of-the-art CNN models with efficient object proposal, and encoding techniques (Girshick 2015; Girshick et al. 2014b; Gupta et al. 2014; Liu et al. 2016; Redmon et al. 2016; Ren et al. 2015). Some of these techniques (max-pooling, spatial pyramidal-pooling (Grauman and Darrell 2005; Lazebnik et al. 2006), efficient object proposals (Lienhart and Maydt 2002; Viola and Jones 2004)) have been developed in various forms in prior work. Their recent consideration in a joint framework, however, has enabled strong recognition performance that was previously considered challenging. Other, more recent works have built on top of CNN-based architectures to enable rich, contextual scene understanding methods using map reconstructions (Song et al. 2017; Xiang and Fox 2017) via RGB-D cameras. In this work, we limit ourselves to standard RGB cameras, and illustrate similar semantic scene understanding capabilities by leveraging state-of-the-art CNN-based object recognition methods coupled with spatial-awareness through monocular Visual-SLAM techniques.

Multi-view Object Detection While classical object detection methods focus on single-view-based recognition performance, some of these methods have been extended to the multi-view case (Collet and Srinivasa 2010; Thomas et al. 2006), by aggregating object evidence across disparate views. Lai et al. (2012) proposed a multi-view-based approach for detecting and labeling objects in a 3D environ-

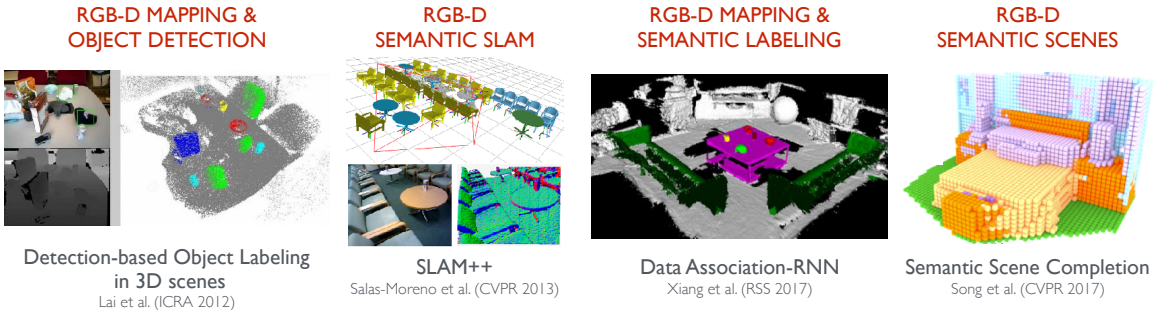


Figure 3-4: **Recognition in Unknown Maps ►** Various works have addressed the ability to combine map or depth information with object detection in order to enable strong recognition performance. However, most of these systems have been limited to RGB-D sensors, and have not proven the functionality with a monocular (RGB) camera (Lai et al. 2012; Salas-Moreno et al. 2013; Song et al. 2017; Xiang and Fox 2017).

ment reconstructed using an RGB-D sensor. They utilize the popular HOG-based sliding-window detectors trained from object views in the RGB-D dataset (Lai et al. 2011; 2014) to assign class probabilities to pixels in each of the frames of the RGB-D stream. Given co-registered image and depth, these probabilities are assigned to voxels in a discretized reconstructed 3D scene, and further smoothed using a Markov Random Field (MRF). Bao et al. (Bao and Savarese 2011; Bao et al. 2012) proposed one of the first approaches to jointly estimate camera parameters, scene points and object labels using both geometric and semantic attributes in the scene. In their work, the authors demonstrate the improved object recognition performance, and robustness by estimating the object semantics and SfM jointly. However, the run-time of 20 minutes per image-pair, and the limited object categories identifiable makes the approach impractical for on-line robot operation. Other works (Bo et al. 2011; Castle et al. 2010; Civera et al. 2011; Gupta et al. 2014; Salas-Moreno et al. 2013) have also investigated object-based SLAM, SLAM-aware, and 3D object recognition architectures, however they have a few of glaring concerns: either (i) the system cannot scale beyond a finite set of object instances (generally limited to less than 10), or (ii) they require RGB-D input to support both detection and pose estimation, or (iii) they require rich object information such as 3D models in its database to match against object instances in a brute-force manner. More recently, Wong et al. (2014; 2015) consider the data-association problem in multi-view detections, and cast the object label assignment as a Dirichlet Process Mixture Model. However, in our case, the Visual SLAM solution produces pixel-level accurate data associations across multiple views, rendering the data association problem simpler in practice.

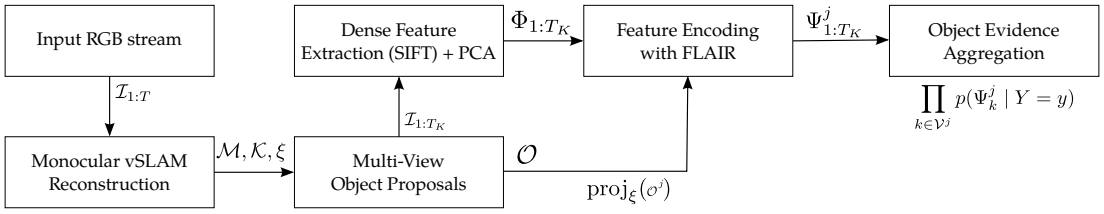


Figure 3-5: **Outline of the SLAM-aware (VLAD-FLAIR) object recognition pipeline ▶** Given an input RGB image stream $\mathcal{I}$, we first perform monocular visual-SLAM to recover the optimized camera trajectory and sparse landmarks. The scene is reconstructed in a semi-dense fashion using the optimized keyframes $\mathcal{K} = \{\mathcal{I}, \xi\}$, to recover the full map $\mathcal{M}$. We perform multi-scale density-based segmentation on the reconstructed scene to obtain object proposals $\mathcal{O}$ that are consistent across multiple views. On each of the images in the input RGB image stream $\mathcal{I}_{1:T}$, we compute Dense-SIFT ($\mathbb{R}^{384}$) in the RGB colorspace and reduce it to $\Phi \in \mathbb{R}^{80}$ via PCA. The features Φ are then used to efficiently encode each of the projected object proposals $o^j \in \mathcal{O}$ (bounding boxes of proposals projected on to the visible keyframes with known poses $\xi_{1:T_K}$) using VLAD with FLAIR, to obtain $\Psi_{1:T_K}$. The resulting feature vector Ψ is used to predict the likelihood of target label/category $p(\Psi^j_k \mid Y = y)$ of the object contained in each of the object proposals o^j. The likelihoods for each object $o^j \in \mathcal{O}$ are aggregated across each of the occlusion-free viewpoints $\xi_{1:T_K}$ to obtain robust object category prediction.

3.3 Monocular SLAM Supported Object Recognition

Traditional object recognition systems detect and recognize objects on an individual image basis, and do not maintain any spatial or temporal context between views of the same scene. Contrary to classical per-frame object proposal methodologies, robots observe the same instances of objects in its environment several times and from disparate viewpoints. Furthermore, robots can also actively control their motion through the world, which can enable active vision and directional attention of objects in a scene (Bajcsy 1988). One of the key realizations stemming from this work is the significance of *spatial-awareness* in perceptual tasks such as object detection and classification. In this section, we introduce the algorithmic components of our method, and further refer the reader to Figures 3-5 and 3-10 that illustrate the steps involved in our system. Algorithm 1 describes the key steps involved in our proposed method.

3.3.1 Monocular Visual SLAM

Throughout this thesis, we advocate that various perceptual tasks such as object recognition in mobile robots can benefit from being spatially cognizant of its environment. By maintaining this spatial awareness simultaneously, we expect robots to be able to contextually incorporate multiple observations of their world around them before making critical scene understanding decisions. In this work, we utilize

Algorithm 1 Monocular SLAM-Supported Object Recognition

Input: $\mathcal{I}_{1:T}$: Input image sequence
Output: $\hat{y}_{MLE}$: Most likely object label ($\forall\, o^j \in \mathcal{O}$)

▷ **1.** Semi-dense Reconstruction (Section 3.3.1)
▷ $\mathcal{M}$: Map points
▷ $\mathcal{K}$: vSLAM Optimized Keyframes ($\{\mathcal{I}_1, \xi_1), \ldots, (\mathcal{I}_{T_K}, \xi_{T_K})\}$)
$\mathcal{M}, \hat{\mathcal{K}}_{1:T_K} \leftarrow$ MonocularVisualSLAM($\mathcal{I}_{1:T}$)

▷ **2.** Multi-scale density based over-segmentation (Section 3.3.2)
▷ $o^j \in \mathcal{O}$: 3D Object Proposals
$\mathcal{O} \leftarrow$ MultiViewObjectProposals($\mathcal{M}$)

▷ Φ_k: Image Description (Dense-SIFT / Fast R-CNN) in the k^{th} keyframe
$\Phi_{1:T_K} \leftarrow$ EncodeImage($\mathcal{I}_{1:T_K}$)

▷ **3.** Pooling and Classification for each proposal
for $o^j \in \mathcal{O}$ **do**
  ▷ $\text{BB}_k^j = \text{BB}(\text{proj}_{\xi_k}(o^j))$: Bounding-box projection of object o^j in k^{th} keyframe with pose ξ_k
  ▷ Ψ_k^j: Pooled features for object o^j (FLAIR Encoding / R-CNN RoI-Pooling) (Section 3.3.3)
  $\Psi_k^j \leftarrow$ RoI_Pooling(BB_k^j, Φ_k) $\quad \forall \quad k = \{1, \ldots, T_K\}$
  ▷ **4.** Occlusion-aware object evidence aggregation (Section 3.3.5)
  $\hat{y}_{MLE}^j \leftarrow$ EvidenceAggregation($\Psi_{1:T_K}^j$)
end for

vision-based SLAM capabilities inherent in most mobile robots to better inform the task of object recognition. We build on top of a recently introduced monocular visual SLAM solution called ORB-SLAM (Mur-Artal et al. 2015). Due to the sparse feature-based representation that ORB-SLAM incorporates, we augment the output with a semi-dense mapping component to increase the map reconstruction-density, thereby providing qualitatively similar maps to those of Engel et al. (2014).

Keyframe-based vSLAM Given a sequence of images $\mathcal{I}_{1:T}$, the map is first initialized via an automatic map initialization step (Mur-Artal et al. 2015), before the incremental mapping proceeds. Once the map points ($\mathbf{L}$) and camera poses ($\mathbf{X} = \mathbf{x}_{1:T}$) are reconstructed and refined by a post-processing two-view bundle adjustment (BA) step, subsequent poses of the camera are tracked on a frame-by-frame basis via the 2D-to-3D EPnP algorithm (Lepetit et al. 2009). As new features are detected and added to the map, it soon becomes computationally expensive to optimize over all the observations in the image sequence. This is typically resolved with a marginalization step using *keyframes* (Klein and Murray 2007), where only a subset of the original frames are considered for the windowed BA optimization (See Figure 3-6).

Keyframes $\mathcal{K} = \{(\mathcal{I}_1, \xi_1), \ldots, (\mathcal{I}_{T_K}, \xi_{T_K})\}$ are tuples of images $\mathcal{I}_k \in \{\mathcal{I}_1, \ldots, \mathcal{I}_{T_K}\}$ and corresponding camera poses $\xi_k \in \{\xi_1, \ldots, \xi_{T_K}\}$, sampled such that they grow

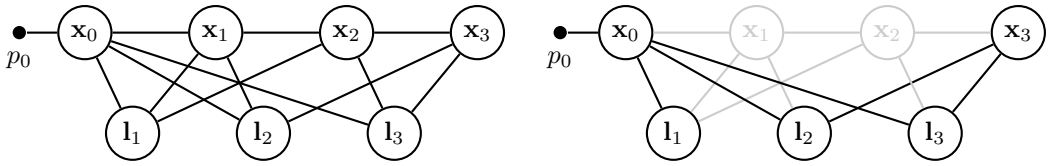


Figure 3-6: **Keyframe-based Visual-SLAM ►** Keyframe-based Bundle Adjustment (BA), where some of the poses (namely $\mathbf{x}_1$, $\mathbf{x}_2$) are marginalized out to reduce the overall complexity of full BA solution. The marginalized camera poses and their associated edges are rendered in a lighter color.

linearly in spatial coverage of the camera's viewing frustum. New keyframes are instantiated based on a set of criteria, typically based on the change in mutual-information (relative 6-DOF pose transformation, or number of features tracked) between the previously instantiated keyframe and the last frame tracked. As windowed BA continues, only the relevant keyframe's poses $\xi_{1:T_K}$ (a subset of the original set of poses $\xi_{1:T_K} \subseteq \mathbf{x}_{1:T}$ with $T_K << T$) and their associated map points $\mathbf{L}$ are optimized in an incremental manner allowing for real-time operation of large-scale visual SLAM problems. This has allowed keyframe-based visual SLAM implementations (Engel et al. 2014; Klein and Murray 2007; Konolige and Agrawal 2008; Mur-Artal et al. 2015; Strasdat et al. 2010; 2011) to truly scale to long operating times, as they are no longer bound linearly by time, but bound by the spatial coverage of the camera as it traverses through an environment. In later sections, we shall emphasize the value of keyframe-based sampling as they provide an elegant solution to the reduced computational complexity of the underlying bundle adjustment (BA) problem, while simultaneously providing *informative views* for efficient object recognition.

Semi-dense Reconstruction As the optimized poses $\hat{\xi}$ and sparse 3D landmarks $\hat{\mathbf{L}}$ converge within the windowed BA optimization, they can be directly used to further densify the 3D scene reconstruction. By sampling high-gradient regions in each of the keyframe images, we perform dense epiploar disparity estimation between the optimized keyframes, that we shall refer to as $\hat{\mathcal{K}}$. Using the proposed *depth filter* strategy (Forster et al. 2014), the relevant patch disparities are estimated directly using the inverse-depth parametrization (Civera et al. 2012), and filtered in a probabilistic and recursive manner. As more images are incrementally added to the system, new keyframes are instantiated and added to the pose-graph optimization and subsequent semi-dense reconstruction procedures.

Simultaneous Optimization and Map Densification Due to the incremental and real-time nature of the algorithm, the bundle adjustment (BA) optimization is only performed for a local window of keyframes that are contained within their *co-*

Figure 3-7: **Key-frame based Multi-View Semi-dense Reconstruction** ▶ Multi-view semi-dense reconstruction using *keyframes* significantly reduces the computational complexity of the underlying bundle adjustment (BA) problem, while simultaneously providing dense disparities and reconstructions for crucial tasks such as recovering object proposals, and reasoning about occlusions.

visibility sub-graph. We define the covisibility graph as (Strasdat et al. 2011), where each node or keyframe in the pose-graph is connected to other keyframes if they share visibility of the same landmark points between them. Since more recent keyframes are being updated and optimized as typically done in *windowed bundle adjustment*, the semi-dense reconstruction is delayed until the keyframes are no longer consumed in the *windowed* optimization (i.e. keyframes that are rendered inactive). Nevertheless, the sparse optimization and semi-dense reconstruction is performed simultaneously, resulting in a reconstruction that is qualitatively similar to those produced by other semi-dense reconstruction methods (Engel et al. 2014). In the following sections, we refer to this semi-dense reconstruction as the environment map $\mathcal{M}$ as it contains the subset of sparse landmarks $\hat{\mathbf{L}}$ optimized in the windowed bundle adjustment procedure. Figures 3-7 and 3-8 illustrate the multi-

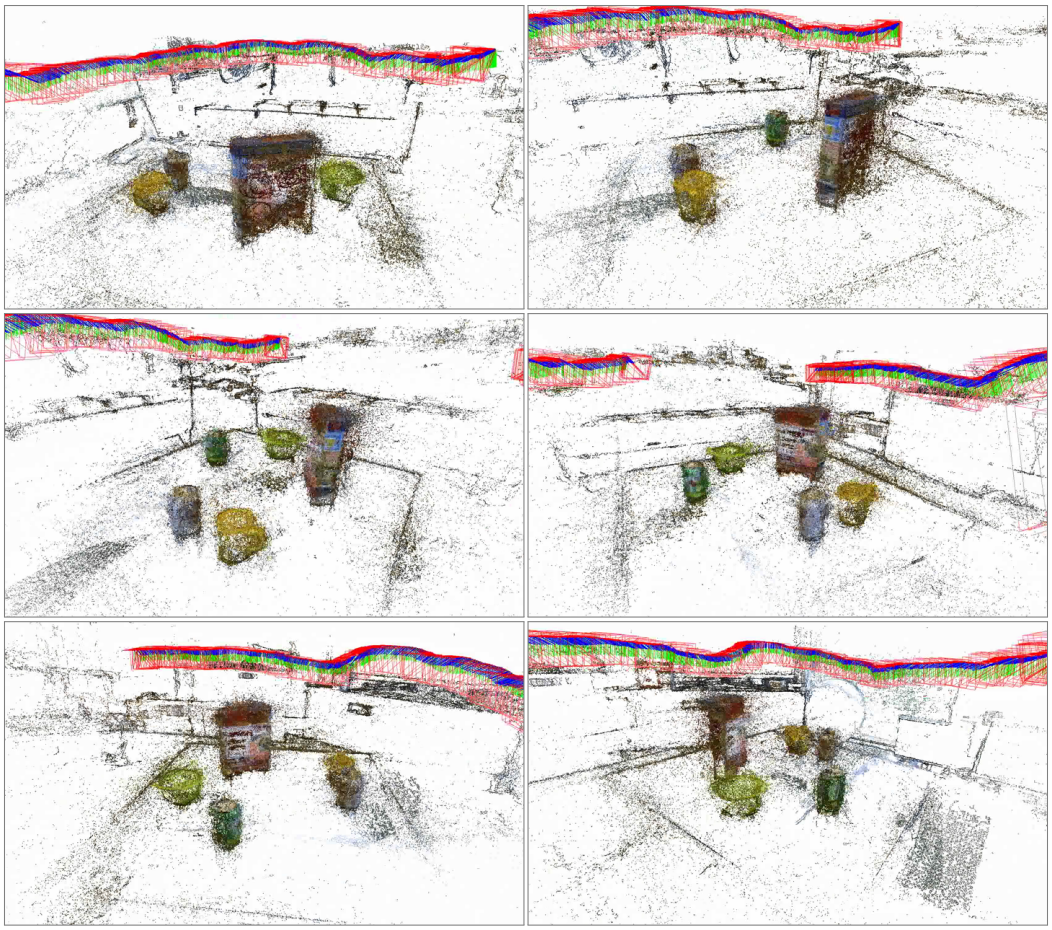


Figure 3-8: **Semi-dense Reconstruction ►** Semi-dense monocular reconstructions using *keyframes* allows our SLAM-aware solution to simultaneously take advantage of 3D reconstructions to recover temporally consistent object proposals, while providing valuable occlusion-aware reasoning during multi-view object classification stage. The images are captured from the same vantage point as the keyframe poses ξ_k, with some of the objects being occluded by other objects in the scene. By developing occlusion-aware, multi-view recognition systems, we are able to reason beyond single-views and maintain a spatially and semantically-cognizant world.

view, semi-dense disparity estimation and reconstruction of an indoor scene from the UW-RGBD Dataset (v2) (Lai et al. 2014). We note that while other semi-dense reconstruction implementations exist, they are typically based on direct-tracking methods that are typically not robust to wide-baseline motions. We leverage the resulting semi-dense reconstruction for subsequent object proposal generation in the next section.

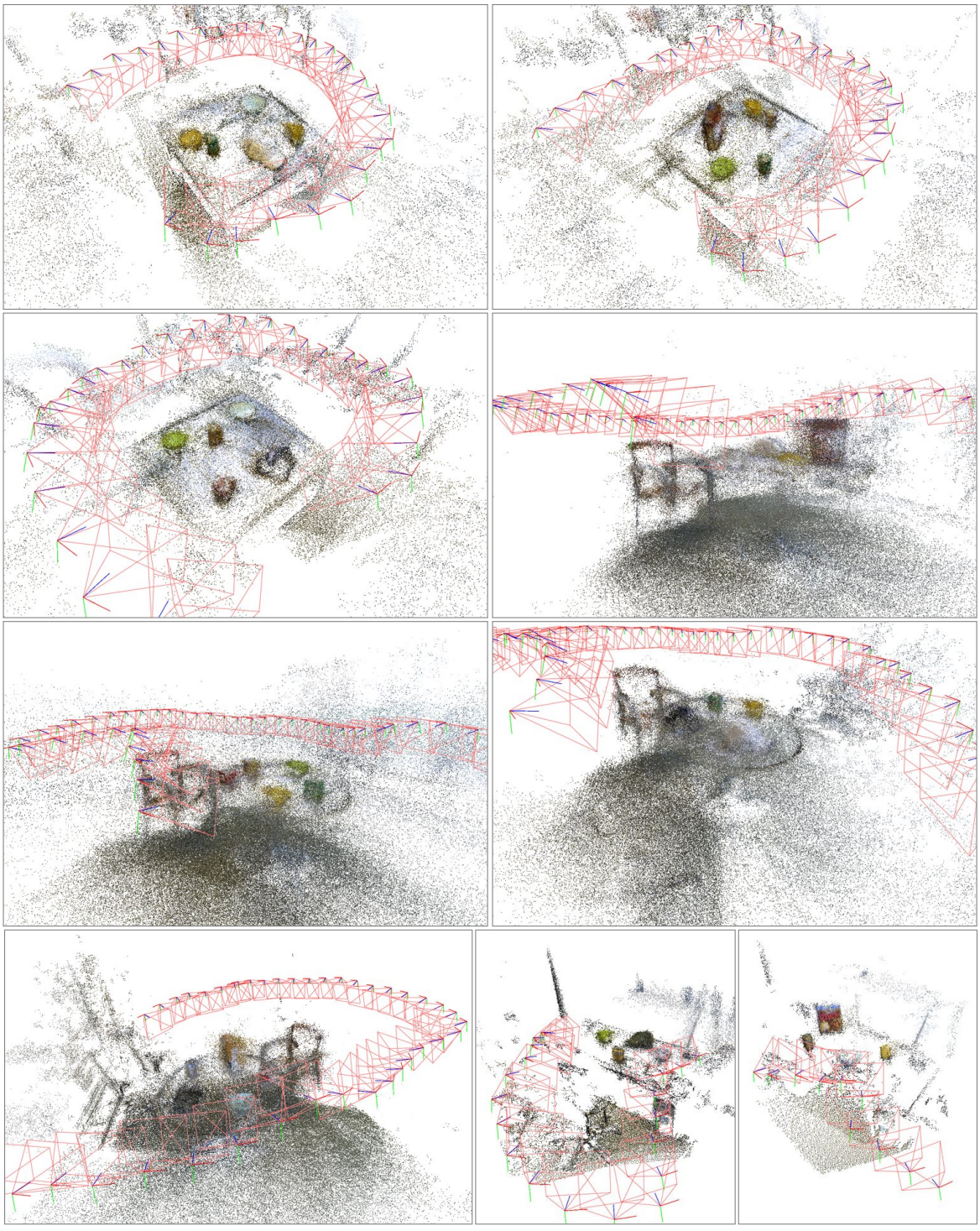


Figure 3-9: **Semi-dense Reconstruction of Indoor Scenes** ► Various indoor scenes from the UW RGBD Dataset (v2) reconstructed using our semi-dense reconstruction approach. The red camera frustums trace the camera's trajectory within the scene and indicate the keyframes estimated in the Visual SLAM optimization.

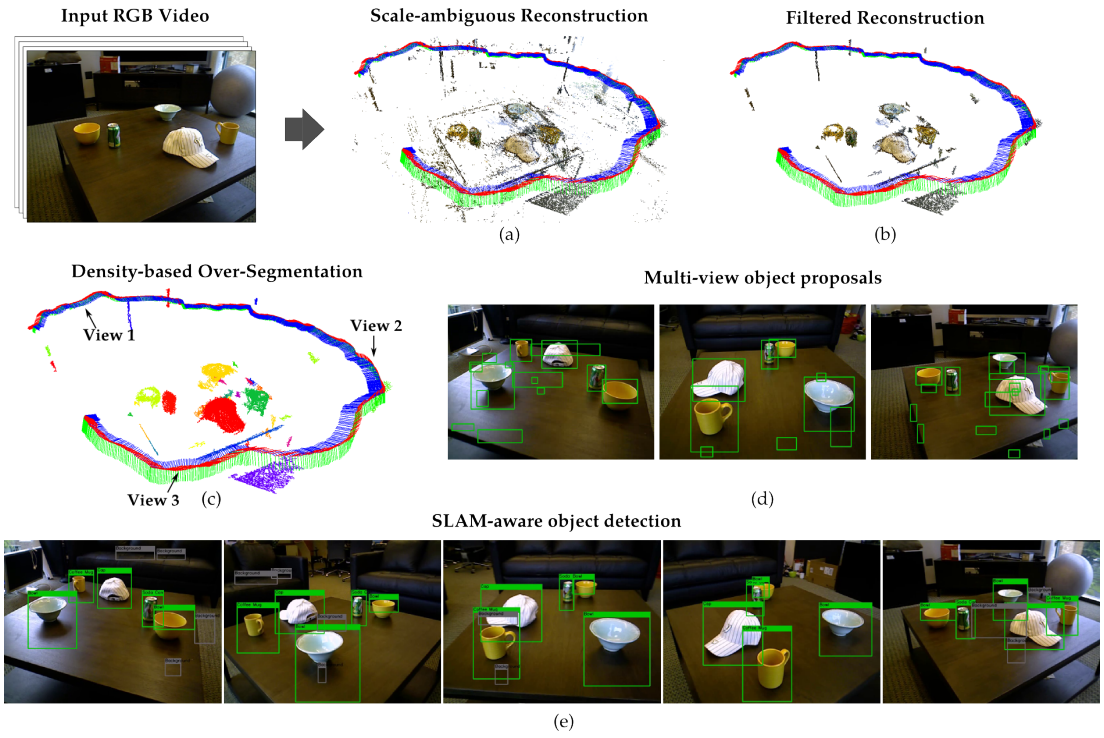


Figure 3-10: **SLAM-aware Object Recognition ►** An illustration of the multi-view object proposal method and subsequent SLAM-aware object recognition. Given an input RGB image stream, a scale-ambiguous semi-dense map is reconstructed (a) via the ORB-SLAM-based (Mur-Artal et al. 2015) semi-dense mapping solution. The reconstruction retains edges that are consistent across multiple views, and is employed in proposing objects directly from the reconstructed space. The resulting reconstruction is (b) filtered and (c) partitioned into several segments using a multi-scale density-based clustering approach that teases apart objects (while filtering out low-density regions) via the semi-dense edge-map reconstruction. Each of the clustered regions are then (d) projected on to each of individual frames in the original RGB image stream, and a bounded candidate region is proposed for subsequent feature description, encoding and classification. (e) The probabilities for each of the proposals per-frame are aggregated across multiple views to infer the most likely object label.

3.3.2 Multi-view Object Proposals

Most object proposal strategies use either superpixel-based or edge-based representations to identify candidate proposal windows in a single image that may likely contain objects. It is natural to think of object proposals from a spatio-temporal or reconstructed 3D context, and a key realization is the added robustness that the temporal component provides in rejecting spatially inconsistent edge observations or candidate proposal regions. Recently, Engel et al. (2014) proposed a scale-drift aware monocular visual SLAM solution called LSD-SLAM, where the scenes are reconstructed in a semi-dense fashion, by fusing spatio-temporally consistent scene edges. Despite being scale-ambivalent, the multi-view reconstructions can be especially advantageous in teasing apart objects from each other in the near-field ver-

sus those in the far-field regions. We take advantage of this insight, and develop an equivalent semi-dense visual-SLAM component that shall be a key enabler for improved object recognition in mobile robots.

In order to retrieve object candidates that are spatio-temporally consistent, we first perform a density-based clustering on the scale-ambiguous reconstruction using both spatial and edge color information. This is done repeatedly for 4 different density threshold values (each varied by a factor of 2), producing an over-segmentation of points in the reconstructed scene that are used as seeds for multi-view object candidate proposal. The spatial density segmentation eliminates any spurious points or edges in the scene, and the resulting point cloud is sufficient for object proposals. These object segments are subsequently projected onto each of the camera views, and directly serve as object proposals $\mathcal{O}$ for further classification. We ignore (i) bounding-box projections of object proposals whose window size is less than 20×20 pixels, (ii) occluded object proposals that do not satisfy the *z-buffer depth test* (see Section 3.3.5), and (iii) overlapping candidates with an IoU threshold of 0.5, to avoid redundant proposals. The filtered set of bounding box projections of object proposals is subsequently considered as candidates for the classification process downstream.

3.3.3 Encoding Object Proposals with VLAD and FLAIR

Given the object proposals computed using the reconstructed scale-ambiguous map, we now direct our attention to describing these proposal regions.

Dense BoVW with VLAD Given an input image and candidate object proposals, we first densely sample the image, describing each of the samples with SIFT across the RGB colorspace, $\Phi_{SIFT-RGB} \in \mathbb{R}^{384}$ i.e. Dense-SIFT (3 * 128-D). Features are extracted with a step size of 4 pixels, and at 4 different pyramid scales with a pyramid scale factor of $\sqrt{2}$. The resulting description is then reduced to a 80-dimensional vector via PCA, called PCA-SIFT $\Phi \in \mathbb{R}^{80}$. A vocabulary $V \in \mathbb{R}^{K_\text{v} \times 80}$ of size $K_\text{v} = 64$ is created via k-means, using the descriptions extracted from a shuffled subset of the training data, as done in classical bag-of-visual-words approaches. In classical BoVW, this vocabulary can be used to encode each of the original SIFT+RGB descriptions in an image into a histogram of occurrences of codewords, which in turn provides a compact description of the original image. Recently, however, more descriptive encodings such as VLAD (Jégou et al. 2010) and Fisher Vectors (Perronnin et al. 2010b) have been shown to outperform clas-

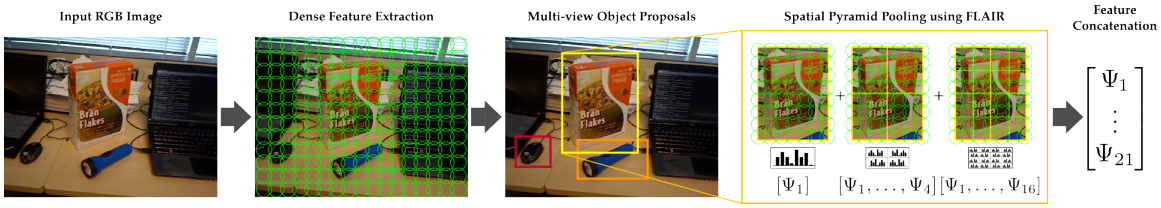


Figure 3-11: **VLAD feature extraction ►** Various steps involved in the feature extraction procedure. Features that are densely sampled from the image are subsequently used to describe the multi-view object proposals using FLAIR. Each proposal is described with multiple ([1x1], [2x2], [4x4]) spatial levels/bins via quick table lookups in the integral VLAD histograms (through FLAIR). The resulting histogram Ψ (after concatenation) is used to describe the object contained in the bounding box.

sical BoVW approaches (Chatfield et al. 2011; Jégou et al. 2010; Perronnin et al. 2010b). Consequently, we chose to describe the features using VLAD as it provides equally as strong performance with slightly reduced computation time as compared to Fisher Vectors.

For each of the bounding boxes, the un-normalized VLAD $\Psi \in \mathbb{R}^{K_v D}$ description is computed by aggregating the residuals of each of the descriptions Φ (enclosed within the bounding box) from their vector-quantized centers in the vocabulary, thereby determining its first order moment (Eq. 3.1).

$$\boldsymbol{v}_k = \sum_{\boldsymbol{x}_i : NN(\boldsymbol{x}_i) = \boldsymbol{\mu}_k} \boldsymbol{x}_i - \boldsymbol{\mu}_k \tag{3.1}$$

The description is then normalized using signed-square-rooting (SSR) or commonly known as power normalization (Eq. 3.2) with $\alpha = 0.5$, followed by L2 normalization, for improved recognition performance as noted in (Arandjelovic and Zisserman 2013).

$$f(z) = sign(\boldsymbol{z})|\boldsymbol{z}|^{\alpha} \quad \text{where} \quad 0 \leq \alpha \leq 1 \tag{3.2}$$

Additional descriptions for each bounding region are constructed for 3 different spatial bin levels or subdivisions as noted in (Lazebnik et al. 2006) (1×1, 2×2 and 4×4, 21 total subdivisions S), and stacked together to obtain the feature vector $\Psi = \left[\Psi_{1\times1}, \Psi_{2\times2}, \Psi_{4\times4}\right] \in \mathbb{R}^{K_v DS}$ that appropriately describes the specific object contained within the candidate object proposal/bounding box.

Efficient Feature Encoding with FLAIR While it may be efficient to describe a few object proposals in the scene with these encoding methods, it can be highly impractical to do so as the number of object proposals grows. To this end, van de Sande et al. (2014) introduced FLAIR — an encoding mechanism that utilizes summed-

area tables of histograms to enable fast descriptions for arbitrarily many boxes in the image. By constructing integral histograms for each code in the codebook, the histograms or descriptions for an arbitrary number of boxes B can be computed independent of their area. As shown in (van de Sande et al. 2014), these descriptions can also be extended to the VLAD encoding technique. Additionally, FLAIR affords performing spatial pyramid binning rather naturally, with only requiring a few additional table look-ups, while being independent of the area of B. We refer the reader to Figure 3-11 for an illustration of the steps involved in describing these candidate object proposals.

Multi-class histogram classification Given training examples, $(\Psi_1, y_1), \ldots, (\Psi_n, y_n)$ where $\Psi_i \in \mathbb{R}^{K_v DS}$ are the VLAD descriptions and $y_i \in \{1, \ldots, \mathcal{C}\}$ are the ground truth target labels, we train a logistic regression model using Stochastic Gradient Descent (SGD), given by:

$$E(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\Psi_i; \boldsymbol{w})) + \alpha R(\boldsymbol{w}) \tag{3.3}$$

where $L(y_i, f(\Psi_i; \boldsymbol{w})) = \log\Big(1+\exp(-y_i \boldsymbol{w}^T \Psi_i)\Big)$ is the logistic loss function, $R(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{w}^T \boldsymbol{w}$ is the L2-regularization term that penalizes model complexity, and $\alpha > 0$ is a non-negative hyperparameter that adjusts the L2 regularization. A one-versus-all strategy is taken to extend the classifiers to multi-class categorization. For hard-negative mining, we follow (van de Sande et al. 2014) closely, bootstrapping additional examples from wrongly classified negatives for 2 hard-negative mining epochs.

We note that in the past few years since this original work, several recent methods leveraging CNN-based methods have outperformed hand-engineered feature descriptors such as those described in this section. However, in the next section, we show how our solution extends easily to incorporating these newer CNN-based methods.

3.3.4 Encoding Object Proposals with CNN-based Methods

With the advent of Convolutional-Neural-Network techniques, there has been a recent surge of image classification and object recognition (Girshick 2015; He et al. 2017; Redmon et al. 2016; Ren et al.; 2015) techniques that have significantly outperformed these classical Bag-of-Visual-Words (van de Sande et al. 2014) or Histogram-

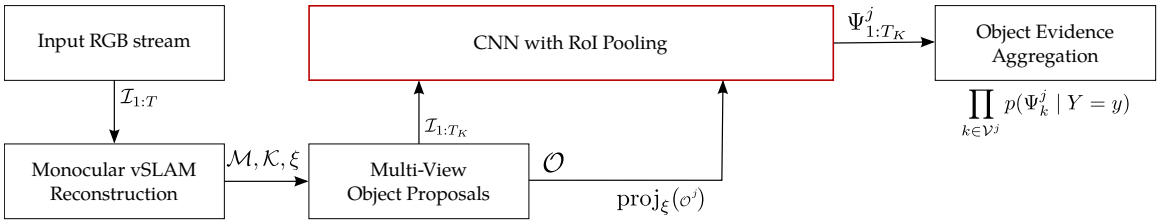


Figure 3-12: **Extensions to R-CNN ►** System architecture of our proposed SLAM-aware object recognition pipeline with more recent Convolutional-Neural Networks. Most of the components are essentially unchanged with the CNN handling the Region-of-Interest Pooling to recover the features $\Psi^j_{1:T_K}$ for classification and subsequent object evidence aggregation.

of-Gradients (Dalal and Triggs 2005; Felzenszwalb et al. 2010) based methods. Figure 3-3 shows the landscape and recent surge in object recognition methods based on state-of-the-art CNN techniques. We show that our SLAM-aware recognition pipeline is trivially extended to these recent CNN-based methods, whose superior performance further bolsters overall recognition performance (See Figure 3-12).

Feed-forward Convolution and RoI Pooling In this work, we leverage the Fast R-CNN (Girshick 2015) network, and shall illustrate how these region-proposals fed CNN-methods are similar in spirit to the FLAIR box-encoding with BoVW or VLAD descriptions. Given an image $\mathcal{I}_k$, and bounding boxes of all object proposal projections $\text{BB}_k^{1:O}$ onto that particular image with pose ξ_k, the R-CNN network first processes the whole image with convolutions (*conv*) and max-pooling layers to produce a convolutional feature map. Then, for each of the proposal region-of-interest (*RoI*), a pooling layer efficiently extracts a fixed-length vector from the convolutional feature map. The fixed-length feature vectors are fed to a sequence of fully connected layers (*fc6*, *fc7*) before they are classified via a soft-max classifier that takes the activated outputs of the final *fc7* layer. While the original Fast R-CNN implementation also simultaneously regresses for the bounding-box positions given the predicted class label, we find our vSLAM map-driven proposals to be sufficiently accurate for bounding box prediction and avoid this additional regression step.

RoI Pooling and FLAIR The RoI pooling layer uses max-pooling to convert features inside a valid bounding box region into a fixed-size feature map. The RoI pooling layer, similar to FLAIR box-encoding described in Section 3.3.3, pools the grid-sampled feature descriptions based on the bounding box dimensions. A RoI window is subdivided into an $H \times W$ grid of sub-windows whose containing features are pooled into, and finally max-pooled in order to recover a fixed-size feature vector. In fact, the RoI pooling layer described in the Fast R-CNN architecture, is

similar to the spatial-pyramid pooling (He et al. 2014; van de Sande et al. 2014) previously described (Figure 3-11) where only a single pyramid level is considered.

Model fine-tuning Most state-of-art object detectors including Fast-RCNN (Girshick 2015) are trained on publicly available object detection datasets such as the Pascal VOC07, 2010 (Everingham et al. 2010) and 2012 (Everingham et al.). While these models and parameters are trained on sufficiently large datasets with special considerations for avoiding over-fitting, they inevitably learn characteristics of the original dataset that may not necessarily transfer to another data domain. This has been evidenced through various works (Ganin and Lempitsky 2015; Glorot et al. 2011; Khosla et al. 2012; Oquab et al. 2014) under the umbrella term of *domain-adaptation*. Typically, they require an additional model fine-tuning step on the limited, representative dataset in order to ensure strong model performance. We take this approach, and further fine-tune the fully connected layers (*fc6*, *fc7*) in the Fast R-CNN using the more representative UW-RGBD dataset (v2). All the pre-trained weights (trained on Pascal VOC 2007) in the network except for the *fc* layers are kept fixed during the model fine-tuning step.

3.3.5 Multi-view Object Recognition

In the case of cluttered environments, all the relevant objects may be visible from only a subset of the views. Furthermore, classifying partially occluded views of an object can potentially harm the overall recognition performance. While it is desirable that partial views of objects are also usefully incorporated into the recognition pipeline, in this work, we are interested in ensuring that the side-effects from classifying partial-views are minimized.

Object Visibility We introduce a *visibility set* such that an object proposal's evidence is only aggregated over the subset of views, thereby ensuring that occluded views are not accidentally mis-classified. This added advantage in spatially-aware systems allows for contextually incorporating measurements, further bolstering recognition performance. For each object proposal $o^j \in \mathcal{O}$, we define a visibility set $\mathcal{V}^j$ that contains the subset of keyframes having an un-occluded view of the proposal for classification purposes (Equation 3.4). First, we define $\text{BB}(\text{proj}_{\xi_k}(o^j))$[2] as the bounding-box projection of object o^j onto view ξ_k, via the image-projection function $\text{proj}_{\xi_k}(\cdot)$[3]. For brevity, we shall refer to $\text{BB}(\text{proj}_{\xi_k}(o^j))$ as BB^j_k.

[2]BB: Refers to the min-max bounds of a 2D point set in the image.

[3]$\text{proj}_{\xi_k}(o^j)$: Refers to the 2D image-projection (onto the view ξ_k) of a set of 3D points in o^j

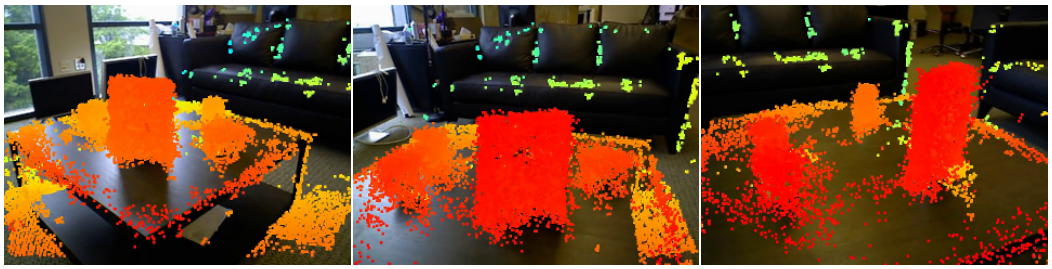


Figure 3-13: **Z-buffering for occlusion handling ►** The underlying semi-dense map reconstructed provides occlusion handling capabilities via z-buffering. This is subsequently used in the object evidence aggregation step in order to avoid mis-identifying object proposals when they are occluded by other objects.

We determine the subset of visible keyframes $\mathcal{V}^j \subseteq \mathcal{K}$ for each proposal o^j based on two criteria (each keyframe is indexed by k):

$$\mathcal{V}^j = \{(\mathcal{I}_k, \xi_k) \mid \text{if } \mathbb{1}_{\text{v}}^{jk} \text{ and } \mathbb{1}_{\text{d}}^{jk} \quad \forall \quad k \in \{1, \ldots, T_K\}\} \tag{3.4}$$

(i) *Intersection-over-Union* (Equation 3.5): The bounding-box projection, given by BB_k^j has an Intersection-over-Union (IoU) measure of less than τ_{occ} with respect to the bounding-box projection of every other object proposal $\text{BB}_k^m, \forall m \in \mathcal{O} \setminus o^j$

$$\mathbb{1}_{\text{v}}^{jk} = \begin{cases} 1 & \text{if } \text{IoU}(\text{BB}_k^j, \text{BB}_k^m) < \tau_{\text{occ}} \quad \forall \quad m \in \mathcal{O} \setminus o^j, \quad \text{and} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

(ii) *Z-buffer depth test* (Equation 3.6): Only the object proposal and corresponding bounding box with the least depth with respect to the view ξ_k is considered, while the rest are ignored. This is otherwise referred to as *z-buffer depth test*, and is a standard procedure in modern graphics renderers for checking visibility of an object.

$$\mathbb{1}_{\text{d}}^{jk} = \begin{cases} 1 & \text{if } d(\text{proj}_{\xi_k}(o^j)) < d(\text{proj}_{\xi_k}(o^m)) \quad \forall \quad o^m \in \mathcal{O} \setminus o^j, \quad \text{and} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

Occlusion-aware evidence aggregation Once the visibility sets are identified for each of the object proposals, we can evaluate our detector only on the subset of views that are identified to be occlusion-free. Subsequently, for each object proposal o^j, the corresponding bounding-box proposals BB_k^j for the k^{th} visible keyframe are described using either FLAIR-VLAD or R-CNN, and denoted by Ψ_k^j.

Thus, for each of the object proposals o^j, we evaluate our detector only on each of the keyframes in the visibility set $\mathcal{V}^j$. Assuming a uniform prior over the C class

Figure 3-14: **Quasi-depth estimation in scale-ambiguous maps** ▶ During the object evidence aggregation step, object proposals $\mathcal{O}^j \in \mathcal{O}$ in the reconstructed scale-ambiguous map $\mathcal{M}$ are projected on to each of the keyframes $\mathcal{K}$, in order to identify objects that may be potentially occluded in that particular view. The median depths (scaled arbitrarily, but consistently across all proposals) of each object proposal is displayed in white.

labels, and that the features Ψ_k are conditionally independent given the class label y, the maximum-likelihood estimate (MLE) reduces to:

$$\hat{y}^j_{MLE} = \underset{y \in \{1,\ldots,C\}}{\operatorname{argmax}} \prod_{k \in \mathcal{V}^j} p(\Psi^j_k \mid Y = y) \tag{3.7}$$

$$= \underset{y \in \{1,\ldots,C\}}{\operatorname{argmax}} \sum_{k \in \mathcal{V}^j} \log p(\Psi^j_k \mid Y = y) \tag{3.8}$$

Thus, the resulting MLE class label of an object proposal o^j is simply the class that corresponds to having the largest sum of log-likelihoods of the class conditional probabilities, estimated for each of their $|\mathcal{V}^j|$ observable viewpoints. Again, we remind the reader that we take advantage of the keyframe selection strategy to evaluate our detector only on an *informative* subset of the vantage points in the scene, and further take leverage of the spatially-aware visibility checks to selectively evaluate occlusion-free object proposals. This property significantly reduces the computational complexity of the overall recognition pipeline, while maintaining strong recognition performance by aggregating object evidence across multiple views.

3.3.6 SLAM-aware, Few-shot Object Learning

One of the primary advantages of maintaining SLAM estimates (as keyframes, and the corresponding scene points) is that they can act as a powerful correspondence-engine for data association purposes. We leveraged this property earlier to bolster recognition performance, via object evidence aggregation (Section 3.3.5). Recovering data associations robustly between views can be challenging; however, they can be particularly useful in recognition tasks such as few-shot visual object learning (Fe-Fei et al. 2003; Fei-Fei et al. 2006; Hariharan and Girshick 2016). In few-

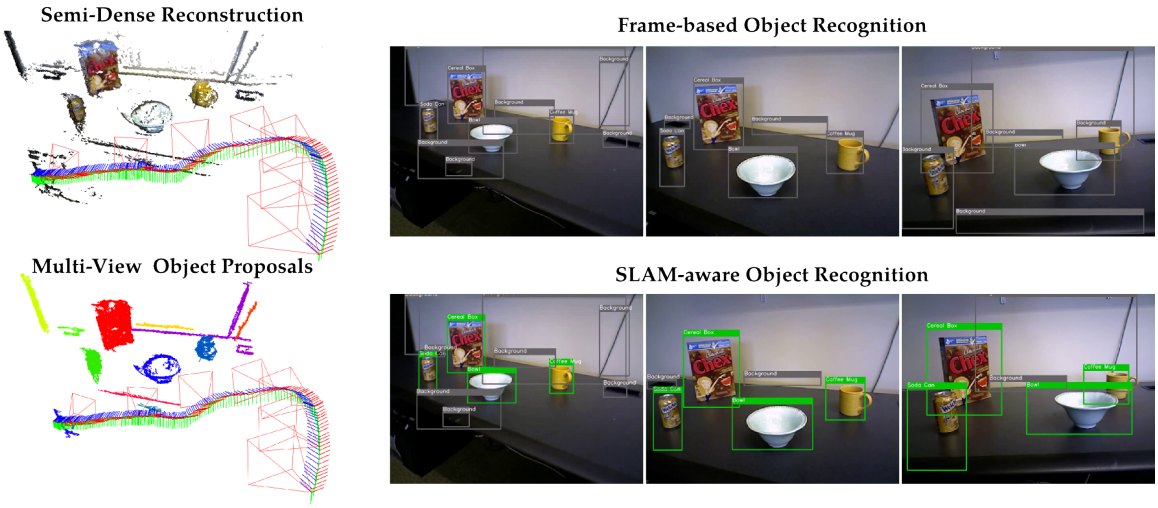


Figure 3-15: **SLAM-oblivious vs. SLAM-aware object detection ►** Illustration of the recognition capabilities of our proposed SLAM-aware object recognition system. Each of the object categories are detected every frame, and their evidence is aggregated across the entire sequence through the set of object hypothesis. In frame-based object recognition, predictions are made on an individual image basis (shown in gray). In SLAM-aware recognition, the predictions are aggregated across all frames in the image sequence to provide robust recognition performance. The green boxes indicate correctly classified object labels, and the gray boxes indicate background object labels. Figure is best viewed in electronic form.

shot recognition, the system is trained on considerably fewer training examples, typically in the tens of samples per category. Despite the limited information provided, few-shot recognition solutions are still able to perform considerably well with a relatively small penalty in overall accuracy. These solutions can be powerful especially when they do not require exhaustive datasets with expensive and tedious ground truth labeling. However, it can be especially difficult to learn from a minimal set of examples without any strong model assumptions (Fe-Fei et al. 2003; Fei-Fei et al. 2006).

Alternatively, data association can be particularly useful in such cases where the same sets of objects may be visible from other viewpoints. Similar to how we leveraged SLAM knowledge to better inform object recognition, we show that the very same SLAM-aware mechanism can be utilized to enable few-shot object learning. By bootstrapping a minimal set of labels from each object category, the 3D bounding volume of the labeled object can be estimated by back-projection, before it is projected onto each of the keyframe views within the same SLAM session. Using the same occlusion-handling procedure described earlier, we only consider projected bounding hulls where more than 80% of its area is un-occluded. With this simple, yet powerful labeling strategy, we are able to train on a fraction of training examples per SLAM session, avoiding the need for tedious ground truth labeling

requirements. We refer to this approach as SLAM-aware, few-shot object learning and present our findings in Section 3.4.2.

3.4 Experiments and Results

In this section, we evaluate the proposed SLAM-aware object recognition method. In our experiments, we extensively evaluate our SLAM-aware recognition system on the popular UW RGB-D Dataset (v2)(Lai et al. 2011; 2014). We compare against the RGB-D based object recognition solutions proposed by Lai et al. (2012) and (Georgakis et al. 2016), that utilize full map and camera location information for improved recognition performance. The UW RGB-D dataset contains a total 51 object categories, however, in order to maintain a fair comparison, we consider the same set of 5 objects as noted in (Lai et al. 2012). With CNN-based methods setting the state-of-the-art in object detection in the past few years (Girshick 2015; Redmon et al. 2016; Ren et al.; 2015), we show that our proposed solution extends easily to incorporate them.

3.4.1 SLAM-Aware Object Recognition Performance

We train and evaluate our system on the UW RGB-D Scene Dataset (Lai et al. 2011; 2014), providing mean-Average Precision (mAP) estimates (see Table 3.1) for the object recognition task and compare against existing methods (Georgakis et al. 2016; Lai et al. 2012). We report our results for detectors trained using both VLAD-FLAIR, and Fast R-CNN in Table 3.1. For visualization purposes, we only show qualitative results using the Fast-RCNN detector. We split our experiments into two categories:

(i) Single-View recognition performance: First, we evaluate the recognition performance of our proposed system on each of the scenes in the UW-RGB-D Scene Dataset on a per-frame basis, detecting and classifying objects that occur every 5 frames in each scene (as done in (Lai et al. 2012)). Each object category is trained from images in the Object Dataset, that includes several viewpoints of object instances with their corresponding mask, and category information. Using training parameters identical to the previous experiment, we achieve a performance of 81.5 mAP (using VLAD-FLAIR, and 88.5 mAP using Fast-RCNN) as compared to the detector performance of 61.7 mAP reported in Lai et al. (2012). Recognition is done

Method	View(s)	Input	Precision/Recall						
			Bowl	Cap	Cereal Box	Coffee Mug	Soda Can	Background	Overall
DetOnly	Single	RGB	46.9/90.7	54.1/90.5	76.1/90.7	42.7/74.1	51.6/87.4	98.8/93.9	61.7/87.9
Det3DMRF	Multiple	RGB-D	91.5/85.1	90.5/91.4	93.6/94.9	90.0/75.1	81.5/87.4	99.0/99.1	91.0/88.8
HMP2D+3D	Multiple	RGB-D	97.0/89.1	82.7/99.0	96.2/99.3	81.0/92.6	97.7/98.0	95.8/95.0	90.9/95.6
(Georgakis et al. 2016)	Single	RGB-D	70.7/56.8	87.2/49.0	84.6/83.3	83.7/34.3	85.6/55.6	89.0/98.1	83.5/62.8
(Georgakis et al. 2016)	Multiple	RGB-D	92.7/89.8	96.9/81.0	87.4/97.8	88.4/87.0	86.7/84.2	97.3/98.0	91.6/89.6
Ours (VLAD-FLAIR)	Single	RGB	88.6/71.6	85.2/62.0	83.8/75.4	70.8/50.8	78.3/42.0	95.0/90.0	81.5/59.4
Ours (VLAD-FLAIR)	Multiple	RGB	88.7/70.2	99.4/72.0	95.6/84.3	80.1/64.1	89.1/75.6	96.6/96.8	89.8/72.0
Ours (Fast-RCNN)	Single	RGB	91.0/68.3	92.0/51.8	77.6/51.9	54.7/70.0	67.9/71.2	92.5/95.0	88.5/88.0
Ours (Fast-RCNN)	Multiple	RGB	93.4/71.6	99.1/33.2	100.0/82.4	72.0/83.9	82.9/81.1	92.2/96.6	91.1/90.7

Table 3.1: **SLAM-aware object classification results on UW-RGBD Dataset ►** Object classification results using the UW RGB-D Scene Dataset (Lai et al. 2011; 2014), providing mean-Average Precision (mAP) estimates for both Single-View, and Multi-View object recognition approaches (using VLAD-FLAIR and Fast-RCNN as detectors). We compare against existing methods DetOnly, Det3DMRF(Lai et al. 2012), HMP2D+3D (Lai et al. 2014) and (Georgakis et al. 2016) that use RGB-D information instead of relying only on RGB images, in our case. Recognition for the single-view approach is done on a *per-frame* basis, where prediction performance is averaged across all frames across all scenes. For the multi-view approach, recognition is done on a *per-scene* basis, where prediction performance is averaged across all scenes.

on a per-image basis, and averaged across all test images for reporting. Figure 3-16 shows the recognition results of our system on a per-frame basis. We ignore regions labeled as background in the figure for clarity and only report the correct and incorrect predictions in green and red respectively.

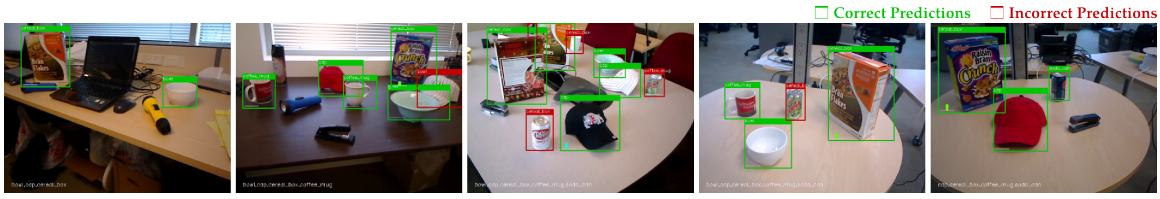
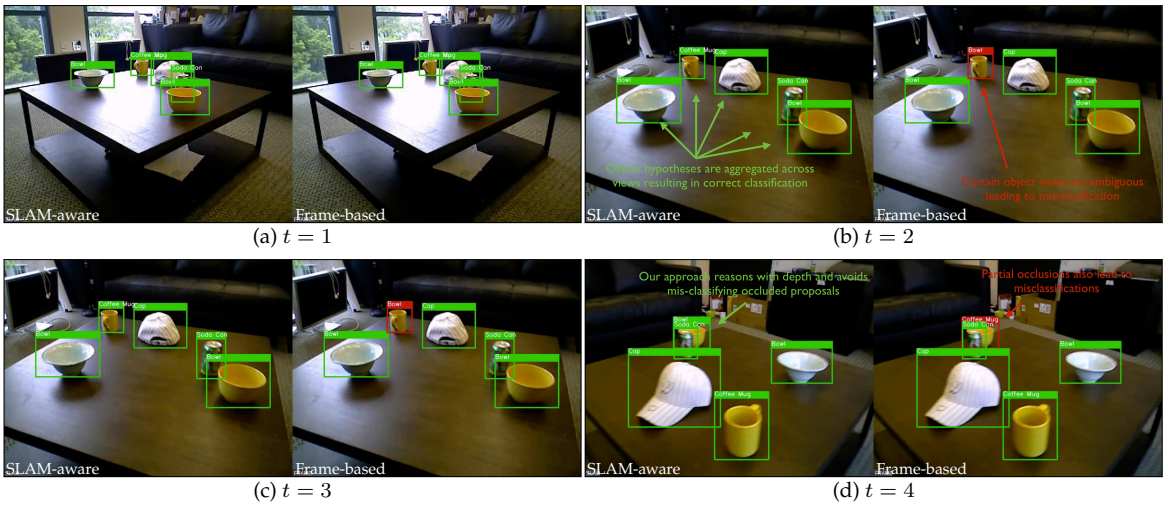


Figure 3-16: **Pitfalls of frame-based object detection ►** Illustration of **per-frame** detection results provided by our object recognition system that is *intentionally SLAM-oblivious* (for comparison purposes only). Object recognition evidence is not aggregated across all frames, and detections are performed on a frame-by-frame basis. Only detections having corresponding ground truth labels are shown. Figure is best viewed in electronic form.

(ii) Multi-View recognition performance: In this section, we investigate the performance of a SLAM-aware object recognition system. We compare this to a SLAM-oblivious object detector described previously, and evaluate using ground truth provided. Using the poses ξ and reconstructed map $\mathcal{M}$, multi-view object candidates are proposed and projected onto each of the images for each scene sequence. Using the candidates provided as input to the recognition system, the system predicts the likelihood and corresponding category of an object (including background) contained in a candidate bounding box. For each of the objects $o^j \in \mathcal{O}$

proposed, the summed log-likelihood is computed (as in Eqn. 3.8) to estimate the most likely object category over all the keyframes for a particular scene sequence. We achieve 89.8 mAP recognition performance on the 5 objects in each of the scenes in (Lai et al. 2014) that was successfully reconstructed by the ORB-SLAM-based semi-dense mapping system. Using Fast-RCNN, the recognition performance can be further improved to 91.1 mAP. Figures 3-15 and 3-17 illustrate the capabilities of the proposed system in providing robust recognition performance by taking advantage of the monocular visual SLAM-backend. Figure 3-19 illustrates the average precision-recall performance on the UW RGB-D dataset, comparing the classical frame-based and our SLAM-aware approach. As expected, with additional object viewpoints, our proposed SLAM-aware solution predicts with improved precision and recall. In comparison to that of HMP2D+3D (Lai et al. 2014), they achieve only slightly higher overall recognition performance of 90.9 mAP, as their recognition pipeline takes advantage of the RGB and depth input to improve overall scene reconstruction.

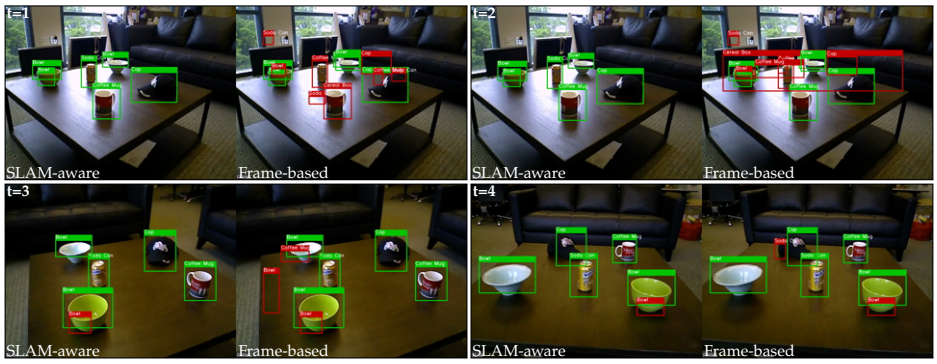


(a) $t = 1$ (b) $t = 2$

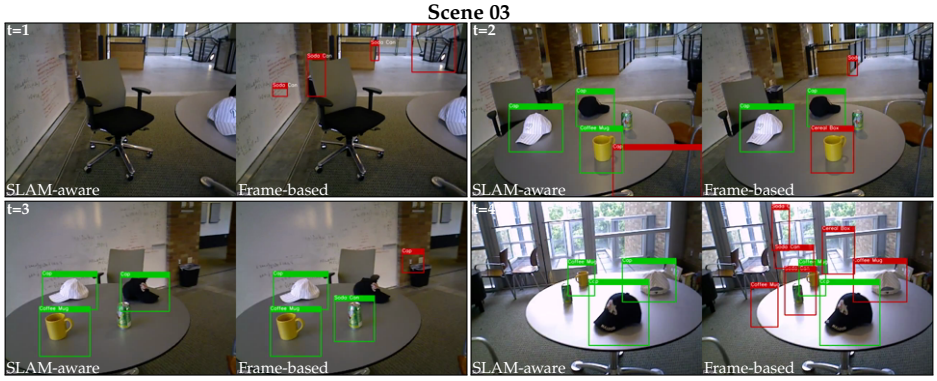
(c) $t = 3$ (d) $t = 4$

Figure 3-17: **Handling occlusions and ambiguous object classification** ▶ Some views in the scene may be ambiguous for the object detector, while some other views may be occluding based on the scene. While most traditional solutions only depend on a single-view, it is imperative to understand the semantics of the world in a spatially-aware manner. This allows our proposed SLAM-aware method to reliably reason about objects that may be partially occluded (the cap and soda can are occluded at $t = 1$), or reason about objects that may be hard to disambiguate for the detector from certain views (e.g. the cup may be mis-identified as a bowl at $t = 2$).

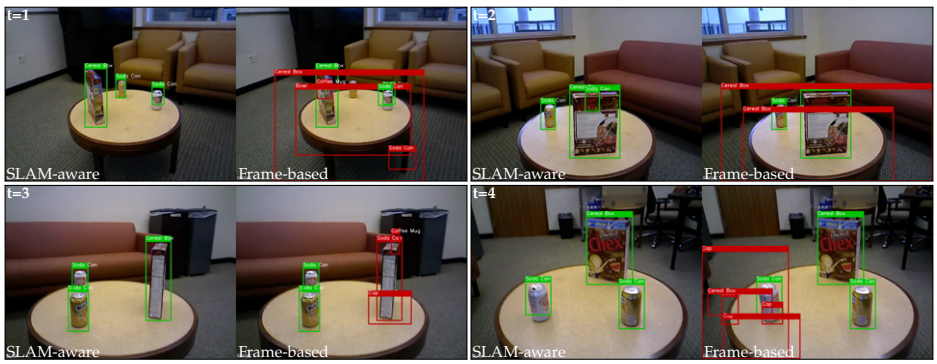
Qualitative results Through qualitative examples (in Figure 3-18), we address a few characteristic differences in frame-based and SLAM-aware recognition systems. In *Scene 03*, the frame-based method occasionally mis-identifies proposals in the scene, while it classifies correctly at other times. In our SLAM-aware solution, these classifications are consistent both spatially and temporally. In one of the



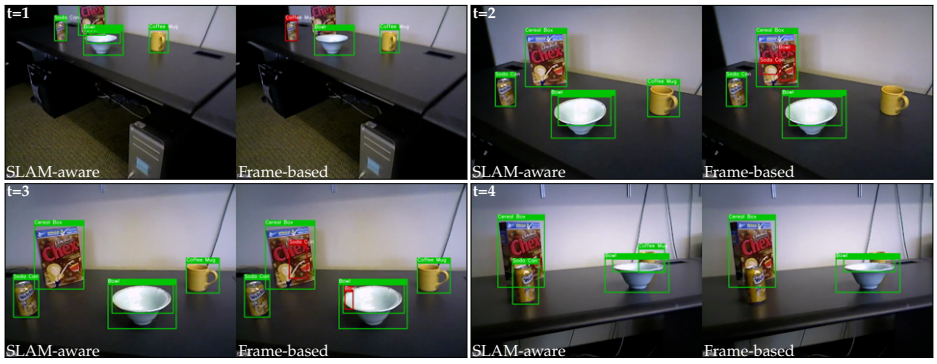
Scene 03



Scene 08



Scene 11



Scene 13

Figure 3-18: **Qualitative results of SLAM-aware recognition with Fast-RCNN** ► More illustrations of the superior performance of the SLAM-aware object recognition in scenarios of ambiguity and occlusions.

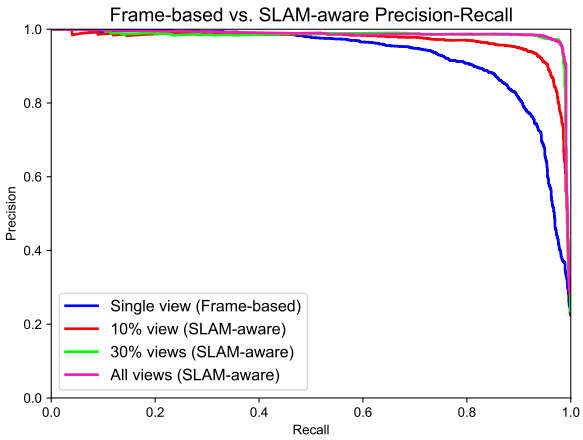


Figure 3-19: **SLAM-aware recognition performance using Fast-RCNN** ► Performance comparison via precision-recall for the Frame-based vs. SLAM-aware object recognition. As expected, the performance of our proposed SLAM-aware solution increases with more recognition evidence is aggregated across multiple viewpoints.

frames at $t = 3$, the correctly classified bounding-box for the green bowl is evaluated to be a failure since the IoU metric is not satisfied. In *Scene 08*, however, the green soda can is not recognized correctly by our method. This is since the bounding-box is fairly loose around it, classifying most distant views to it as background (from other vantage points). This results in the most-likely label to be of the background class, and thereby resulting in an incorrect classification. In *Scene 11*, some of the views of the coffee table are incorrectly identified (class:*cereal box*), as a part of the bounding box includes a cereal box. This intermittent classification in frame-based methods is easily identified and remedied in SLAM-aware systems when the classification probabilities are aggregated in a sound manner. Finally, in *Scene 13*, we show another example of the superior classification performance of our SLAM-aware method against frame-based methods that may occasionally identify false positives, and false negatives.

3.4.2 Few-shot Object Learning

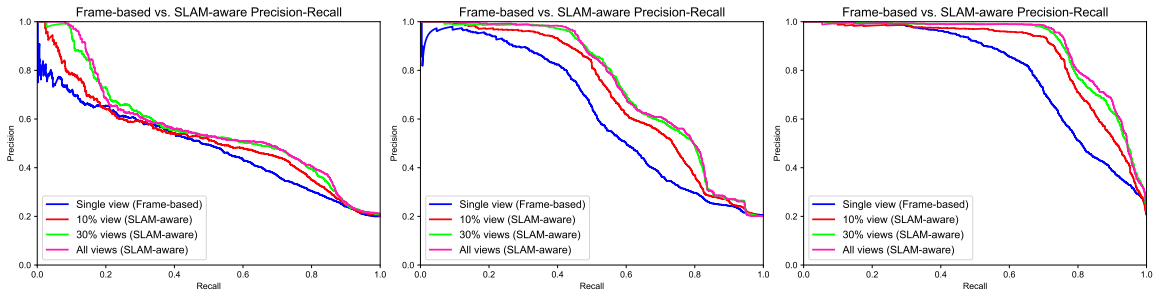
We evaluate the proposed SLAM-aware, few-shot object learning with two sets of experiments. In the first experiment, we randomize the few-shot object learning procedure, where the training set is curated with only a randomly sub-sampled set of ground truth examples. The resulting object detector learned from the reduced training set is used to perform SLAM-supported recognition as described earlier. We refer to this as *Randomized few-shot object learning with SLAM-aware recognition*. In the second experiment, we leverage the strong correspondences that SLAM pro-

vides to robustly propagate ground truth labels for objects in a scene. This allows our approach to significantly improve recognition performance with even fewer training examples. Coupled with the SLAM-supported evidence aggregation step, we refer to this approach as *SLAM-aware few-shot object learning with SLAM-aware recognition*.

For the few-shot experiments, training is performed using the ground truth labels in all scenes from the UW-RGBD Dataset (v2), except for the held-out test scene. In the SLAM-aware few-shot learning case, the training can afford to take advantage of the full keyframe-based vSLAM solution to make strong data associations for robust label propagation. In subsequent experiments, we use the pre-trained Fast-RCNN model (Girshick 2015) (CaffeNet, `vgg_cnn_m_1024`) as the black-box object classifier, and fine-tune the fully-connected layers (*fc6*, *fc7*, *fc8*) for our dataset. The model parameters are estimated using grid-search with a cross-validation step, and a train/test split of 70-30 to mitigate over-fitting. In Table 3.2, we compare the detectors learned via our proposed SLAM-aware few-shot learning against the randomized few-shot case, and clearly validate that a SLAM-aware system can be beneficial in both training and testing/deployment purposes.

Randomized few-shot object learning with SLAM-aware recognition In the randomized few-shot object learning, we investigate poorly trained/calibrated object classifiers by deliberately limiting the amount of ground truth labels available during model training. By only providing only a subset of the object views during training, the randomized few-shot training is not able to capture the full extent of an object's variance in its feature space, thereby resulting in a poorly calibrated object classifier. In such situations, certain views of an object may be identified as more confident than others, potentially resulting in mis-classification of object views with low classification scores.

The SLAM-aware view aggregation addresses this particular concern, where certain confident views of an object can help disambiguate the object label in other less-confident vantages. In Figure 3-20, we compare few-shot recognition performance for 2, 5 and 10 training examples per category (referred to as 2-shot, 5-shot and 10-shot respectively). As expected, considering more training examples tends to improve overall recognition performance. Despite poorly calibrated classifiers with randomized few-shot training, we notice that the SLAM-aware solution significantly bolsters recognition performance over the single-view, frame-based methods in all of the few-shot scenarios illustrated in Figure 3-21. Additionally, we show that with an increasing fraction of keyframe views considered in the view-



(a) 2-shot **(b) 5-shot** **(c) 10-shot**

Figure 3-20: **SLAM-aware recognition with randomized few-shot training ▶** The figures illustrate the performance results of our proposed SLAM-aware recognition solution when the detector is trained on a few examples (Few-shot training). The performance of our SLAM-aware method considerably outperforms frame-based methods, despite poorly trained classifiers in the 2, 5 and 10-shot cases. Furthermore, our approach seamlessly provides improved accuracy with more views considered for aggregation. Here, *10% views* implies that only a tenth of the keyframes are used for inferring the object label.

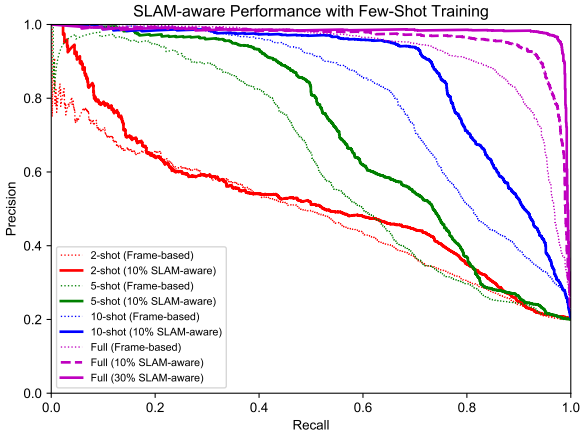


Figure 3-21: **Randomized few-shot training with increasing examples considered ▶** We illustrate the performance of our proposed SLAM-aware recognition solution when the detector is trained on a randomized subset of the ground truth labels. Despite being trained on incomplete information, the SLAM-supported recognition solution can considerably bolster recognition performance by aggregating evidence across multiple-views. In the above experiments, we also investigate the SLAM-aware solution where only a fraction of keyframe views are considered (10%, 30%) to infer the object label. With increasing views considered, our proposed method is able to seamlessly provide better recognition performance.

aggregation step, our approach is able to seamlessly provide better recognition performance. In Figures 3-21 and 3-20, we notice a considerable bump in performance with the first 10% of views considered for SLAM-aware view-aggregation, before we observe diminishing returns with additional views.

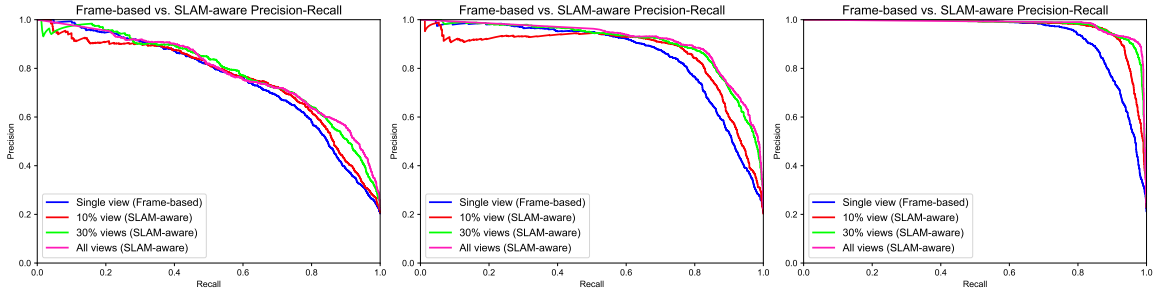
SLAM-aware few-shot object learning with SLAM-aware recognition Given limited labeled information, few-shot object learning can especially benefit from additional constraints or assumptions as it allows to reason over the unlabeled data. By leveraging accurate data associations between various keyframe views, a SLAM-

Method	Frame-based Recognition mAP / Recall / F1-score	SLAM-aware Recognition mAP / Recall / F1-score
2-shot (Randomized)	80.5 / 63.4 / 69.7	83.1 / 74.8 / 77.1
5-shot (Randomized)	76.0 / 72.6 / 73.7	81.6 / 80.9 / 80.5
10-shot (Randomized)	79.6 / 74.5 / 76.0	81.6 / 82.2 / 81.5
20-shot (Randomized)	85.9 / 80.5 / 82.2	91.0 / 89.8 / 90.2
1-shot (SLAM-aware)	85.3 / 85.2 / 82.6	87.9 / 87.0 / 84.3
2-shot (SLAM-aware)	87.4 / 87.6 / 86.3	89.6 / 89.0 / 87.3
4-shot (SLAM-aware)	89.6 / 89.3 / 89.2	90.6 / 90.8 / 90.5

Table 3.2: **Comparison of SLAM-aware and randomized few-shot object learning ▶** With significantly fewer examples, our SLAM-aware few-shot approach is able to achieve strong performance compared to randomized few-shot training. Here, the Fast-RCNN detector is fine-tuned only on a few examples (2, 5, 10, 20 samples per class). Additionally, the performance of our SLAM-aware recognition (via multi-view aggregation) considerably outperforms frame-based methods, despite poorly trained classifiers in the few-shot cases.

aware system can provide useful additional constraints to further propagate the user-provided labels to unlabeled data. Furthermore, we expect SLAM-aware systems to be especially suited to this task of object learning, as they allow the propagation of ground truth labels to considerably new vantage points that the object classifier is uncertain about. The labeled ground truth bounding box allows us to identify the most relevant object proposal within the 3D semi-dense reconstruction, based on their intersection-over-union (IoU) measure. We pick the corresponding 3D object proposal with the maximum IoU score, and subsequently project the bounding volume onto each of the individual keyframe views recovered during the keyframe-based Visual SLAM procedure. Similar to the view aggregation step, we ensure that the projected bounding boxes are un-occluded after the z-buffer check, with at least 80% of its original area visible. Subsequently, all bounding volume projections onto the T_K keyframes are considered as ground-truth labels for few-shot training purposes. After this step, training follows exactly as the original training procedure described in Section 3.3.4.

As expected, our SLAM-aware few-shot training solution is able to propagate the provided ground truth labels to more keyframe views, thereby aggregating more labeled examples for training purposes. With considerably fewer training examples, we are able to achieve similar recognition performance compared to the randomized few-shot case. Figure 3-22 illustrates the performance of detectors estimated via SLAM-aware few-shot learning. In Figure 3-23, we compare the one-shot, 2-shot, and 4-shot detector and illustrate that we are able to achieve strong recognition performance (89.6% mAP) for the SLAM-aware 4-shot trained case. Additionally, the performance can be further bolstered to (90.6% mAP) with SLAM-aware view aggregation step. Interestingly, we notice that the SLAM-aware, few-



(a) 1-shot **(b) 2-shot** **(b) 4-shot**

Figure 3-22: **SLAM-aware recognition with few-shot SLAM-aware training ►** The figures illustrate the performance results of our proposed SLAM-aware recognition solution when the detector is trained on a few examples (Few-shot training). The performance of our SLAM-aware method considerably outperforms frame-based methods, despite poorly trained classifiers in the 2, 5 and 10-shot cases. Furthermore, approach seamlessly provides improved accuracy with more views considered for aggregation. Here, *10% views* implies that only a tenth of the keyframes are used for inferring the object label.

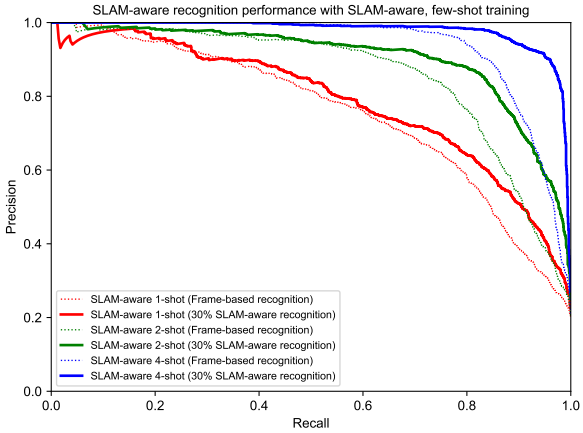


Figure 3-23: **SLAM-aware few-shot training with increasing examples considered ►** We illustrate the performance of our proposed SLAM-aware few-shot training solution. The SLAM-aware few-shot training solution is able to achieve high precision-recall, with significantly fewer training examples considered compared to the randomized few-shot training case.

shot learning approach also shows promising results even in the extreme one-shot training case (Figure 3-22), achieve an mAP of 85.3% in the frame-based evaluation.

3.5 Discussion and Future Work

Our earlier contribution (Pillai and Leonard 2015) has inspired several recent works in SLAM-supported recognition (Bogun et al. 2015; Tateno et al. 2016) and semantic mapping (Dong et al. 2016; Sünderhauf et al. 2016). With the availability of pre-trained CNN-based models, the semantic image understanding landscape has considerably changed. Recognition models are getting considerably stronger to

support object-based representations for SLAM (Bowman et al. 2017; Gálvez-López et al. 2016), thereby opening avenues to truly scalable SLAM solutions that can extend to hundreds of thousands of objects within a scene. We motivate this with an illustration of object-based SLAM (Figures 3-24 and 3-25), where the landmarks in the Visual SLAM formulation are maintained as *semantically identifiable objects*.

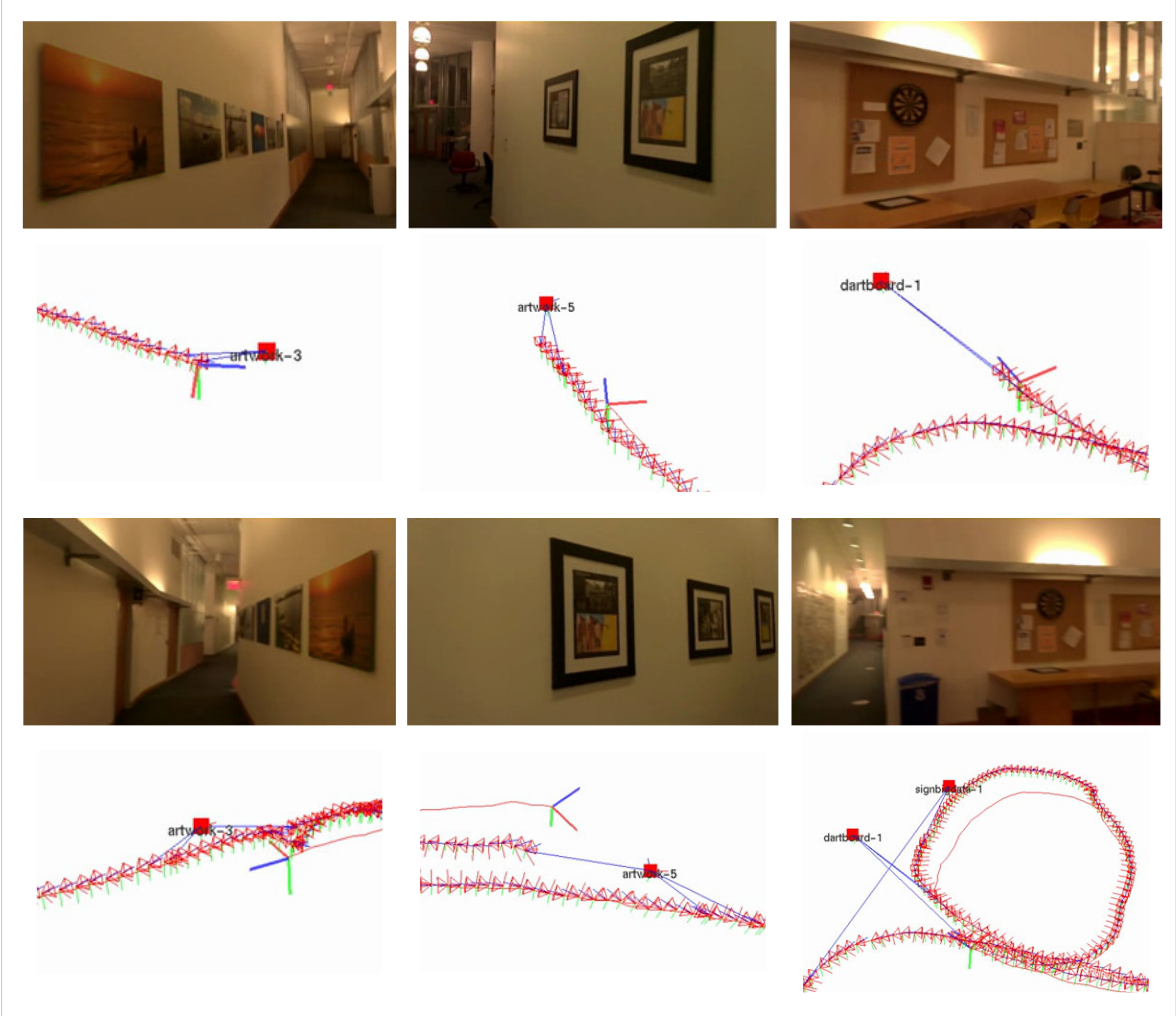


Figure 3-24: **Recognition-Supported SLAM** ▶ We illustrate the concept of Object-based SLAM, where the object recognition cues (object category and instance) are used to establish data-associations across views in a typical Visual SLAM problem. Since the landmark sightings are sparse (but semantically rich in description), the equivalent factor-graph formulation is considerably smaller compared to its Visual-SLAM counterpart. The figure shows various sightings of distinct objects and the corresponding vSLAM solution recovered from incorporating these object instance-level semantic data associations. The solid red line indicates the visual-odometry solution, while the camera frustums (also drawn in red) represent the incremental keyframe-based vSLAM solution using these objects as landmarks.

Recognition-Supported SLAM One particularly interesting and recurring theme in this thesis is the robustness of the visual SLAM solution to provide reliable

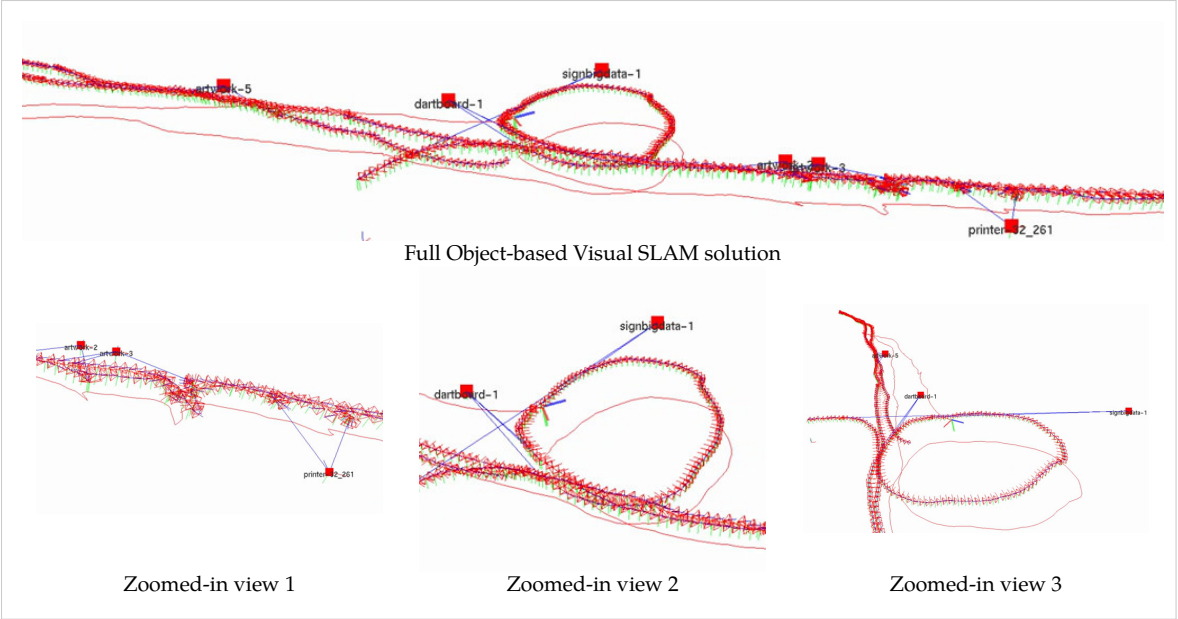


Figure 3-25: **Optimized Visual-SLAM solution via Object-based SLAM ►** Optimized Visual-SLAM solution while using *objects* as landmarks in the visual-SLAM formulation. The visual-odometry solution (shown in solid red) drifts as expected over long durations, but provides useful short-term, fine-grained resolution for pose estimation. The semantic detection of various objects (artwork, dartboard, and signage) provide sparse and rich correspondence information between temporally distant nodes in the pose-graph, thereby constraining the overall drift incurred.

3D landmarks $\mathbf{L}^*$ (map reconstruction) and camera pose estimates $\mathbf{X}^*$. As with any SLAM system, the solutions to SLAM only get better if these optimization back-ends are fed with reliable measurements whose data associations are consistent. While vision-based front-end solutions have long existed to provide reliable measurements to the back-end, they still struggle with difficult problems such as perceptual aliasing (different places having similar appearance can be ambiguous), and appropriately identifying scene saliency (determining importance to certain landmarks over others). Furthermore, existing visual SLAM front-ends have mostly been limited to extracting low-level semantics in the form of bag-of-visual-words descriptions, and do not encode any high-level semantic description of the scene that may be valuable for tackling the perceptual aliasing and saliency concerns. We revisit this problem of recognition-supported SLAM in Chapter 6 in the context of scene recognition and provide compelling results towards bolstering existing visual SLAM front-ends.

3.6 Chapter Summary

In this chapter, we develop a SLAM-aware object-recognition system, that is able to provide robust and scalable recognition performance as compared to classical SLAM-oblivious recognition methods. We leverage some of the recent advancements in semi-dense monocular SLAM to propose objects in the environment, and incorporate efficient feature encoding techniques via VLAD-FLAIR and CNN-based RoI pooling, to provide an improved object recognition solution. Additionally, by maintaining a spatially-cognizant view of the world, we show that a SLAM-aware, few-shot object learning strategy can be especially advantageous to mobile robots that can learn quickly from a minimal set of experiences. With this effective training strategy, we are able to train on a fraction of examples per SLAM session, avoiding the need for tedious and expensive ground truth labeling requirements. With considerably fewer training examples, we are able to achieve similar recognition performance compared to the randomized few-shot case. Furthermore, we notice that the SLAM-aware, few-shot learning approach also shows promising results even in the extreme one-shot training case. Through various experiments, we show that our SLAM-aware monocular recognition solution is competitive with the current state-of-the-art in the RGB-D object recognition literature. We believe that robots equipped with such a monocular system will be able to robustly recognize and accordingly act on objects in their environment, in spite of object clutter and recognition ambiguity inherent from certain object viewpoint angles.

Chapter 4

Map Representations for Vision-Based Navigation

So far in this thesis, we have advocated for the ability to leverage SLAM as a sensor to better inform tasks such as object recognition, that benefit from mobile robots being spatially-cognizant. As robots take advantage of their inherent SLAM capabilities, it is critical to revisit the problem of map representation. Depending on the immediate task at hand, the robot may require geometric maps resolved at various spatial resolutions. Most mapping components contained with Visual-SLAM systems today are tailored towards constructing high-fidelity maps, without much concern for how these representations are used elsewhere by the robot. This leads to severe fragmentation in map representations, with certain modules maintaining discrete voxel-based maps for planning and obstacle avoidance purposes, while other Visual-SLAM modules maintaining larger point-based representations that allow for continuous optimization over the landmarks and camera poses.

Ideally, we would like to minimize redundant map representations across subsystems, and seek a *flexible map representation* that can be directly estimated in SLAM sub-components, while being readily usable in the context of motion planning in mobile robots with little modification. Additionally, we expect this common representation to be especially amenable to planning feedback so that these systems can perform in a resource-constrained and plan-aware setting. With agile mobile robots in context, we consider the mapping problem using stereo-vision, and propose a potential solution towards enabling this tunable map representation goal.

Traditionally, stereo algorithms have focused their efforts on reconstruction quality and have largely avoided prioritizing for run time performance. Robots, on the

other hand, require quick maneuverability and effective computation to observe its immediate environment and perform tasks within it. In this chapter, we propose a high-performance and tunable stereo disparity estimation method, with a peak frame-rate of 120Hz (VGA resolution, on a single CPU-thread), that can potentially enable robots to quickly reconstruct their immediate surroundings and maneuver at high-speeds. Our key contribution is a disparity estimation algorithm that iteratively approximates the scene depth via a piece-wise planar mesh from stereo imagery, with a fast depth validation step for semi-dense reconstruction. The mesh is initially seeded with sparsely matched keypoints, and is recursively tessellated and refined as needed (via a resampling stage), to provide the desired stereo disparity accuracy. The inherent simplicity and speed of our approach, with the ability to tune it to a desired reconstruction quality and run-time performance makes it a compelling solution for applications in high-speed vehicles.

4.1 Introduction

Stereo disparity estimation has been a classical and well-studied problem in computer vision, with applications in several domains including large-scale 3D reconstruction, scene estimation and obstacle avoidance for autonomous driving and flight etc. Most state-of-the-art methods (Žbontar and LeCun 2015) have focused its efforts on improving the reconstruction quality on specific datasets (Geiger et al. 2012; Scharstein and Szeliski 2002), with the obvious trade-off of employing sophisticated and computationally expensive techniques to achieve such results. Some recent methods, including Semi-Global Matching (Hirschmüller 2005), and ELAS (Geiger et al. 2011a), have recognized the necessity for practical stereo matching applications and their real-time requirements. However, none of the state-of-the-art stereo methods today can provide meaningful scene reconstructions in real-time ($\geq$ 25Hz) except for a few FPGA or parallel-processor-based methods (Banz et al. 2010; Gehrig et al. 2009; Honegger et al. 2014). Other methods have achieved high-speed performance by matching fixed disparities, fusing these measurements in a push-broom fashion with a strongly-coupled state estimator (Barry and Tedrake 2015). Most robotics applications, on the other hand, require real-time performance guarantees in order for the robots to make quick decisions and maneuver their immediate environment in an agile fashion. Additionally, as requirements for scene reconstruction vary across robotics applications, existing methods cannot be reconfigured to various accuracy-speed operating regimes.

In this work, we propose a high-performance, iterative stereo matching algorithm, capable of providing semi-dense disparities at a peak frame-rate of 120Hz (see Figure 4-2). An iterative stereo disparity hypothesis and refinement strategy is proposed that provides a tunable iteration parameter to adjust the accuracy-versus-speed trade-off requirement on-the-fly. Through experiments, we show the strong reliability of disparity estimates provided by our system despite the low computational requirements. We provide several evaluation results comparing accuracies against current stereo methods, and provide performance analysis for varied runtime requirements. We validate the performance of our system on both publicly available datasets, and commercially available stereo sensors for comparison. In addition to single view disparity estimates, we show qualitative results of large-scale stereo reconstructions registered via stereo visual odometry, illustrating the consistent stereo disparities our approach provides on a per-frame basis.

4.2 Related Work

Classical stereo matching methods have mostly considered dense reconstructions, and are generally divided into two categories, *local* and *global* methods. The naive approach to stereo matching involves finding corresponding pixels in the left and right images that have similar color or intensity. Since the intrinsics and extrinsics of the stereo cameras are known, the matching search space is limited to the epipolar line with a pre-defined disparity level, assuming a maximum distance observed.

Dense Methods As one may expect, the above formulation results in a noisy disparity map, due to the high pixel-level ambiguity in matching. This is addressed by matching fixed size windows instead, reducing the noise and inherent ambiguity in the stereo imagery. Additionally, the resulting disparity is smoothed, allowing neighboring pixels to have similar disparities. Despite several advances in adaptive-supports, slanted window matching and edge-preserving filtering approaches (Bleyer and Breiteneder 2013), local methods suffer from being unable to estimate disparities at low-textured regions.

For the past decade, global methods have dominated stereo benchmarks (Geiger et al. 2012; Scharstein and Szeliski 2002). They differ from local methods in that their smoothness regularization assumptions are no longer limited to a fixed window size, but extend throughout the image. Typically, the disparity estimation is

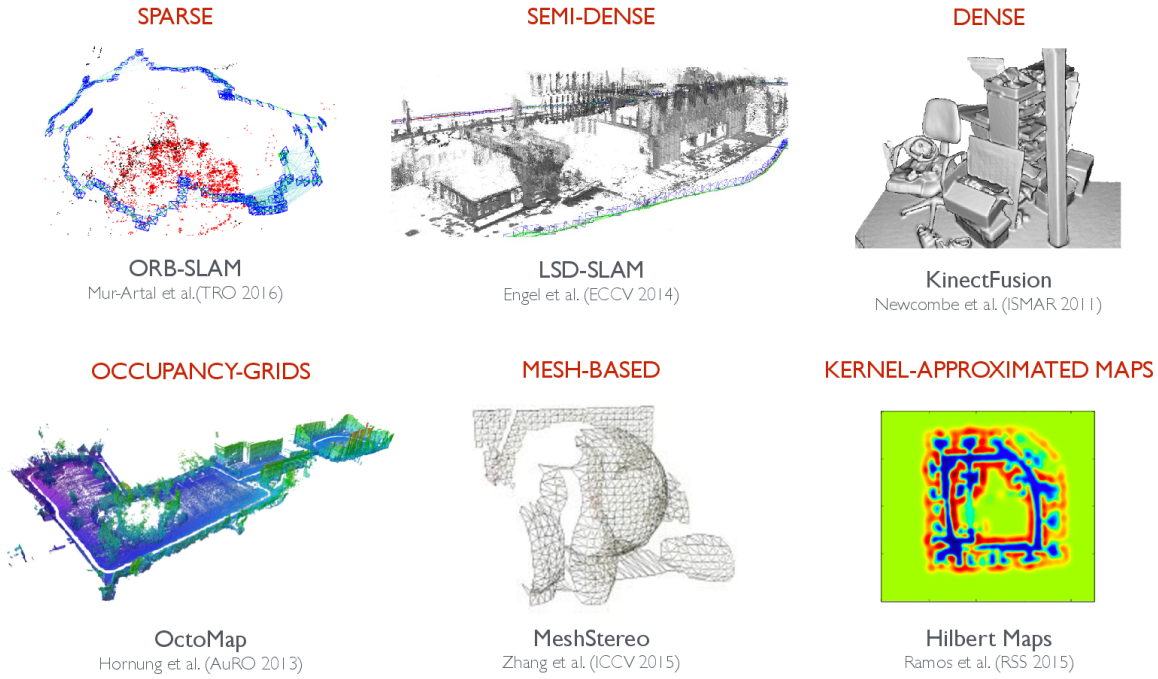


Figure 4-1: **Map representations** ▶ Illustrated are the various state-of-the-art map representations developed in the past decade. While these representations are powerful in structurally or geometrically describing the scene, they each reconstruct the scene at a pre-specified density and do not admit the ability to tune for scene reconstruction density on-the-fly. All the figures above are drawn from recent works (Engel et al. 2014; Hornung et al. 2013; Mur-Artal et al. 2015; Newcombe et al. 2011; Ramos and Ott 2016; Zhang et al. 2015).

modeled as an energy minimization, given by:

$$E(D) = \sum_{p \in I_l} c(p, p - d_p) + \lambda \sum_{\{p,q\} \in \mathcal{N}} s(d_p, d_q) \tag{4.1}$$

where $c(p, p - d_p)$ is the pixel matching cost for a disparity level d_p, $s(d_p, d_q)$ is the smoothness regularization or penalty enforced between pixels p and q that are neighbors defined by $\mathcal{N}$. The above energy minimization formulation allows several optimization strategies to be employed including (i) graph-cuts (ii) belief-propagation (iii) dynamic programming. For a more thorough description of state-of-the-art stereo matching, we refer the reader to (Bleyer and Breiteneder 2013).

Sparse and Semi-Dense Methods Sparse stereo matching methods have been prevalent in robotics applications primarily due to their low-computational complexity (Schauwecker et al. 2012). These methods, including monocular keypoint-based SLAM techniques, have been combined with tessellation or meshing techniques to represent the scene as piece-wise planar (Concha and Civera 2015), making it a fairly rich representation for navigation and scene reconstruction purposes

with a significantly low memory footprint.

Recently, there has been an increased interest in semi-dense representations for mapping, navigation (Engel et al. 2014; Mur-Artal and Tardos 2015; Ramalingam et al. 2015; Veksler 2002) and object detection (Pillai and Leonard 2015). Qualitatively, these semi-dense methods can be a compelling middle-ground, between dense stereo and sparse stereo matching methods, potentially paving the way to newer representations for navigation and reconstruction. LSD-SLAM (Engel et al. 2014), has recently shown large-scale 3D reconstructions by fusing the depth estimates for high-gradient pixels from short and wide-baseline frames in monocular videos, without the use of any interest point matches. However, monocular methods suffer from the well-known scale-drift problem (corrected using an IMU), and rely on the availability of several images to provide metrically accurate reconstructions. Recently, we proposed a semi-dense stereo reconstruction of high gradient pixels using a Line-Sweep algorithm (Ramalingam et al. 2015), which uses cross-ratio constraints on locally planar region. Our method relies on Delaunay triangulation and support point re-sampling, leading to better accuracy and improved computational performance. Furthermore, our method can reconstruct heavily occluding objects like poles, which will be challenging for Line-sweep.

Depth-priors and Plane-based Stereo Our work closely relates to that of ELAS (Geiger et al. 2011a) that takes a generative approach, using tessellated support points from sparse stereo matching as a depth prior to enable efficient sampling of disparities in a dense fashion. Most recently, MeshStereo (Zhang et al. 2015) has been proposed, where the global stereo model is designed for view interpolation via a similar 3D triangular mesh. The authors model the difficult depth discontinuity problem as a two-layer MRF, where the upper layer models the splitting of depth discontinuities, while the lower layer regularizes the depths via a region-based optimization. In this work, we take a discriminative approach to stereo matching, and continue to maintain the piece-wise planar assumption while re-tessellating poorly reconstructed regions in the interpolated disparity image that correspond to having a high matching cost. Furthermore, we propose an iterative method that continues to re-tessellate and approximate complex surfaces with more piece-wise planar regions, with every additional iteration.

Similar to Patch-Match Stereo (Bleyer et al. 2011), our method implicitly computes disparities with sub-pixel precision, without the need for an additional post-processing step (Yang et al. 2007) that fits a parabolic curve within the cost volume. As duly noted in (Bleyer and Breiteneder 2013), parabolic fitting leads to noisy sub-

pixel estimation across heavily slanted surfaces. We do note that our approach is reminiscent of plane-sweeping algorithms that include fronto-parallel and slanted windows to their label space for improved disparity estimation along varied surfaces (Gallup et al. 2007), however, we draw candidate planes and disparities from the tessellations constructed with sparse keypoint-based stereo matches that in turn reduces the search space drastically.

High-speed Stereo Matching To the best of our knowledge, we are unaware of any semi-dense stereo method that can compute disparities at speeds of ≥ 100Hz, without the use of GPUs, FPGAs or other specialized-hardware. We consider disparity estimation for the approximate piece-wise planar case, as this representation can be especially useful in robotics applications where obstacles are to be observed and avoided in real-time. We propose an iterative stereo matching method, that maintains a spatially-adaptive piece-wise planar representation, significantly speeding up stereo disparity estimation by a factor of 32x compared to popular stereo implementations (Hirschmüller 2005), while providing sufficiently accurate disparity estimates.

4.3 High-Performance and Tunable Stereo Reconstruction

This section introduces the algorithmic components of our method (see Alg. 2). We propose a tunable (and iterative) stereo algorithm that consists of four key steps: (i) Depth prior construction from Delaunay triangulation of sparse key-point stereo matches; (ii) Disparity interpolation using piece-wise planar constraint imposed by the tessellation with known depths; (iii) Cost evaluation step that validates interpolated disparities based on matching cost threshold; and (iv) Re-sampling stage that establishes new support points from previously validated regions and via dense epipolar search. The newly added support points are re-tessellated and interpolated to hypothesize new candidate planes in an iterative process. Since we are particularly interested in collision-prone obstacles and map structure in the immediate environment, we focus on estimating the piece-wise planar reconstruction as an approximation to the scene, and infer stereo disparities in a semi-dense fashion from this underlying representation. Unless otherwise noted, we consider and perform all operations on only a subset of image pixels that have high image gradients $\Omega_I \subset \Omega$, and avoid reconstructing non-textured regions in this work.

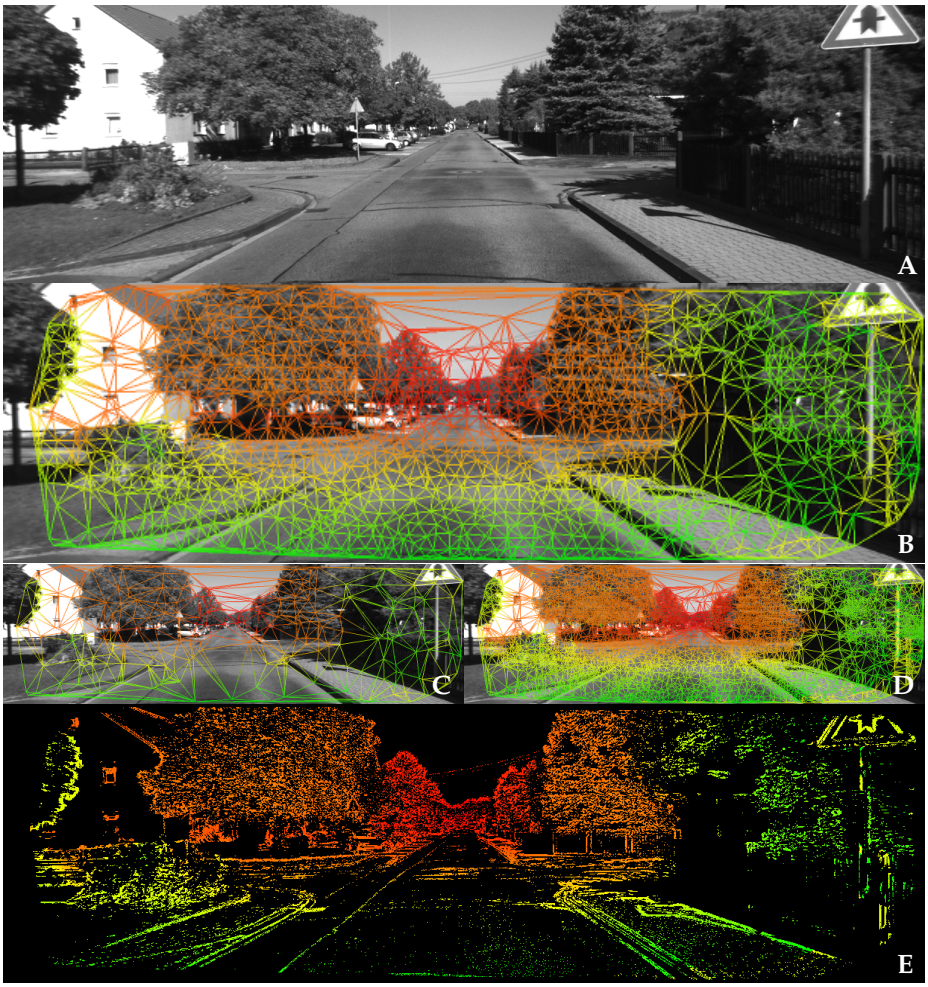


Figure 4-2: **High-Performance and Tunable Stereo Reconstruction** ▶ The proposed high-performance stereo matching method provides semi-dense reconstruction (**E**) of the scene, capable of running at a peak frame-rate of 120Hz (8.2 ms, VGA resolution). Our approach maintains a piece-wise planar representation that enables the computation of disparities (semi-densely, and densely) for varied spatial densities over several iterations (**B**-2 iterations, **C**-1 iteration, **D**-4 iterations). Colors illustrate the scene depths, with green indicating near-field and red indicating far-field regions.

4.3.1 Spatial Support via Sparse Stereo Matching

Many state-of-the-art stereo algorithms start by exhaustively computing a pixel-level cost volume $\mathcal{O}(N_P N_D)$, for all pixels (N_P) with a fixed number of disparities N_D (usually 128) considered. Instead, we employ a similar strategy to (Geiger et al. 2011a), and first construct a piece-wise planar scene depth estimate to quickly inform a coarse depth prior or mesh. First, a sparse set of support keypoints $S = \{s_1, \ldots, s_n\}$ are detected via FAST features (Rosten and Drummond 2006) (sampled from 12x10 spatial-bins), and matched along their epipolar lines as in (Schauwecker et al. 2012) (see SparseStereo in Alg. 2). We define each support point $s_n = (u_n, v_n, d_n)^T$,

Algorithm 2 Iterative Stereo Reconstruction

Input: (I_l, I_r, Ω_I): Input gray-scale stereo images and high-gradient regions
Output: D_f: Disparities at high-gradient regions (Semi-Dense)
Globals: Refer to Table 4.1 for description of variables

▷ Initialize final disparity and associated cost
$D_f \leftarrow [0]_{[H\times W]}, C_f \leftarrow [t_{hi}]_{[H\times W]}, sz_{occ} \leftarrow 32$
▷ S: Set of N support points (Section 4.3.1)
$S_1 \leftarrow$ SparseStereo(I_l, I_r)
▷ Tessellated mesh with estimated disparities
$\mathcal{G}(S_1) \leftarrow$ DelaunayTriangulation(S_1)
for $it = 1 \rightarrow n_{iters}$ **do**
  ▷ Dense piece-wise planar disparity (Section 4.3.2)
  $D_{it} \leftarrow$ DisparityInterpolation$(\mathcal{G}(S_{it}))$
  ▷ Cost evaluation given interpolated disparity (Section 4.3.3)
  $C_{it} \leftarrow$ CostEvaluation(I_l, I_r, D_{it})
  ▷ Refine disparities (Section 4.3.4)
  $C_g, C_b \leftarrow$ DisparityRefinement(D_{it}, C_{it})
  ▷ Prepare for next iteration, if not last iteration
  if $it \neq n_{iters}$ **then**
    ▷ Re-sample regions with high matching cost (Section 4.3.5)
    $S_{it+1} \leftarrow$ SupportResampling(C_g, C_b, S_{it})
    ▷ Tessellated mesh with estimated disparities
    $\mathcal{G}(S_{it+1}) \leftarrow$ DelaunayTriangulation(S_{it+1})
    ▷ Decrease occupancy grid size by factor of 2
    $sz_{occ} = \max(1, sz_{occ}/2)$
  end if
end for

similar to (Geiger et al. 2011a), as the concatenation of their image coordinates $(u_n, v_n) \in \mathbb{N}^2$, and their corresponding disparity $d_n \in \mathbb{N}$. Using these support points as vertices with known depths, a piece-wise planar mesh is constructed via Delaunay-Triangulation. (see Figure 4-3, DelaunayTriangulation in Alg. 2).

4.3.2 Mesh Triangulation and Disparity Interpolation

We refer to the planar regions in the delaunay triangulation as candidate planes, as they are constructed from the sparse set of support points whose disparities are estimated via epipolar search. These candidate planes provide a strong measure of an underlying surface, and can be used to quickly verify the hypothesized planes. Inspired by previous work on candidate-plane validation (Bleyer et al. 2011), we leverage this efficient verification step to iteratively hypothesize candidate regions in the disparity image, thereby limiting the effective disparity search space to fewer

Name	Scope	Description
I_l, I_r	L	Input gray-scale stereo images
H, W	G	Dimensions of input image I_l
Ω, Ω_I	G	Set of all pixels in image, and subset of high-gradient pixels
S	L	Sparse support pixels with valid depths
$\mathcal{G}(S)$	L	Graph resulting from Delaunay Triangulation over S
X	L	Re-sampled or detected support pixels with unknown depths
D_f	G	Final disparity image
C_f	G	Cost matrix associated to D_f
D_{it}	L	Intermediate disparity (interpolated)
C_{it}	L	Cost associated to D_{it}
C_g	L	Cost associated with regions of high confidence matches
C_b	L	Cost associated with regions of invalid disparities
N_D	G=	Maximum number of disparities considered
sz_{occ}	G	Occupancy grid size used for re-sampling
t_{lo}, t_{hi}	G	Lower and upper cost threshold for validating disparities
n_{iters}	G	Number of iterations the algorithm is allowed to run

Table 4.1: **Algorithm Nomenclature ▶** Description of symbols used in the proposed stereo matching algorithm, and their corresponding scope (G:Global or L:Local) within the implementation. Parameter values: $N_D = 128$, $sz_{occ} = 64$, $t_{lo} = 0.07$, $t_{hi} = 0.2$, $n_{iters} = \{1, \ldots, 10\}$.

than 3-5 disparity levels (not limiting to integer-valued disparities as most dense methods do).

At every intermediate step, we treat the stereo disparity image D_{it} as being constructed in a piece-wise planar manner via the Delaunay tessellated mesh. Each 3D planar surface or triangle, can be described by its 3D plane parameters $(\pi_1, \pi_2, \pi_3, \pi_4) \in \mathbb{R}^4$ given by

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0 \tag{4.2}$$

For a stereo setup with a known baseline B, and known calibration ($u = fX/Z$, $v = fY/Z$, and $d = fB/Z$), the above equation reduces to

$$\pi'_1 u + \pi'_2 v + \pi'_3 = d \tag{4.3}$$

where $\pi' = (\pi'_1, \pi'_2, \pi'_3) \in \mathbb{R}^3$ are the plane parameters in disparity space.

In order to estimate interpolated disparities on a pixel-level basis, we first construct a lookup-table that identifies the triangle and its plane coefficients for each pixel (u, v) in the left image. Subsequently, the parameters π' for each triangle are obtained by solving a linear system as done in (Geiger et al. 2011a), and are re-estimated every time after the Delaunay triangulation step. The resulting piece-wise planar tessellation can be used to linearly interpolate regions within the disparity image using the estimated plane parameters π' (see DisparityInterpolation in Alg. 2).

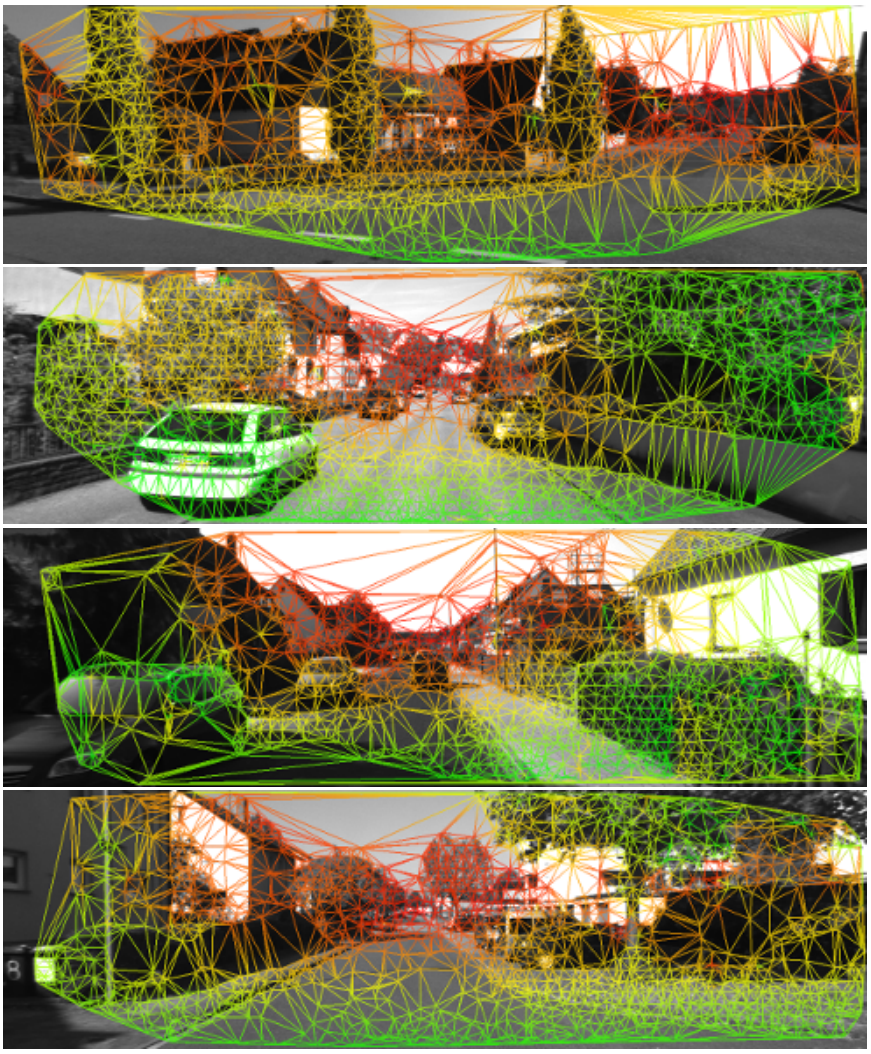


Figure 4-3: **Depth prior determined via Delaunay triangulation of sparse support points ►** Vertices in the mesh correspond to the sparse support points, or re-sampled support, while the triangular regions represent the piece-wise planar scene reconstruction.

4.3.3 Cost Evaluation

The interpolated disparity image resulting from every tessellation provides a set of candidate depths that could potentially contain valid scene points. In order to validate these interpolated disparities, we perform Census window-based matching on a 5x5 patch (Hirschmüller and Scharstein 2009; Zabih and Woodfill 1994) between the left and right stereo images. The resulting matching cost is normalized and retained to be validated in the next step (see CostEvaluation in Alg. 3).

Algorithm 3 CostEvaluation

Input: (I_l, I_r, D_{it}): Left/Right stereo image, and interpolated disparity
Output: C_{it}: Matching cost corresponding to D_{it}

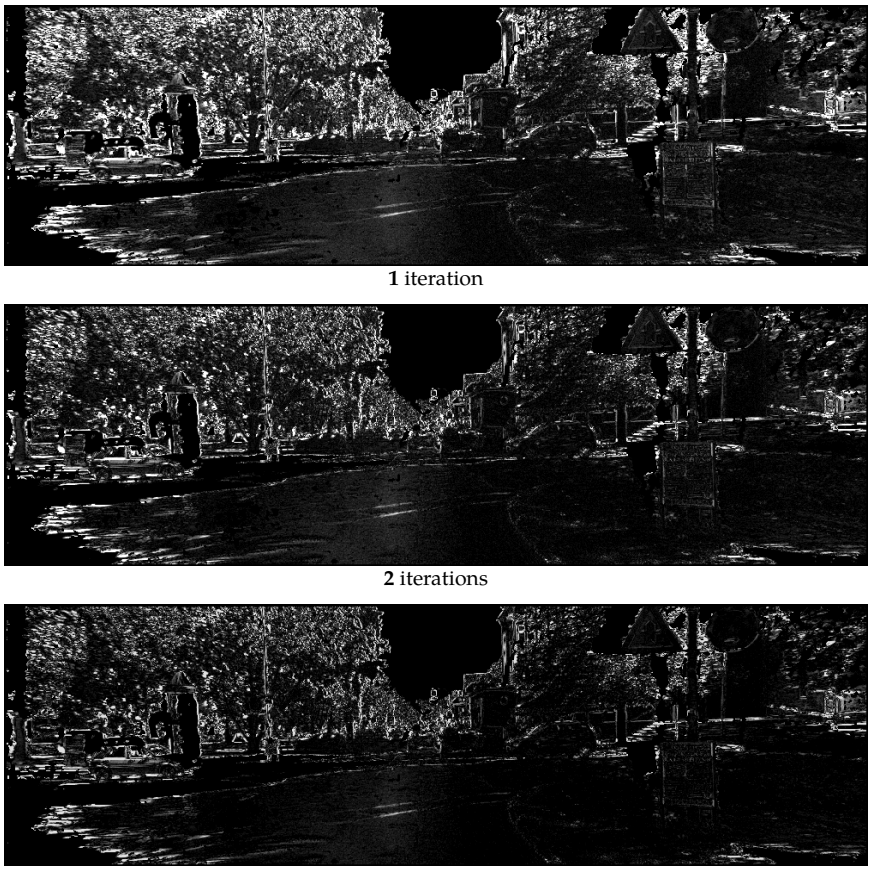
1: **for** $(u, v) \in \Omega_I$ **do**
▷ Interpolated disparity at (u, v)
2: $d \leftarrow D_{it}(u, v)$
▷ Census-based 5x5 window matching
3: $C_{it}(u, v) \leftarrow$ CensusMatchingCost$(I_l(u, v), I_r(u - d, v))$
4: **end for**

4.3.4 Disparity Refinement

The interpolated disparities computed from the tessellation may or may not necessarily hold true for all pixels. For high-gradient regions in the image, the cost computed between the left and right stereo patch for the given interpolated disparity can be a sufficiently good indication to validate the candidate pixel disparity. We use this assumption to further refine and prune candidate disparities based on the per-pixel cost computed in the previous step, as characterized by validated (C_g) and invalidated (C_b) cost regions. Thus, we can invalidate every pixel p in the left image, if the cost associated $c(p, p - d_i)$ with matching the pixel in the right image with a given interpolated disparity d_i is above a maximum permissible cost t_{hi}. The same approach is used to validate pixels that fall within a suitable cost range ($< t_{lo}$) whose correspondence certainty is high. This step also allows incorrectly matched regions to be resampled and re-evaluated for new stereo matches as the interpolated costs of regions around the falsely matched corners are driven sufficiently high. Additionally, the disparities corresponding to the least cost for each pixel is updated with every added iteration, ensuring that the overall stereo matching cost is always reduced (see Step 6 in Alg. 4). For more details regarding this step see DisparityRefinement in Alg. 4.

4.3.5 Support Resampling

The disparity refinement step establishes pixels or regions in the image whose disparities need to be re-evaluated, while also simultaneously providing reliable disparities to further utilize in the matching process. With a discretized occupancy grid of size ($sz_{occ} \times sz_{occ}$), pixels with the highest matching cost within a 32x32 (sz_{occ} is initialized to 32) window are established, and re-sampled. These re-sampled pixels are strong indicators of occluding edges, and sharp discontinuities in depth, making them viable candidates for epipolar-constrained dense stereo matching.



1 iteration

2 iterations

3 iterations

Figure 4-4: **Iterative refinement** ► With every successive iteration, more piece-wise planar regions are added to approximate the scene, thereby reducing the overall cost incurred by the approximation. The lower the intensity value of the cost image (i.e. lower matching cost), the better approximated the scene is to the ground truth.

Subsequently, the re-sampled keypoints are densely matched via epipolar search, and new support points $S_{matched}$ are established as a result. Another valuable feature is the ability to inform disparities at greater resolution and accuracy with every subsequent iteration; the discretization of the occupancy grid is reduced by a factor of 2 so that pixels are more densely sampled with every successive iteration (see SupportResampling in Alg. 5).

4.3.6 Iterative Reconstruction

The stereo matching proceeds to reduce the overall stereo matching cost associated with the interpolated piece-wise planar disparity map. High-matching cost regions are re-sampled and re-estimated to better fit the piece-wise planar disparity map to the true scene disparity. With every subsequent iteration, new keypoints are

Algorithm 4 DISPARITYREFINEMENT

Input: (D_{it}, C_{it}): Interpolated Disparity and associated matching cost
Output: C_g, C_b: Costs associated with regions of high and low matching confidence disparities

$H' \leftarrow \frac{H}{sz_{occ}}, W' \leftarrow \frac{W}{sz_{occ}}$
▷ C_g: Cost matrix of confident supports: $(u, v, d, cost)$
$C_g \leftarrow [0, 0, 0, t_{lo}]_{[H' \times W']}$
▷ C_b: Cost matrix of invalid matches: $(u, v, cost)$
$C_b \leftarrow [0, 0, t_{hi}]_{[H' \times W']}$
for $(u, v) \in \Omega_I$ **do**
    ▷ Establish occupancy grid for resampled points
    $u' \leftarrow \frac{u}{sz_{occ}}, v' \leftarrow \frac{v}{sz_{occ}}$
    ▷ If matching cost is lower than previous best final cost
    if $C_{it}(u, v) < C_f(u, v)$ **then**
        $D_f(u, v) \leftarrow D_{it}(u, v)$
        $C_f(u, v) \leftarrow C_{it}(u, v)$
    end if
    ▷ If matching cost is lower than previous best valid cost
    if $C_{it}(u, v) < t_{lo}$ **and** $C_{it}(u, v) < C_g(u', v', 4^{\dagger})$ **then**
        $C_g(u', v') \leftarrow (u, v, D_{it}(u, v), C_{it}(u, v))$
    end if
    ▷ If matching cost is higher than previous worst invalid cost
    if $C_{it}(u, v) > t_{hi}$ **and** $C_{it}(u, v) > C_b(u', v', 3^{\dagger})$ **then**
        $C_b(u', v') \leftarrow (u, v, C_{it}(u, v))$
    end if
end for

†Matrices are 1-indexed

Algorithm 5 SUPPORTRESAMPLING

Input: (C_g, C_b, S_{it}): Matching costs for confident/invalid matches
Output: S_{it+1}: New support points for tessellation

$S_{it+1} \leftarrow S_{it}, X \leftarrow \emptyset$
for $(u, v) \in \Omega_I$ **do**
    ▷ Perform sparse epipolar stereo for resampled invalid pixels
    if $C_b(u, v) \neq 0$ **then**
        $X \leftarrow \{X, (u, v)\}$
    end if
    ▷ Resample confident pixels and add to support
    if $C_g(u, v) \neq 0$ **then**
        $S_{it+1} \leftarrow \{S_{it+1}, (u, v)\}$
    end if
end for
▷ Re-estimate disparities via epipolar search
$S_{matched} \leftarrow$ SPARSEEPIPOLARSTEREO(I_l, I_r, X)
$S_{it+1} \leftarrow \{S_{it+1}, S_{matched}\}$

sampled, tessellated to inform a piece-wise planar depth prior, and further evaluated to reduce the overall matching cost. With such an iterative procedure, the

overall stereo matching cost is reduced, with the obvious cost of added computation or run-time requirement (see Figure 4-5).

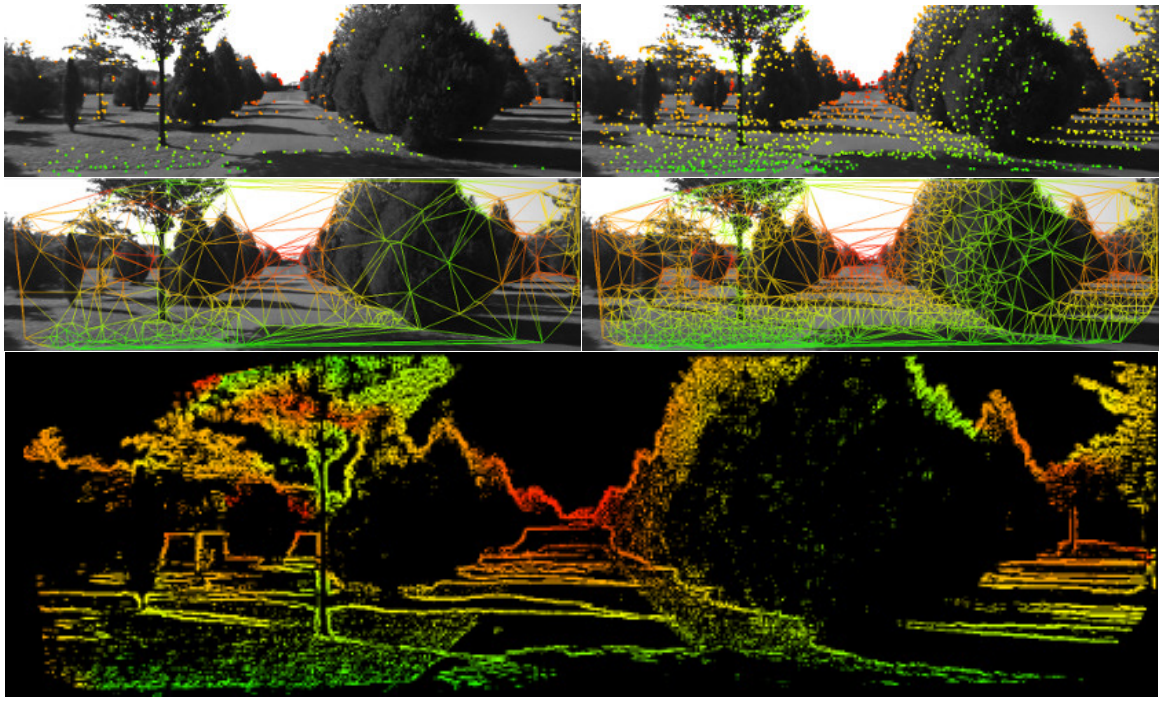


Figure 4-5: **Depth prior estimated with every subsequent iteration ►** As expected, the density of support points increase, with the piece-wise planar representation better fitting to the true scene disparity map. (Rows 1-2, Column 1: After 1 iteration, Column 2: After 2 iterations). Row 3 illustrates the final semi-dense reconstruction after 2 iterations.

4.4 Experiments and Results

In this section, we evaluate the proposed high-performance stereo matching method. We evaluate the matching accuracy and runtime performance of our proposed method on the popular KITTI dataset (Geiger et al. 2012) and on two different stereo cameras, namely the Point Grey Bumblebee2 1394a[1], and the ZED Stereo Camera[2]. The KITTI dataset contains rectified gray-scale stereo imagery at a resolution of 1241x376 (0.46 MP), captured from 2 Point Grey Flea2 cameras mounted with a baseline of 0.54m. We compare against stereo matching algorithms that are commonly used in robotics applications - the popular implementation of Semi-Global Matching (Hirschmüller 2005) in OpenCV (Semi-Global Block-Matching or SGBM), ELAS (Geiger et al. 2011a) and Line-Sweep (Ramalingam et al. 2015). We provide

[1] http://www.ptgrey.com/stereo-vision-cameras-systems
[2] https://www.stereolabs.com/zed/

a thorough analysis of the trade-offs between matching accuracy and run-times achievable by our proposed method, across varied hardware and environmental setups.

4.4.1 Evaluation on KITTI Dataset

Method	Accuracy (%)			
	< 2px	< 3px	< 4px	< 5px
SGBM (Hirschmüller 2005)	89.0	93.9	95.6	96.5
ELAS (Geiger et al. 2011b)	92.7	96.1	97.3	97.9
Line-Sweep (Ramalingam et al. 2015)	72.6	81.2	84.7	86.7
Ours-1†	83.1	89.9	92.9	94.7
Ours-2†	83.5	90.2	93.2	94.9
Ours-4†	85.4	91.4	94.0	95.5

†The number next to the method indicates the number of iterations the algorithm is allowed to run.

Table 4.2: **Disparity estimation on KITTI dataset** ▶ Analysis of accuracy of our system on the KITTI dataset (Geiger et al. 2012), as compared to popular stereo implementations including OpenCV's Semi-Global Block-Matching (Hirschmüller 2005), ELAS (Geiger et al. 2011b) and Line-Sweep (Ramalingam et al. 2015). The number next to the method indicates the number of iterations the algorithm is allowed to run. The accuracy results are evaluated **only** over high-gradient (semi-dense) regions in the image.

Disparity Estimation Accuracy In order to evaluate our proposed semi-dense method against existing methods, we only consider disparities in the image that have large image gradients or edges. Currently, semi-dense methods cannot be fully evaluated on the KITTI dataset, since the test server interpolates missing disparities, introducing several errors in the disparity estimates and overall accuracy. For all valid and non-occluding semi-dense edges, we report the absolute difference between the proposed method and existing state-of-the-art stereo implementations. In our experiments on the provided KITTI stereo evaluation kit, we find that greater than 89.9% of edge pixels had a disparity value of less than 3 pixels with respect to ground truth for the single pass variant (*Ours-1*). As seen in table 4.2, with increased number of iterations, the same algorithm improves overall performance (*Ours-2*: 90.2%, *Ours-4*: 91.4%). For the stereo setup on the KITTI dataset, 3 pixels correspond to $\pm$3cm at a depth of 2 meters and $\pm$80cm at a depth of 10m. In addition, we compare against recent work (Ramalingam et al. 2015) on semi-dense reconstruction on the KITTI dataset, and achieve significantly better disparity accuracy using our approach compared to 81.2% of (Ramalingam et al. 2015). In Table 4.2 below, we compare the disparities computed by our proposed method, and compare against existing stereo matching implementations, including

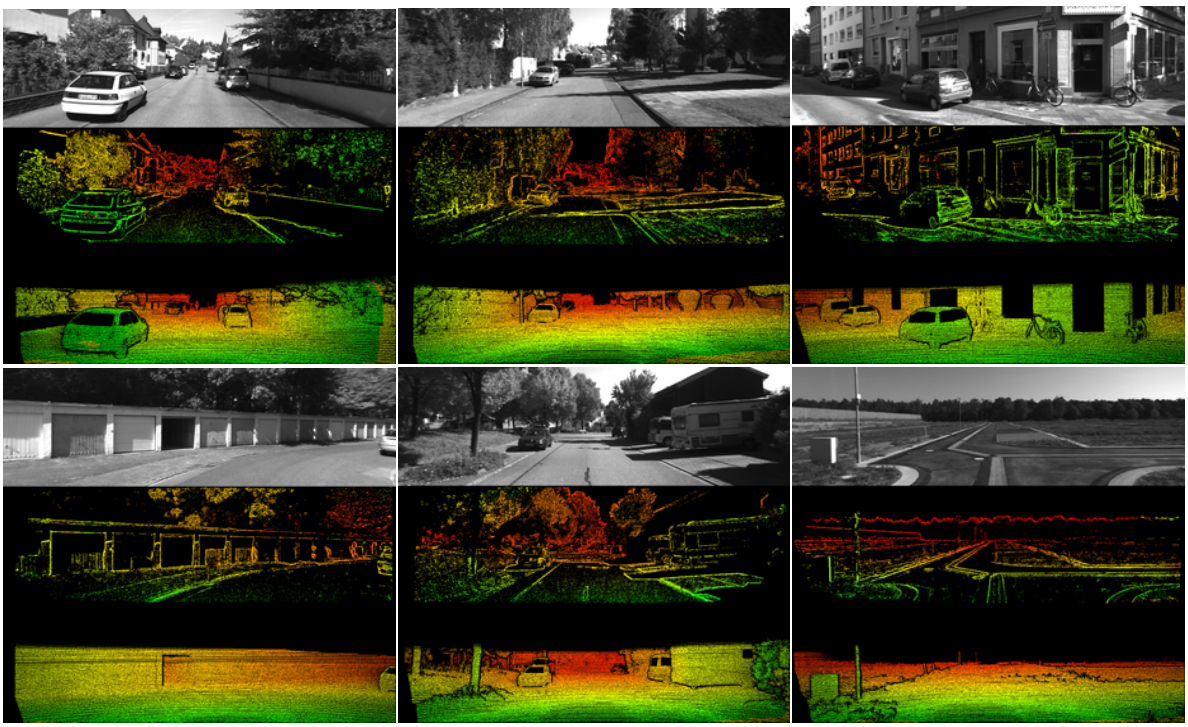


Figure 4-6: **Proposed Stereo Disparity Estimation ►** Illustrations of our proposed stereo disparity estimation method (*Ours-2*, **row 2**) on the KITTI dataset with corresponding ground truth estimates (**row 3**) obtained from projecting LiDAR data on to the left camera. Despite its short execution time, our approach shows accurate estimates of disparities for a variety of scenes. The ground truth estimates are provided as reference, and are valid points that fall below the horizon. Similar colors indicate similar depths at which points are registered.

Semi-Global Matching, ELAS, and Line-Sweep. We do note that the main reason for reduced accuracy compared to state-of-the-art methods is due to the local nature of the algorithm, as compared to the global regularization methods used in SGBM and ELAS. We visualize the results of our proposed method in Figure 4-6 with the corresponding ground truth disparities.

Stereo Reconstruction In this section, we show the qualitative performance of our stereo disparity estimation approach via stereo reconstructions fused over multiple frames from a moving camera. We use the stereo imagery from the KITTI dataset, and the corresponding ground truth poses to reconstruct scenes over a short window time frame to qualitatively illustrate the stereo matching consistency our approach provides. In Figure 4-7, we show our reconstruction results from various sequences. The reconstructions of building facades, cars, road terrain, and road curbs are well-detailed with little noise. Furthermore, unstructured and thin occluding edges such as trees, and their trunks are also well reconstructed. See

video via the following link[3].

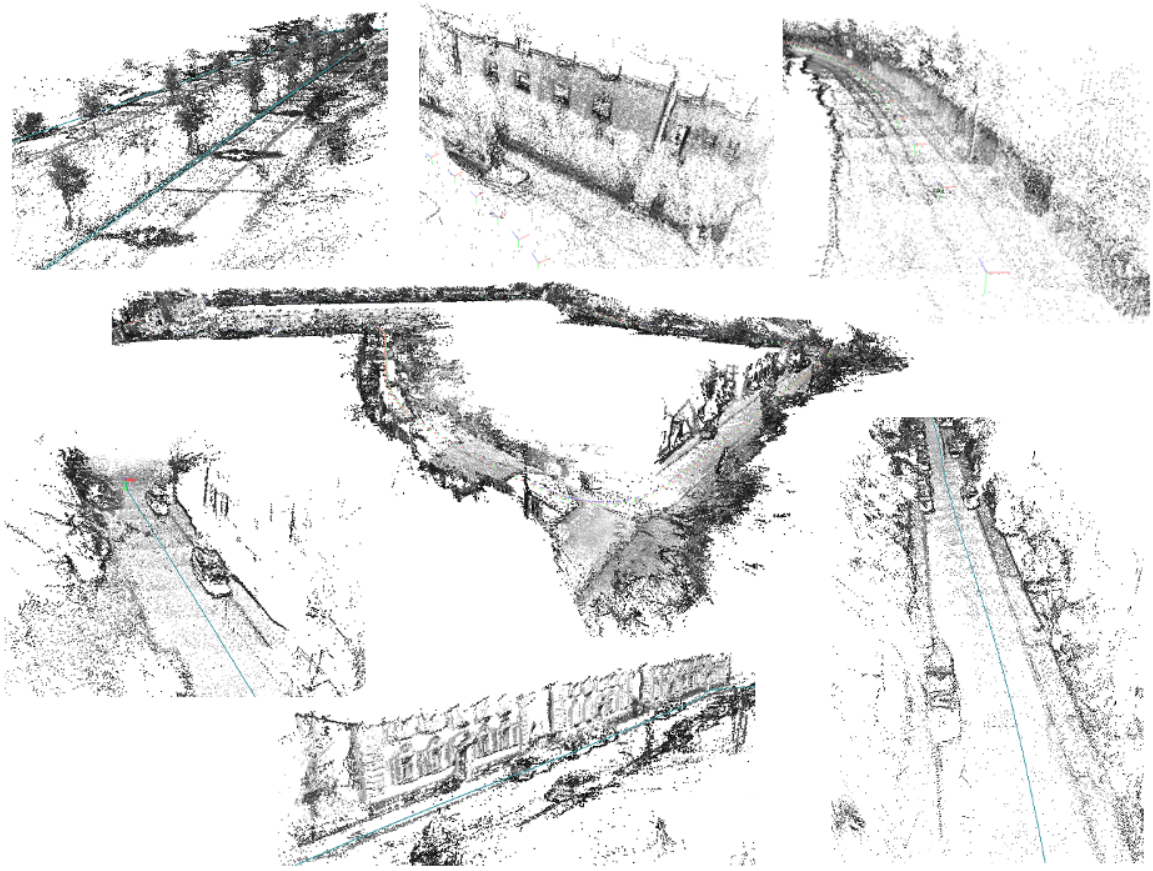


Figure 4-7: **Scene Reconstructions ►** Illustrated above are various scenes reconstructed using our proposed stereo matching approach. We use the ground truth poses from the KITTI dataset to merge the reconstructions from multiple frames, qualitatively showing the consistency in stereo disparity estimation of our approach.

Runtime Performance Most existing stereo matching algorithms have focused their efforts on the accuracy, without much regard for the runtime performance of these systems. In this work, we focus on the potential benefits and trade-offs of stereo matching accuracy and runtime performance. Due to the iterative nature of our proposed method, we show that our approach can be tuned to various accuracy and runtime operational levels, particularly beneficial for robotics applications. In our experiments (Table 4.3 and 4.4), we evaluate the runtime performance of our proposed method across several standard image resolutions ranging from WVGA (320x240) to HD1080 (1920x1080). For the common stereo image resolutions (800x600), our approach provides a speed-up factor of 32x for the single-pass stereo matching case, and a factor of 12x for the two-pass stereo matching case, as

[3] http://people.csail.mit.edu/spillai/projects/fast-stereo-reconstruction/pillai_fast_stereo16.mp4

compared to OpenCV's SGBM (Hirschmüller 2005) implementation.

Method	Accuracy (%)	Run-time (Hz/ms)	Speed-up
SGBM (Hirschmüller 2005)	93.9	2.8 Hz / 351.9 ms	1x
ELAS (Geiger et al. 2011b)	**96.1**	6.2 Hz / 160.9 ms	2.1x
Line-Sweep (Ramalingam et al. 2015)	81.2	14.2 Hz / 70.0 ms	5x
Ours-1†	89.9	**92.2 Hz / 10.8 ms**	**32.4x**
Ours-2†	90.2	**34.6 Hz / 28.9 ms**	**12.2x**
Ours-4†	91.4	**17.2 Hz / 58.2 ms**	**6.0x**

Table 4.3: **Run-time performance ►** Analysis of run-time performance of our system on the KITTI (1241 x 376 px, 0.46 MP) dataset (Geiger et al. 2012), as compared to popular stereo implementations including OpenCV's Semi-Global Block-Matching (Hirschmüller 2005) and ELAS (Geiger et al. 2011b). The number next to the method indicates the number of iterations the algorithm is allowed to run. We achieve comparable performance, with a run-time speed-up of approximately **32 x**. Accuracy is reported for disparities that are within 3 pixels of ground truth.

Method		Image Resolution (px)				
		320x240	640x480	800x600	1280x720	1920x1080
SGBM (Hirschmüller 2005)	Runtime (ms)	53.4	216.7	360.0	763.7	1873.7
ELAS (Geiger et al. 2011b)		22.7	107.2	170.3	332.7	650.9
Ours-1†		**3.0**	**8.2**	**10.9**	**18.2**	**35.9**
Ours-2†		**6.4**	**19.2**	**27.4**	**46.0**	**81.0**
Ours-4†		**18.7**	**64.9**	**99.2**	**172.9**	**287.2**

Table 4.4: **Running Time vs. Image Resolution ►** We compare the runtime performance of our proposed approach (*Ours*) with existing state-of-the-art solutions for varied image resolutions. As shown in the table, our proposed stereo algorithm performs an order of magnitude faster that other popular approaches for high-resolution (720P) stereo imagery. The number next to the method indicates the number of iterations the algorithm is allowed to run.

Failure Modes While the iterative reconstruction algorithm admits adjustable precision in disparity estimation, it however is still vulnerable to the typical concerns in image-based reconstruction systems. With varied lighting conditions (i.e. over-exposed, under-exposed, blurry imaging conditions), the algorithm struggles to determine the set of salient and sparse features for initializing the iterative refinement procedure. Figure 4-8 illustrates a few failure modes in our proposed approach. In future work, we hope to mitigate some of these challenging scenarios by incorporating robust lighting-invariant features and adaptive reconstruction policies.

4.4.2 Evaluation on Commodity Hardware

With the advent of the USB3 standard, high-framerate stereo cameras have now started to become mainstream. These devices open the door to newer data through-

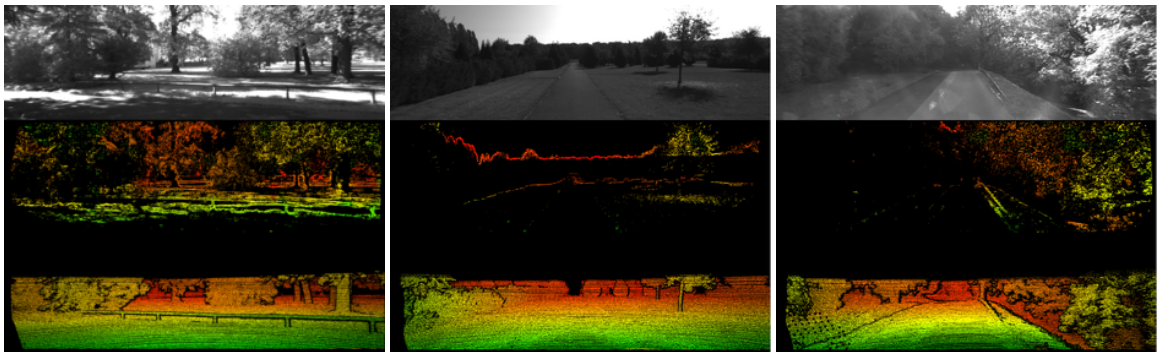


Figure 4-8: **Disparity estimation failure on the KITTI dataset ► Top row:** Left image from stereo pair, **Middle row:** Disparity estimated with our proposed algorithm, **Bottom row:** LiDAR ground truth disparity. Under varied imaging conditions (over-exposed, under-exposed, or blurry images) our algorithm struggles with similar concerns that typical stereo-based algorithms face. In most cases, the difficulty is in the recovering the initial set of features whose disparities are used to seed the iterative algorithm. In some scenarios, only a very small fraction of the image is reconstructed.

put capacities, however, existing state-of-the-art stereo algorithms fail to meet such high throughput requirements. To this end, in addition to the KITTI dataset, we benchmark our proposed method on two different stereo platforms including the BumbleBee2, and the newly introduced USB3-driven ZED Stereo Camera. The Bumblebee2 (12cm baseline) operates at 48 FPS providing gray-scale stereo imagery at a resolution of 648x488, while the ZED Camera is configured to operate at 60Hz with a resolution of 1280x720. In our experiments, we compare the disparities estimated from our approach against that of SGBM and report results on its accuracy and runtime performance (see Table 4.5).

Method	**Accuracy (%)**	
	BumbleBee2	**ZED**
ELAS (Geiger et al. 2011b)	81.1	**91.6**
Line-Sweep (Ramalingam et al. 2015)	83.9	77.2
Ours-1†	89.6	87.5
Ours-2†	**90.8**	87.3

Table 4.5: **Disparity estimation with commodity hardware ►** Analysis of accuracy of our system on the BumbleBee2 and ZED Stereo Camera, with Semi-Global Block-Matching (Hirschmüller 2005) considered as ground truth. We compare against other stereo implementations including ELAS and Line-Sweep and report the accuracy for disparities that are within 3 pixels of ground truth. The number next to the method indicates the number of iterations the algorithm is allowed to run.

4.4.3 Implementation Details

We use the high-speed sparse-stereo implementation of (Schauwecker et al. 2012), and the Delaunay Tessellation is performed via the Triangle[4] library for the initial set of support tessellation. Besides the 5x5 Census-based block matching that is implemented using specialized SSE instructions (Schauwecker et al. 2012), the rest of the code is implemented on a single-CPU thread in C++, without any specialized instruction sets or GPU-specific code. All the results of our code are tested on an Intel(R) Core(TM) i7-3920XM CPU @ 2.90GHz. We do note that while our current implementation refines disparities every iteration in batch, this step can be highly-parallel and asynchronous due to the recursive nature of the refinement over the tessellated structure.

4.5 Discussion and Future Work

Resource-aware Computation Several robotics applications adhere to strict computational budgets and runtime requirements, depending on their task domain. Some systems require the ability to actively adapt to varying runtime requirements and conditions, and adjust parameters accordingly. In the context of mapping and navigation, robots may need to map the world around them, in a slow but accurate manner, while also requiring the ability to avoid dynamic obstacles quickly and robustly. Such systems would need to dynamically change the accuracy requirements in order to achieve their desired runtime performance, given a fixed compute budget. We hope this work encourages such capabilities, and intend to consider a tighter integration with mobile platforms that can leverage this capability.

Plan-aware Reconstruction Another potential direction for improvement would be in the generation of rapid and high-fidelity reconstructions, given a sufficiently coarse trajectory plan or foveation. Trajectory plans can be advantageous in reducing overall computation, especially when only a small fraction of the scene needs to be queried and reconstructed. Figure 4-9 provides a glimpse into this capability where the volume swept by the predicted trajectory can be used to restrict regions that are reconstructed by our algorithm. Given a reasonable exploration-exploitation strategy, our approach can provide promising flexibility in exploiting accurate and rich scene information, while also being able to adjust itself to rapidly handle dynamic scenes during the exploration stage.

[4] https://www.cs.cmu.edu/~quake/triangle.html

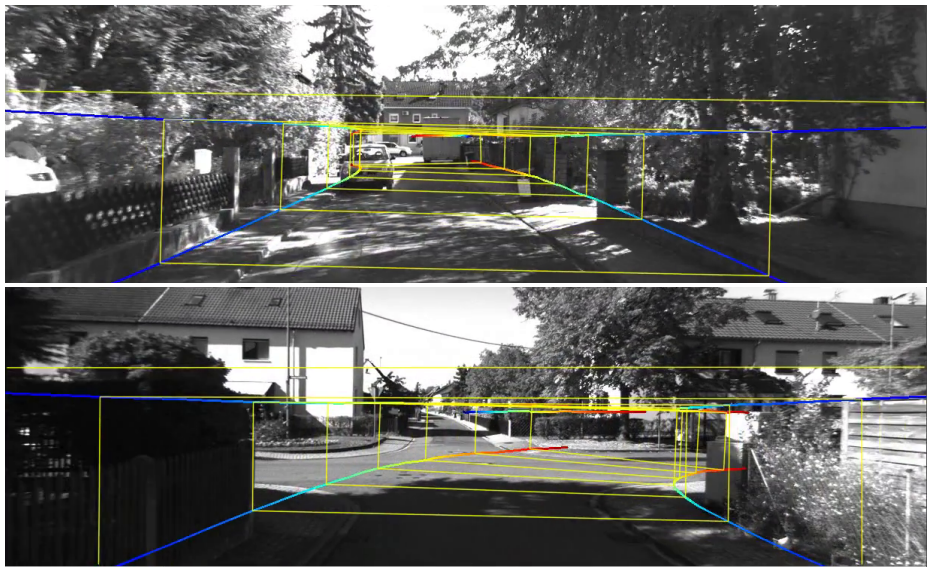


Figure 4-9: **Plan-aware Reconstruction ►** Trajectory plans can provide useful information such as volumes within which the stereo disparity reconstruction can be restricted to, thereby increasing overall efficiency of the algorithm.

4.6 Chapter Summary

As robots leverage their SLAM capabilities in other tasks, it is critical to re-evaluate the underlying map representation that is maintained for the purpose of vision-based SLAM. We seek a *flexible map representation* that can be directly estimated during mapping, while being readily usable in the context of motion planning in mobile robots with little modification. With agile mobile robots in context, we consider the mapping problem using stereo-vision, and propose a potential solution towards enabling this map-representation goal. In this work, we propose an iterative and high-performance mesh reconstruction algorithm estimated from stereo imagery. By maintaining a piece-wise planar assumption, we develop a stereo matching strategy that recursively tessellates the scene into piece-wise planar regions so that it appropriately reconstructs it, given a fixed run-time requirement as provided by the user. By evaluating the matching costs for candidate planes, our approach quickly identifies planar regions, and repeats the process for non-planar regions by introducing more stereo matches within these regions and re-tessellating them. We compare against stereo matching algorithms that are commonly used in robotics applications and provide promising results of the trade-offs between matching accuracy and run-times achievable by our proposed method, across varied stereo dataset and hardware setups. We envision that in the future, these tunable mesh representations can potentially enable robots to quickly recon-

struct their immediate surroundings while being able to directly generate plans from them and maneuver at high-speeds.

Chapter 5

Self-Supervised Visual Ego-motion Learning in Robots

Fundamental to this thesis is the ability for mobile robots to perform vision-based SLAM. While visual-SLAM algorithms have enabled significant advances in various industries today, they are still limited in their ability to learn from new experiences and adapt to newer environments. We envision robots to be able to learn new model representations with experience by bootstrapping known model-based solutions as a supervisory signal. In this thesis, we advocate for the ability to leverage SLAM-as-a-sensor to provide this bootstrapping mechanism. In the following chapters, we show how a GPS-aided SLAM solution could potentially bootstrap a *self-supervised* visual SLAM front-end.

SLAM has been studied in a broad range of applications, and is typically considered the sensor-agnostic back-end optimization problem to an application-specific front-end. In this thesis, we focus on vision-based SLAM front-ends that transform raw image-based sensor measurements into meaningful constraints that the SLAM-backend can eventually solve. Most state-of-the-art implementations of vision-based SLAM front-ends heavily rely on heuristics and design choices at various stages of the pipeline (including feature detection, description, tracking/matching, RANSAC). Furthermore, these methods are designed for a specific lens-characteristic and require further calibration and tuning before they can be deployed in a standard visual-SLAM architecture.

In this chapter, we focus on learning a visual odometry front-end, and show how a GPS-aided SLAM solution can be used to develop a fully trainable solution to visual ego-motion estimation for varied camera optics. We propose a neural net-

work architecture that maps observed optical flow vectors to an ego-motion density estimate via a Mixture Density Network (MDN). By modeling the architecture as a Conditional Variational Autoencoder (C-VAE), our model is able to provide introspective reasoning and prediction for ego-motion induced scene-flow. Additionally, our proposed model is especially amenable to *bootstrapped ego-motion learning* in robots where the supervision in ego-motion estimation for a particular camera sensor can be obtained from a GPS-aided SLAM solution (i.e. GPS/INS and wheel-odometry fusion). Through experiments, we show the utility of our proposed approach in enabling the concept of self-supervised learning for visual ego-motion estimation in autonomous robots.

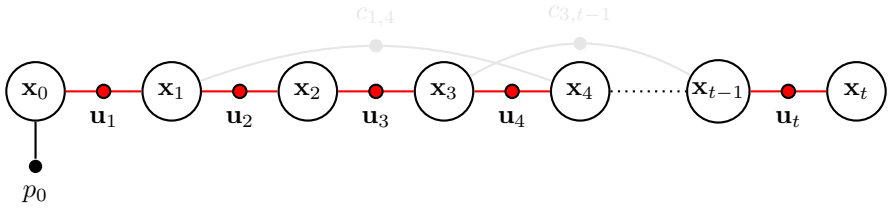


Figure 5-1: **Visual Ego-motion Learning ►** In a typical factor-graph formulation of Pose-Graph SLAM, the visual ego-motion contributes to the factors ($u_{i-1,i}$) in the odometry chain (*in red*). This chapter focuses on recovering these odometry factors by visually tracking subsequent images.

5.1 Introduction

Visual odometry (VO) (Nistér et al. 2004), commonly referred to as ego-motion estimation, is a fundamental capability that enables robots to reliably navigate its immediate environment. With the wide-spread adoption of cameras in various robotics applications, there has been an evolution in visual odometry algorithms with a wide set of variants including monocular VO (Konolige et al. 2010a; Nistér et al. 2004), stereo VO (Howard 2008; Kitt et al. 2010) and even non-overlapping *n*-camera VO (Hee Lee et al. 2013; Kneip et al. 2013). Furthermore, each of these algorithms has been custom tailored for specific camera optics (pinhole, fisheye, catadioptric) and the range of motions observed by these cameras mounted on various platforms (Scaramuzza 2011).

With increasing levels of model specification for each domain, we expect these algorithms to perform differently from others while maintaining lesser generality across various optics and camera configurations. Moreover, the strong dependence of these algorithms on their model specification limits the ability to actively monitor and optimize their intrinsic and extrinsic model parameters in an online fash-

ion. In addition to these concerns, autonomous systems today use several sensors with varied intrinsic and extrinsic properties that make system characterization tedious. Furthermore, these algorithms and their parameters are fine-tuned on specific datasets while enforcing little guarantees on their generalization performance on new data.

To this end, we propose a fully trainable architecture for visual odometry estimation in generic cameras with varied camera optics (*pinhole*, *fisheye* and *catadioptric* lenses). In this work, we take a geometric approach by posing the regression task of ego-motion as a density estimation problem. By tracking salient features in the image induced by the ego-motion (via Kanade-Lucas-Tomasi/KLT feature tracking), we learn the mapping from these tracked flow features to a probability mass over the range of likely ego-motion. We make the following contributions:

- **A fully trainable ego-motion estimator**: We introduce a fully-differentiable density estimation model for visual ego-motion estimation that robustly captures the inherent ambiguity and uncertainty in relative camera pose estimation (See Figure 5-3).
- **Ego-motion for generic camera optics**: Without imposing any constraints on the type of camera optics, we propose an approach that is able to recover ego-motions for a variety of camera models including *pinhole*, *fisheye* and *catadioptric* lenses.
- **Bootstrapped ego-motion training and refinement**: We propose a bootstrapping mechanism for autonomous systems whereby a robot self-supervises the ego-motion regression task. By fusing information from other sensor sources including GPS and INS (Inertial Navigation Systems), these indirectly inferred trajectory estimates serve as ground truth target poses/outputs for the aforementioned regression task. Any newly introduced camera sensor can now leverage this information to learn to provide visual ego-motion estimates without relying on an externally provided ground truth source.
- **Introspective reasoning via scene-flow predictions**: We develop a generative model for optical flow prediction that can be utilized to perform outlier-rejection and scene flow reasoning.

Through experiments, we provide a thorough analysis of ego-motion recovery from a variety of camera models including pinhole, fisheye and catadioptric cameras. We expect our general-purpose approach to be robust, and easily tunable for accuracy

during operation. We illustrate the robustness and generality of our approach and provide our findings in Section 5.5.

5.2 Related Work

Recovering relative camera poses from a set of images is a well studied problem under the context of Structure-from-Motion (SfM) (Hartley and Zisserman 2003; Triggs et al. 1999). SfM is usually treated as a non-linear optimization problem, where the camera poses (extrinsics), camera model parameters (intrinsics), and the 3D scene structure are jointly optimized via non-linear least-squares (Triggs et al. 1999).

Unconstrained VO: Visual odometry, unlike incremental Structure-from-Motion, only focuses on determining the 3D camera pose from sequential images or video imagery observed by a monocular camera. Most of the early work in VO was done primarily to determine vehicle egomotion (Matthies 1989; Moravec 1980; Olson et al. 2000) in 6-DOF, especially for the Mars planetary rover. Over the years several variants of the VO algorithm were proposed, leading up to the work of Nistér et al. (2004), where the authors proposed the first real-time and scalable VO algorithm. In their work, they developed a 5-point minimal solver coupled with a RANSAC-based outlier rejection scheme (Fischler and Bolles 1981) that is still extensively used today. Other researchers (Corke et al. 2004) have extended this work to various camera types including catadioptric and fisheye lenses.

Constrained VO: While the classical VO objective does not impose any constraints regarding the underlying motion manifold or camera model, it however contains several failure modes that make it especially difficult to ensure robust operation under arbitrary scene and lighting conditions. As a result, imposing egomotion constraints has been shown to considerably improve accuracy, robustness, and run-time performance. One particularly popular strategy for VO estimation in vehicles is to enforce planar homographies during matching features on the ground plane (Ke and Kanade 2003; Liang and Pears 2002), thereby being able to robustly recover both relative orientation and absolute scale. For example, Scaramuzza et al. (Scaramuzza 2011; Scaramuzza et al. 2009b) introduced a novel 1-point solver by imposing the vehicle's non-holonomic motion constraints, thereby speeding up the VO estimation up to 400Hz.

Data-driven VO: While several model-based methods have been developed specif-

ically for the VO problem, a few have attempted to solve it with a data-driven approach. Typical approaches have leveraged dimensionality reduction techniques by learning a reduced-dimensional subspace of the optical flow vectors induced by the egomotion (Roberts et al. 2009). In Ciarfuglia et al. (2014), Ciarfuglia et al. employ Support Vector Regression (SVR) to recover vehicle egomotion (3-DOF). The authors further build upon their previous result by swapping out the SVR module with an end-to-end trainable convolutional neural network (Costante et al. 2016) while showing improvements in the overall performance on the KITTI odometry benchmark (Geiger et al. 2012). Recently, Clark et al. (2016) introduced a visual-inertial odometry solution that takes advantage of a neural-network architecture to learn a mapping from raw inertial measurements and sequential imagery to 6-DOF pose estimates. By posing visual-inertial odometry (VIO) as a sequence-to-sequence learning problem, they developed a neural network architecture that combined convolutional neural networks with Long Short-Term Units (LSTMs) to fuse the independent sensor measurements into a reliable 6-DOF pose estimate for ego-motion. Our work closely relates to these data-driven approaches that have recently been developed. We provide a qualitative comparison of how our approach is positioned within the visual ego-motion estimation landscape in Table 5.1.

Method Type	Varied Optics	Model Free	Robust	Self Supervised
Traditional VO (Scaramuzza and Fraundorfer 2011)	✗	✗	✓	✗
End-to-end VO (Clark et al. 2016; Costante et al. 2016)	✗	✓	✓	✗
This work	✓	✓	✓	✓

Table 5.1: **Visual odometry landscape ►** A qualitative comparison of how our approach is positioned amongst existing solutions to ego-motion estimation.

5.3 Background

5.3.1 Visual Odometry

While VO has been widely adopted in the realm of autonomous systems today, it still poses some challenges. One of the major difficulties in recovering robust odometry estimates over long operating periods is the inherent ambiguity in solutions that 2D-to-2D matching exhibits. Fundamentally, the geometric relation between two images $\mathcal{I}_{i-1}$, and $\mathcal{I}_i$ from a calibrated camera is given by its *Essential Matrix* E. The Essential Matrix contains motion parameters that describes the relative transformation between two views up to an unknown scale factor: $E \simeq \mathbf{t}_{\times}\mathbf{R}$,

where $\mathbf{t}_\times$ is the skew-symmetric matrix of the vector $\mathbf{t} = [t_x, t_y, t_z]$. The main geometric relationship of the essential matrix and the set of 2D-to-2D measurements is given by the epipolar constraint: $x'^T E x = 0$ where x and x' are the corresponding features in image $\mathcal{I}_{i-1}$ and $\mathcal{I}_i$. As described earlier, the essential matrix is solved minimally using 5 feature correspondences, via the seminal work of (Nistér et al. 2004). The determined E is then decomposed into the scale-ambiguous rotation ($\mathbf{R}$) and translation ($\mathbf{t}$) component, with 4 possible solutions given by:

$$\mathbf{R} = U(\pm W^T)V^T \tag{5.1}$$

$$\mathbf{t} = U(\pm W)SU^T \tag{5.2}$$

where U, and S are recovered via the Singular Value Decomposition (SVD) of E, $E = USV^T$, and

$$W = \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

Due to this multi-modal solution space as is typical in *inverse problems*, we consider density estimation as a possible alternative to the classical L_2 VO regression formulation that existing data-driven methods have taken (Ciarfuglia et al. 2014; Clark et al. 2016; Costante et al. 2016). In fact, evaluating with an L_2 loss will likely result in solutions that straddle the modes without having any guarantees on finding any or all the relevant modes (Bishop 1994; Murphy 2012).

Furthermore, VO can also be done with a variety of cameras including perspective and omnidirectional ones. These variants require lens-specific calibration routines, and intrinsic camera model correction before they are normalized into usable bearing vectors. Most lens types require their own correction models (for both projection and distortion coefficients), and lack a unified model that is able to capture their implicit intrinsics. With these two concerns in mind, we seek a unified model that is able to implicitly learn both the non-linear intrinsics and while being able to handle the ambiguities inherent in classical 2D-to-2D visual odometry. Mixture Density Networks (MDN) (Bishop 1994) do exactly this; while capturing non-linearities in mapping the input space to a low-dimensional latent space, it also assumes the output space to be multi-modal and parameterized via a Gaussian Mixture Model (GMM).

5.3.2 Density Estimation with Mixture Density Networks

The promise of conditional probabilistic modeling arises typically in the context of *inverse problems*, where the output may sometimes be multi-modal or multi-valued for the same input value. A standard least-squares (LS) regression formulation in a potentially multi-modal belief space typically leads to a model converging to nonsensical solutions that may lie between two or more obvious modes. Instead, we seek a general purpose mechanism to model the conditional probability distribution (CPD) of the underlying manifold, that potentially exhibits multiple modes.

Mixture Density Networks (MDNs) (Bishop 1994) are a class of fully-differentiable density estimation techniques that leverage conventional neural networks to regress the parameters of a generative model such as a finite Gaussian Mixture Model (GMM). The appealing nature of MDNs is in its representational capacity of conditional probability distributions. Analogous to how conventional neural networks are capable of approximating arbitrary functions, MDNs theoretically provide a similar capability in approximating arbitrary conditional probability distributions. Additionally, one may also observe that the least-squares regression formulation turns out to be a special case of the MDN with a single mixture component $K = 1$ in the finite mixture model layer.

Here, we consider the finite Gaussian Mixture Model (GMM) whose parameters are the outputs of conventional neural network. The conditional probability density is thus represented as a convex combination of K Gaussian components, given by

$$p(\mathbf{y}|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) \mathcal{N}(\mathbf{y}|\mu_k(\mathbf{x}), \sigma_k^2(\mathbf{x})) \tag{5.4}$$

where $\pi_k(\mathbf{x})$ is the mixing coefficient for the k-th component as specified in a typical GMM. The Gaussian kernels are parameterized by their mean vector $\mu_k(\mathbf{x})$ and diagonal covariance $\sigma_k(\mathbf{x})$. It is important to note that the parameters $\pi_k(\mathbf{x}), \mu_k(\mathbf{x})$, and $\sigma_k(\mathbf{x})$ are general and continuous functions of $\mathbf{x}$. This allows us to model these parameters as the output (a^π, a^μ, a^σ) of a conventional neural network which takes $\mathbf{x}$ as its input, and is parameterized by its weights $\mathbf{w}$. Since the output of the network depends on its learned weights, we subsequently explicitly include $\mathbf{w}$ in our notation. Following (Bishop 1994), the outputs of the neural network are constrained as follows: (i) The mixing coefficients must sum to 1, i.e. $\sum_K \pi_k(\mathbf{x}; \mathbf{w}) = 1$ where

$0 \leq \pi_k(\mathbf{x}; \mathbf{w}) \leq 1$. This is accomplished via the *softmax* activation, given by

$$\pi_k(\mathbf{x}; \mathbf{w}) = \frac{\exp(a_k^\pi)}{\sum_{l=1}^{K} \exp(a_l^\pi)} \tag{5.5}$$

(ii) Variances $\sigma_k(\mathbf{x})$ are strictly positive via the *exponential* activation (Eqn 5.6).

$$\sigma_k(\mathbf{x}; \mathbf{w}) = \exp(a_k^\sigma) \tag{5.6}$$

The mean components of the output $\mu_{kj}(\mathbf{x})$ are represented by their network activations a_{kj}^μ given by

$$\mu_{kj}(\mathbf{x}; \mathbf{w}) = a_{kj}^\mu \tag{5.7}$$

The parameters of the model $\mathbf{w}$ can subsequently be learned by maximizing the data log-likelihood, or alternatively minimizing the negative log-likelihood (denoted as $\mathcal{L}_{\mathcal{MDN}}(\mathbf{w})$ in Eqn 5.8).

$$\mathcal{L}_{\mathcal{MDN}}(\mathbf{w}) = -\sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k(\mathbf{x}_n; \mathbf{w}) \mathcal{N}(\mathbf{y} | \mu_k(\mathbf{x}_n; \mathbf{w}), \sigma_k^2(\mathbf{x}_n; \mathbf{w})) \right\} \tag{5.8}$$

Once the model parameters are estimated, we are able to draw samples from the conditional distribution $p(\mathbf{y}|\mathbf{x}; \mathbf{w})$, which in some cases may be multi-valued representing the ambiguities or one-to-many mappings that inverse problems typically exhibit. More concretely, we are interested in the conditional modes of the resulting learned CPD, whose values need to be numerically determined. For a network with K components and O outputs, the network maintains K outputs for the mixing coefficients $\pi_k(\mathbf{x})$, K outputs for the tied covariance widths $\sigma_k(\mathbf{x})$, and $K \times O$ outputs for the means $\mu_{kj}(\mathbf{x})$.

5.3.3 Variational Auto-Encoder

The Variational Auto-encoder (VAE) (Kingma and Welling 2013; Rezende et al. 2014) is a directed graphical model (DGM) with Gaussian latent variables (Figure 5-2). The VAE consists of a generative process or component with latent variables $\mathbf{z}$ such that sample z is generated from a prior distribution $z \sim p_\theta(\mathbf{z})$, and data x given by $x \sim p_\theta(\mathbf{x}|\mathbf{z})$. The choice of latent variable $\mathbf{z}$ depends on the latent information that needs to be captured in order to describe the true distribution of the data. The latent description may lie in some high-dimensional space with dependencies

between dimensions. In VAEs, the output distribution is typically Gaussian in that $p(\mathbf{x}|\mathbf{z};\theta)$ is given by

$$p(\mathbf{x}|\mathbf{z};\theta) = \mathcal{N}(f(\mathbf{z};\theta), \sigma^2 * I) \tag{5.9}$$

Ideally, we want to avoid explicitly providing these dependencies, and want to be able to learn the inherent dependencies and structure as informed by the data drawn from it. We start with considering a high-dimensional latent space $\mathbf{z}$ that we can easily sample from, given by its probability density function $p_\theta(\mathbf{z})$. Assuming that a family of distributions $f(\mathbf{z};\theta)$ paramterized by θ exists such that $x \sim f(\mathbf{z};\theta)$, we seek the deterministic function f such that $f(\mathbf{z};\theta)$ is a proxy for sampling from $p_\theta(\mathbf{x})$. More precisely, this is equivalent to maximizing the probability of each sample $\mathbf{x}$ under the generative process given by:

$$p(\mathbf{x}) = \sum \underbrace{p(\mathbf{x}|\mathbf{z};\theta)}_{f(\mathbf{z};\theta)} p(\mathbf{z}) d\mathbf{z} \tag{5.10}$$

In a VAE, the choice of the output distribution is a scaled unit-variance Gaussian $P(\mathbf{x}|\mathbf{z};\theta) = \mathcal{N}(\mathbf{x}|f(\mathbf{z};\theta), \sigma^2 * I)$, with mean $f(\mathbf{z};\theta)$ and diagonal co-variance σ^2. The choice of Gaussian approximation allows the analytic computation of $p(\mathbf{x}|\mathbf{z})$, and can be replaced with an appropriate distribution depending on the expected *output* distribution.

Just as traditional neural networks are known to be universal function approximators, given sufficient depth and non-linearity in the architecture, distributions can be approximated using a similar strategy. Intuitively, the key idea behind VAEs is based on this claim: any arbitrary D-dimensional distribution can be approximated by taking D variables that are normally distributed and passing them through a sufficiently powerful function approximator (chosen such that it approximates the target distribution) (Devroye 1986). They are called auto-encoders since training involves minimizing a similar objective function containing an encoder and a decoder. The encoder (parameterized by θ) *compresses* the high-dimensional input $\mathbf{x}$ to a latent representation given by $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$, and the decoder (parameterized by ϕ) *reconstructs* the original representation from the latent lower-dimensional space $\hat{\mathbf{x}} \sim p_\theta(\mathbf{x}|\mathbf{z})$.

Instead of maximizing the marginal likelihood $p_\theta(\mathbf{x})$, VAEs maximize the vari-

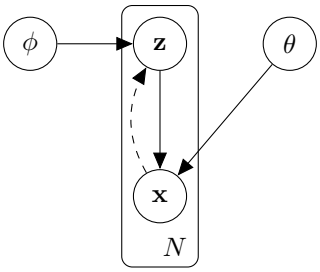


Figure 5-2: **Variational Auto-Encoder (VAE)** ▶ The plate notation indicates that we can sample from the latent variable $\mathbf{z}$ N times, while keeping the model parameters θ fixed. The dashed lines represent the variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ to the intractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$. The generative model parameters θ, and variational parameters ϕ are learned jointly in the Auto-Encoding Variational Bayes estimation.

ational lower-bound of the log-likelihood term $\log p_\theta(\mathbf{x})$,

$$\log p_\theta(\mathbf{x}) = D_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})\big] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x},\mathbf{z})\big] \tag{5.11}$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x},\mathbf{z})\big] \tag{5.12}$$

$$\geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z})\big]}_{\text{Reconstruction Error}} - \underbrace{D_{KL}\big[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\big]}_{\text{Variational Regularization}} \tag{5.13}$$

where the first term corresponds to the cost of reconstructing $\hat{\mathbf{x}}$ given $\mathbf{z}$, while the second KL-divergence term corresponds to the variational regularization term that enforces a prior $p_\theta(\mathbf{z})$ on the proposal distribution $q_\phi(\mathbf{z}|\mathbf{x})$. As described earlier, the choice of priors are application dependent, and is chosen to be Gaussian in the cases described later in this chapter. The parameters of the VAE are estimated efficiently via stochastic gradient variational Bayes (SGVB) (Kingma and Welling 2013) where the above variational lower bound is used as the surrogate objective function. In the Conditional Variational Auto-encoder (C-VAE), the samples are generated from the conditional distribution $\mathbf{x} \sim p_\theta(\mathbf{x}|y,\mathbf{z})$ where y is the conditional variable available during sampling. The variational bound for C-VAE is a simple extension of Equation 5.13,

$$\log p_\theta(\mathbf{x},y) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},y)}\big[\log p_\theta(\mathbf{x}|y,\mathbf{z}) + \log p_\theta(y) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x},y)\big] \tag{5.14}$$

We refer the reader to Kingma and Welling (2013); Kingma et al. (2014); Sohn et al. (2015) for more detailed introduction to variational auto-encoders (VAEs) and their conditional VAE variants (C-VAE).

5.4 Visual Ego-motion Regression

As with most ego-motion estimation solutions, it is imperative to determine the minimal parameterization of the underlying motion manifold. In certain restricted scene structures or motion manifolds, several variants of ego-motion estimation are proposed (Ke and Kanade 2003; Liang and Pears 2002; Scaramuzza 2011; Scaramuzza et al. 2009b). However, we consider the case of modeling cameras with varied optics and hence are interested in determining the full range of ego-motion, often restricted, that induces the pixel-level optical flow. This allows the freedom to model various unconstrained and partially constrained motions that typically affect the overall robustness of existing ego-motion algorithms. While model-based approaches have shown tremendous progress in accuracy, robustness, and run-time performance, a few recent data-driven approaches have been shown to produce equally compelling results (Clark et al. 2016; Costante et al. 2016; Konda and Memisevic 2015). An adaptive and trainable solution for relative pose estimation or ego-motion can be especially advantageous for several reasons: (i) a general-purpose end-to-end trainable model architecture that applies to a variety of camera optics including pinhole, fisheye, and catadioptric lenses; (ii) simultaneous and continuous optimization over both ego-motion estimation and camera parameters (intrinsics and extrinsics that are implicitly modeled); and (iii) joint reasoning over resource-aware computation and accuracy within the same architecture is amenable. We envision that such an approach is especially beneficial in the context of bootstrapped (or weakly-supervised) learning in robots, where the supervision in ego-motion estimation for a particular camera can be obtained from the fusion of measurements from other robot sensors (GPS, wheel encoders etc.).

Our approach is motivated by previous minimally parameterized models (Scaramuzza 2011; Scaramuzza et al. 2009b) that are able to recover ego-motion from a *single tracked feature*. We find this representation especially appealing due to the simplicity and flexibility in *pixel-level* computation. Despite the reduced complexity of the input space for the mapping problem, recovering the full 6-DOF ego-motion is ill-posed due to the inherently under-constrained system. However, it has been previously shown that under non-holonomic vehicle motion, camera ego-motion may be fully recoverable up to a sufficient degree of accuracy using a single point (Scaramuzza 2011; Scaramuzza et al. 2009b).

We now focus on the specifics of the ego-motion regression objective. Due to the under-constrained nature of the prescribed regression problem, the pose esti-

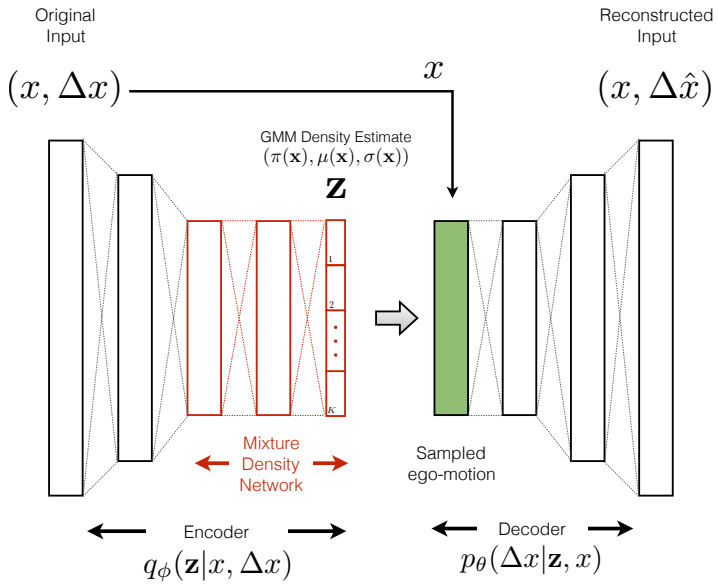


Figure 5-3: **Visual Ego-motion Learning Architecture** ▶ We propose a visual ego-motion learning architecture that maps optical flow vectors (derived from feature tracking in an image sequence) to an ego-motion density estimate via a Mixture Density Network (MDN). By modeling the architecture as a Conditional Variational Autoencoder (C-VAE), our model is able to provide introspective reasoning and prediction for scene-flow conditioned on the ego-motion estimate and input feature location.

mation is modeled as a density estimation problem over the range of possible ego-motions[1], conditioned on the input flow features. It is important to note that the output of the proposed model is a density estimate $p(\hat{\mathbf{z}}_{t-1,t}|\mathbf{x}_{t-1,t})$ for every feature tracked between subsequent frames.

5.4.1 Density Estimation for Ego-motion

In typical associative mapping problems, the joint probability density $p(\mathbf{x}, \mathbf{z})$ is decomposed into the product of two terms: (i) $p(\mathbf{z}|\mathbf{x})$: the conditional density of the target pose $\mathbf{z} \in SE(3)$ conditioned on the input feature correspondence $\mathbf{x} = (\boldsymbol{x}, \Delta\boldsymbol{x})$ obtained from sparse optical flow (via the KLT tracker) (Birchfield 2007); and (ii) $p(\mathbf{x})$: the unconditional density of the input data $\mathbf{x}$. While we are particularly interested in the first term $p(\mathbf{z}|\mathbf{x})$ that predicts the range of possible values for $\mathbf{z}$ given new values of $\mathbf{x}$, we can observe that the density $p(\mathbf{x}) = \sum_z p(\mathbf{x}, \mathbf{z})d\mathbf{z}$ provides a measure of how well the prediction is captured by the trained model.

The critical component in estimating the ego-motion belief is the ability to accurately predict the conditional probability distribution $p(\mathbf{z}|\mathbf{x})$ of the pose estimates

[1] Although the parameterization is maintained as $SE(3)$, it is important to realize that most autonomous ground vehicles predominantly traverse a lower-dimensional ($SE(2)$) motion manifold.

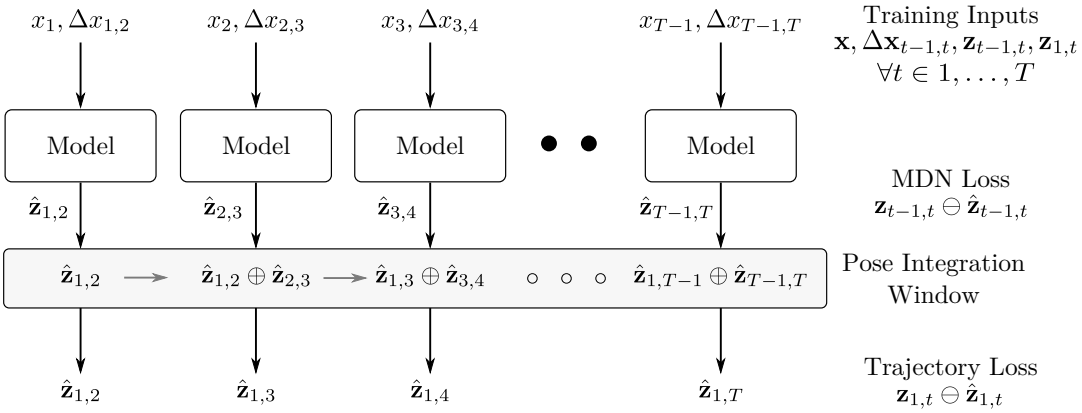


Figure 5-4: **Windowed trajectory optimization** ▶ An illustration of the losses introduced for training frame-to-frame ego-motion (*local*) and windowed ego-motion (*global*) by compounding the poses determined from each of the individual frame-to-frame measurements.

that is induced by the given input feature $\boldsymbol{x}$ and the flow $\Delta\boldsymbol{x}$. Due to its powerful and rich modeling capabilities, we use a *Mixture Density Network* (MDN) to parametrize the conditional density estimate. The powerful representational capacity of neural networks coupled with rich probabilistic modeling that GMMs admit, allows us to model multi-valued or multi-modal beliefs that typically arise in inverse problems such as visual ego-motion. For each of the N_F input flow features $\mathbf{x}_i$ extracted via KLT, the conditional probability density of the target pose data $\mathbf{z}_i$ is given by $p(\mathbf{z}_i|\mathbf{x}_i)$, modeled directly with an K-component mixture density network.

$$p(\mathbf{z}_i|\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k(\mathbf{x}_i)\mathcal{N}(\mathbf{z}_i|\mu_k(\mathbf{x}_i), \sigma_k^2(\mathbf{x}_i)) \tag{5.15}$$

The proposed model is learned end-to-end by maximizing the data log-likelihood, or alternatively minimizing the negative log-likelihood (denoted as $\mathcal{L}_{\mathcal{MDN}}$ in Eqn 5.8), given the N_F input feature tracks $(\mathbf{x}_1 \ldots \mathbf{x}_{N_F})$ and expected ego-motion estimate $\mathbf{z}$. The resulting ego-motion density estimates $p(\mathbf{z}_i|\mathbf{x}_i)$ obtained from each individual flow vectors $\mathbf{x}_i$ are then fused by taking the product of their densities. However, to maintain tractability of density products, only the mean and covariance corresponding to the largest mixture coefficient (i.e. most likely mixture mode) for each feature is considered for subsequent trajectory optimization.

5.4.2 Trajectory Optimization

While minimizing the MDN loss ($\mathcal{L}_{MDN}$) as described above provides a reasonable regressor for ego-motion estimation, it is evident that optimizing frame-to-frame measurements does not ensure long-term consistencies in the ego-motion trajectories obtained by integrating these regressed estimates. As one expects, the integrated trajectories are sensitive to even negligible biases in the ego-motion regressor.

Two-stage optimization: To circumvent the aforementioned issue, we introduce a second optimization stage that jointly minimizes the *local* objective ($\mathcal{L}_{MDN}$) with a *global* objective that minimizes the error incurred between the overall trajectory and the trajectory obtained by integrating the regressed pose estimates obtained via the *local* optimization. This allows the *global* optimization stage to have a warm-start with an almost correct initial guess for the network parameters.

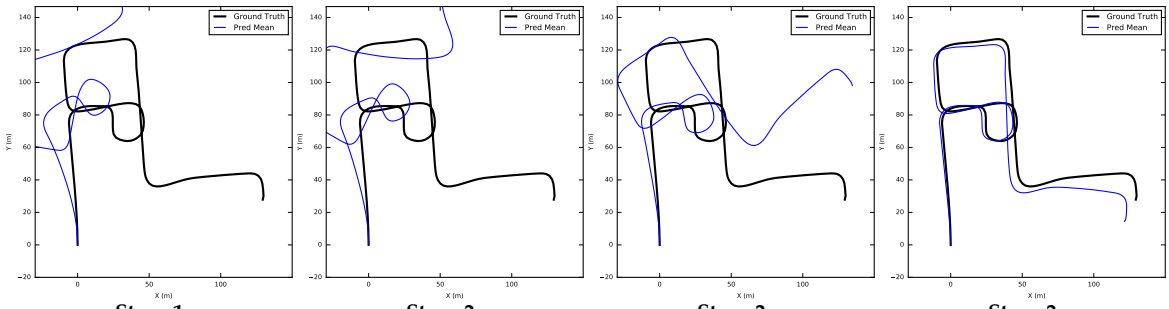
As seen in Eqn 5.17, $\mathcal{L}_{TRAJ}$ pertains to the overall trajectory error incurred by integrating the individual regressed estimates over a batched window (we typically consider 200 to 1000 frames). This allows us to fine-tune the regressor to predict valid estimates that integrate towards accurate long-term ego-motion trajectories. As expected, the model is able to roughly learn the curved trajectory path, however, it is not able to make accurate predictions when integrated for longer time-windows (due to the lack of the *global* objective loss term in Stage 1). Figure 5-4 provides a high-level overview of the input-output relationships of the training procedure, including the various network losses incorporated in the ego-motion encoder/regressor. We refer the reader to Figure 5-5 where we illustrate this two-stage approach over a simulated dataset (Zhang et al. 2016).

In Eqn 5.17, $\hat{\mathbf{z}}_{t-1,t}$ is the frame-to-frame ego-motion estimate and the regression target/output of the MDN function f^{vo} given by

$$f^{vo} : \mathbf{x} \mapsto \Big(\mu(\mathbf{x}_{t-1,t}), \sigma(\mathbf{x}_{t-1,t}), \pi(\mathbf{x}_{t-1,t})\Big) \tag{5.16}$$

where $\hat{\mathbf{z}}_{1,t}$ is the overall trajectory predicted by integrating the individually regressed frame-to-frame ego-motion estimates and is defined by $\hat{\mathbf{z}}_{1,t} = \hat{\mathbf{z}}_{1,2} \oplus \hat{\mathbf{z}}_{2,3} \oplus \cdots \oplus \hat{\mathbf{z}}_{t-1,t}$.

$$\mathcal{L}_{\text{ENC}} = \underbrace{\sum_t \mathcal{L}^t_{MDN}\Big(f^{vo}(\mathbf{x}), \mathbf{z}_{t-1,t}\Big)}_{\text{MDN Loss}} + \underbrace{\sum_t \mathcal{L}^t_{TRAJ}(\mathbf{z}_{1,t} \ominus \hat{\mathbf{z}}_{1,t})}_{\text{Overall Trajectory Loss}} \tag{5.17}$$



Stage 1 (Final) | **Stage 2** (Epoch 4) | **Stage 2** (Epoch 8) | **Stage 2** (Epoch 18)

Figure 5-5: **Two-stage Optimization ►** An illustration of the two-stage optimization procedure. The *first* column shows the final solution after the first stage. Despite the minimization of the MDN loss term, the integrated trajectory is clearly biased and poorly matches the expected trajectory result. The *second*, *third* and *fourth* column shows the gradual improvement of the second stage (global minimization including the overall trajectory loss) and matches the expected ground truth trajectory better (i.e. estimates the regressor biases better).

5.4.3 Bootstrapped Learning for Ego-motion Estimation

Typical robot navigation systems consider the fusion of visual odometry estimates with other modalities including estimates derived from wheel encoders, IMUs, GPS etc. Considering odometry estimates (for e.g. from wheel encoders) as-is, the uncertainties in open-loop chains grow in an unbounded manner. Furthermore, relative pose estimation may also be inherently biased due to calibration errors that eventually contribute to the overall error incurred. GPS, despite being noise-ridden, provides an absolute sensor reference measurement that is especially complementary to the open-loop odometry chain maintained with odometry estimates. The probabilistic fusion of these two relatively uncorrelated measurement modalities allows us to recover a sufficiently accurate trajectory estimate that can be directly used as ground truth data $\mathbf{z}$ (in Figure 5-7) for our supervised regression problem.

The indirect recovery of training data from the fusion of other sensor modalities in robots falls within the *self-supervised or bootstrapped* learning paradigm. We envision this capability to be especially beneficial in the context of life-long learning in future autonomous systems. Using the fused and optimized pose estimates $\mathbf{z}$ (recovered from GPS and odometry estimates), we are able to recover the required input-output relationships for training visual ego-motion for a completely new sensor (as illustrated in Figure 5-7). Figure 5-12 illustrates the realization of the learned model in a typical autonomous system where it is treated as an additional sensor source. Through experiments 5.5.3, we illustrate this concept with the recovery of ego-motion in a robot car equipped with a GPS/INS unit and a single camera.

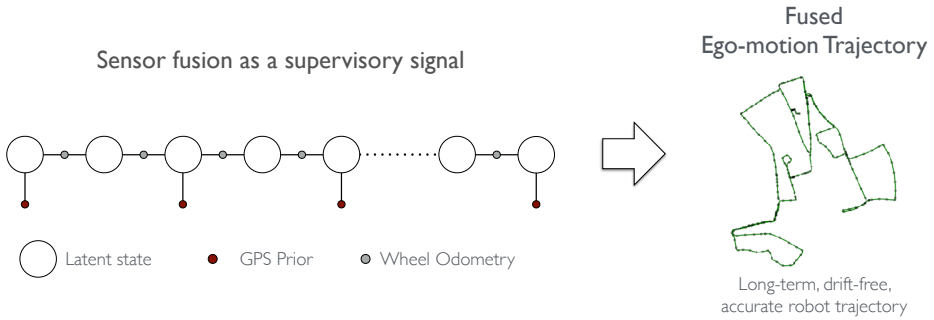


Figure 5-6: **Ground-truth Trajectory Generation ►** By fusing odometric measurements from wheel odometry or IMU with intermittent GPS measurements, standard filtering-based techniques can be leveraged to recover long-term, and drift-free trajectory of the vehicle in a fully automatic fashion. We directly use this recovered vehicle trajectory as "*ground-truth*" for subsequent visual ego-motion regression.

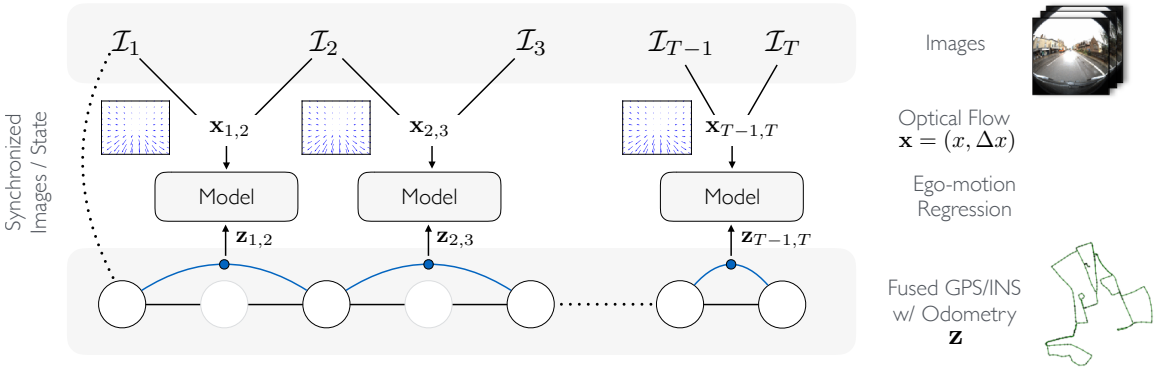


Figure 5-7: **Bootstrapped Ego-motion Regression ►** Illustration of the bootstrap mechanism whereby a robot self-supervises the proposed ego-motion regression task in a new camera sensor by fusing information from other sensor sources such as GPS and INS.

5.4.4 Introspective Reasoning for Scene-Flow Prediction

Scene flow is a fundamental capability that provides directly measurable quantities for ego-motion analysis. The flow observed by sensors mounted on vehicles is a function of the inherent scene depth, the relative ego-motion undergone by the vehicle, and the intrinsic and extrinsic properties of the camera used to capture it. As with any measured quantity, one needs to deal with sensor-level noise propagated through the model in order to provide robust estimates. While the input flow features are an indication of ego-motion, some of the features may be corrupted due to lack of or ambiguous visual texture or due to flow induced by the dynamics of objects other than the ego-motion itself. Evidently, we observe that the dominant flow is generally induced by ego-motion itself, and it is this flow that we intend to fully recover via a conditional variational auto-encoder (C-VAE). By inverting the regression problem, we develop a generative model able to predict the most-

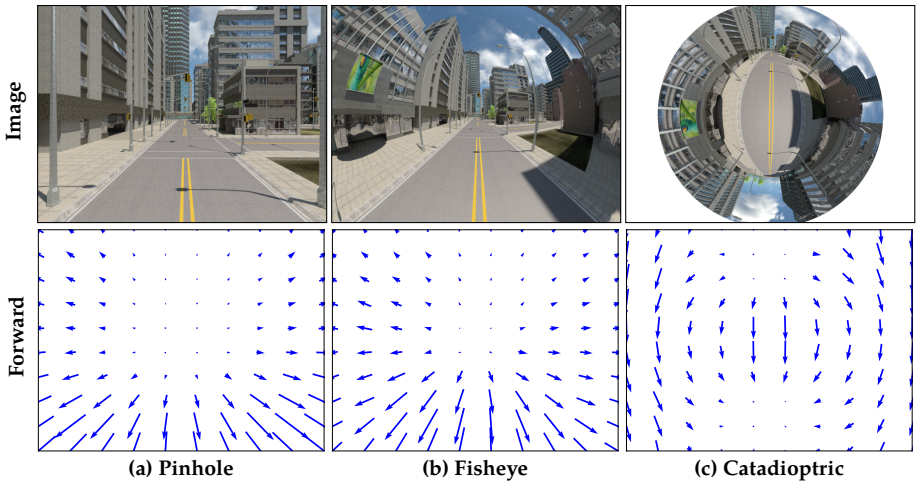


Figure 5-8: **Introspective reasoning for scene-flow prediction** ▶ Illustrated above are the dominant flow vectors corresponding to scene-flow given the corresponding ego-motion. While this module is currently only used for introspection purposes, we expect it to be critical in outlier rejection for robust ego-motion estimation. **Row 1**: Sample image from camera, **Row 2**: Flow induced by forward motion

likely flow $\Delta\hat{x}$ induced given an ego-motion estimate $\mathbf{z}$, and feature location x. We propose a scene-flow specific autoencoder that encodes the implicit egomotion observed by the sensor, while jointly reasoning over the latent depth of each of the individual tracked features. In order to make the entire architecture fully differentiable, only the dominant mode (mode corresponding to largest mixture coefficient) is sampled to recover the induced ego-motion flow. While the re-parameterization trick, applied in VAEs to allow the back-propagation over the stochastic nodes, can be applied to the full mixture model (Graves 2016), we only consider recovering the flow given the dominant mode in this work.

$$\mathcal{L}_{\text{CVAE}} = \underbrace{\mathbb{E}\big[\log p_\theta(\Delta x|\mathbf{z}, x)\big]}_{\text{Reconstruction Error}} - \underbrace{D_{KL}\big[q_\phi(\mathbf{z}|x, \Delta x)||p_\theta(\mathbf{z}|x)\big]}_{\text{Variational Regularization}} \tag{5.18}$$

Through the proposed denoising autoencoder model, we are also able to attain an introspection mechanism for the presence of outliers. We incorporate this additional module via an auxiliary loss as specified in Eqn 5.18. An illustration of these flow predictions are shown in Figures 5-8 and 5-9.

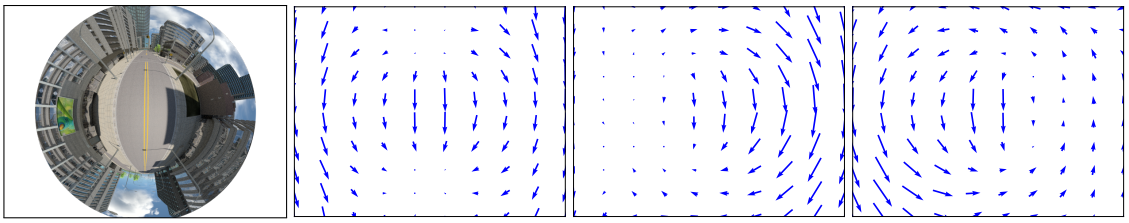


Figure 5-9: **Scene-flow prediction with odometry ►** Illustrated above are the dominant flow vectors corresponding to scene-flow in a *catadioptric* lens conditioned on the ego-motion direction.

5.5 Experiments and Results

In this section, we provide detailed experiments on the performance, robustness and flexibility of our proposed approach on various datasets. Our approach differentiates itself from existing solutions on various fronts as shown in Table 5.1. We evaluate the performance of our proposed approach on various publicly-available datasets including the KITTI dataset (Geiger et al. 2012), the Multi-FOV synthetic dataset (Zhang et al. 2016) (pinhole, fisheye, and catadioptric lenses), an omnidirectional-camera dataset (Schönbein and Geiger 2014), and on the Oxford Robotcar 1000km Dataset (Maddern et al. 2016).

Navigation solutions in autonomous systems today typically fuse various modalities including GPS, odometry from wheel encoders and INS to provide robust trajectory estimates over extended periods of operation. We provide a similar solution by leveraging the learned ego-motion capability described in this work, and fuse it with intermittent GPS updates[2] (Section 5.5.1). While maintaining similar performance capabilities (Table 5.2), we re-emphasize the benefits of our approach over existing solutions:

- **Versatile**: With a fully trainable model, our approach is able to simultaneously reason over both ego-motion and implicitly modeled camera parameters (*intrinsics* and *extrinsics*). Furthermore, online calibration and parameter tuning is implicitly encoded within the same learning framework.
- **Model-free**: Without imposing any constraints on the type of camera optics, our approach is able to recover ego-motions for a variety of camera models including *pinhole*, *fisheye* and *catadioptric* lenses. (Section 5.5.2)
- **Bootstrapped training and refinement**: We illustrate a bootstrapped learning

[2]For evaluation purposes only, the absolute ground truth locations were added as weak priors on datasets without GPS measurements

example whereby a robot self-supervises the proposed ego-motion regression task by fusing information from other sensor sources including GPS and INS (Section 5.5.3)

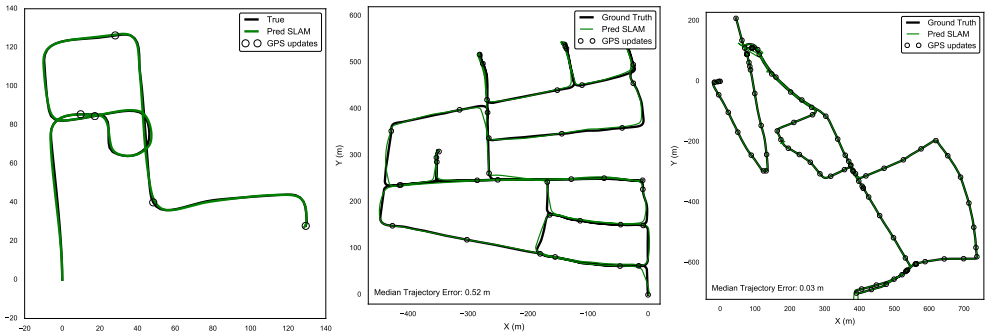
- **Introspective reasoning for scene-flow prediction**: Via the C-VAE generative model, we are able to reason/introspect over the predicted flow vectors in the image given an ego-motion estimate. This provides an obvious advantage in *robust* outlier detection and identifying dynamic objects whose flow vectors need to be disambiguated from the ego-motion scene flow (Figure 5-8)

5.5.1 Evaluating Ego-motion Performance with Sensor Fusion

In this section, we evaluate our approach against a few state-of-the-art algorithms for monocular visual odometry (Kitt et al. 2010). On the KITTI dataset (Geiger et al. 2012), the pre-trained estimator is used to robustly and accurately predict ego-motion from KLT features tracked over the dataset image sequence. The frame-to-frame ego-motion estimates are integrated for each session to recover the full trajectory estimate and simultaneously fused with intermittent GPS updates (incorporated every 150 frames). In Figure 5-10, we show the qualitative performance in the overall trajectory obtained with our method. The entire pose-optimized trajectory is compared against the ground truth trajectory. The translational errors are computed for each of the ground truth and prediction pose pairs, and their median value is reported in Table 5.2 for a variety of datasets with varied camera optics.

5.5.2 Varied Camera Optics

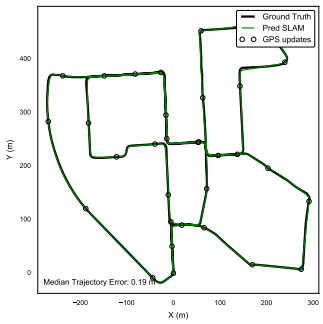
Most of the existing implementations of VO estimation are restricted to a class of camera optics, and generally avoid implementing a general-purpose VO estimator for varied camera optics. Our approach on the other hand, has shown the ability to provide accurate VO with intermittent GPS trajectory estimation while simultaneously being applicable to a varied range of camera models. In Figure 5-11, we compare with intermittent GPS trajectory estimates for all three camera models, and verify their performance accuracy compared to ground truth. In our experiments, we found that while our proposed solution was sufficiently powerful to model different camera optics, it was significantly better at modeling pinhole lenses as compared to fisheye and catadioptric cameras (See Table 5.2). In future work, we would



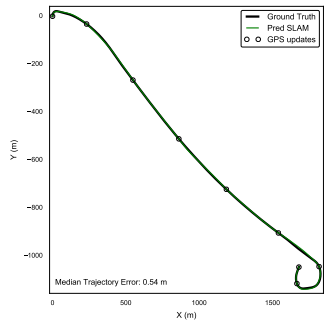
(a) Multi-FOV Synthetic Dataset

(b) Omnicam Dataset

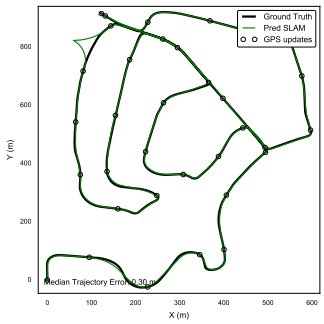
(c) Oxford 1000km



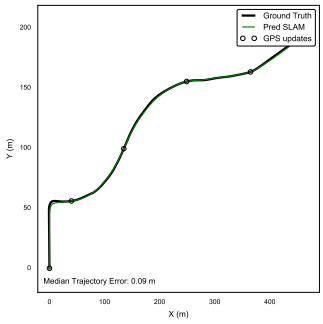
(d) KITTI 00



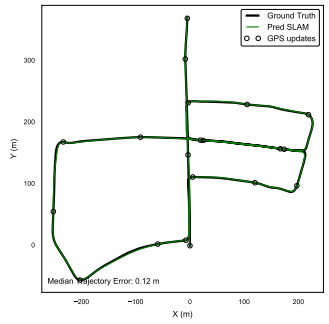
(e) KITTI 01



(f) KITTI 02

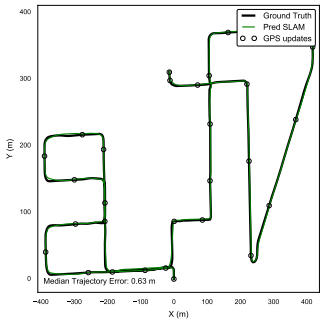


(g) KITTI 03

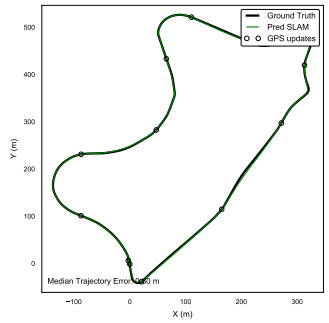


(h) KITTI 05

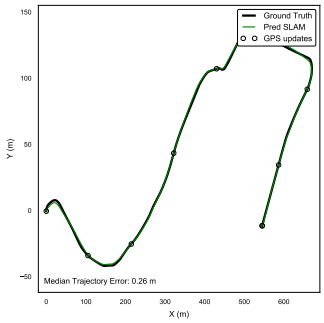
(i) KITTI 07



(j) KITTI 08



(k) KITTI 09



(l) KITTI 10

Figure 5-10: **Sensor fusion with learned ego-motion** ▶ On fusing our proposed VO method with intermittent GPS updates (every 150 frames, black circles), the pose-graph optimized ego-motion solution (in green) achieves sufficiently high accuracy relative to ground truth. We test on a variety of publicly-available datasets including **(a)** Multi-FOV synthetic dataset (Zhang et al. 2016) (*pinhole* shown above), **(b)** an omnidirectional-camera dataset (Schönbein and Geiger 2014), **(c)** Oxford Robotcar 1000km Dataset (Maddern et al. 2016) (2015-11-13-10-28-08) **(d-l)** KITTI dataset (Geiger et al. 2012). *Weak supervision* such as GPS measurements can be especially advantageous in recovering improved estimates for localization, while simultaneously minimizing uncertainties associated with pure VO-based approaches.

Dataset	Camera Optics	Median Trajectory Error
KITTI-00	Pinhole	0.19 m
KITTI-01	Pinhole	0.54 m
KITTI-02	Pinhole	0.30 m
KITTI-03	Pinhole	0.09 m
KITTI-04	Pinhole	0.02 m
KITTI-05	Pinhole	0.12 m
KITTI-06	Pinhole	0.16 m
KITTI-07	Pinhole	0.18 m
KITTI-08	Pinhole	0.63 m
KITTI-09	Pinhole	0.30 m
KITTI-10	Pinhole	0.26 m
Multi-FOV (Zhang et al. 2016)	Pinhole	0.18 m
Multi-FOV (Zhang et al. 2016)	Fisheye	0.48 m
Multi-FOV (Zhang et al. 2016)	Catadiopt	0.36 m
Omnidirectional (Schönbein and Geiger 2014)	Catadiopt	0.52 m
Oxford 1000km† (Maddern et al. 2016)	Pinhole	0.03 m

Table 5.2: **Trajectory prediction performance ►** An illustration of the trajectory prediction performance of our proposed ego-motion approach when fused with intermittent GPS updates (every 150 frames). The errors are computed across the entire length of the optimized trajectory and ground truth. For Oxford 1000km dataset, we only evaluate on a single session (2015-11-13-10-28-08 [80GB]: †Stereo Centre)

like to investigate further extensions that improve the accuracy for both fisheye and catadioptric lenses.

5.5.3 Self-supervision via Synchronized Cross-Modal Learning

We envision the capability of robots to self-supervise tasks such as visual ego-motion estimation to be especially beneficial in the context of life-long learning and autonomy. We experiment and validate this concept through a concrete example using the 1000km Oxford Robot Car dataset (Maddern et al. 2016). We train the task of visual ego-motion on a new camera sensor by leveraging the fused GPS and INS information collected on the robot car as ground truth trajectories (6-DOF), and extracting feature trajectories (via KLT) from image sequences obtained from the new camera sensor. The timestamps from the cameras are synchronized with respect to the timestamps of the fused GPS and INS information, in order to obtain a one-to-one mapping for training purposes. We train on the `stereo_centre` *(pinhole)* camera dataset and present our results in Table 5.2. As seen in Figure 5-10, we are able to achieve considerably accurate long-term state estimates by fusing our proposed visual ego-motion estimates with even sparser GPS updates (every 2-3 seconds, instead of 50Hz GPS/INS readings). This allows the robot to reduce its reliance on GPS/INS alone to perform robust, long-term trajectory estimation.

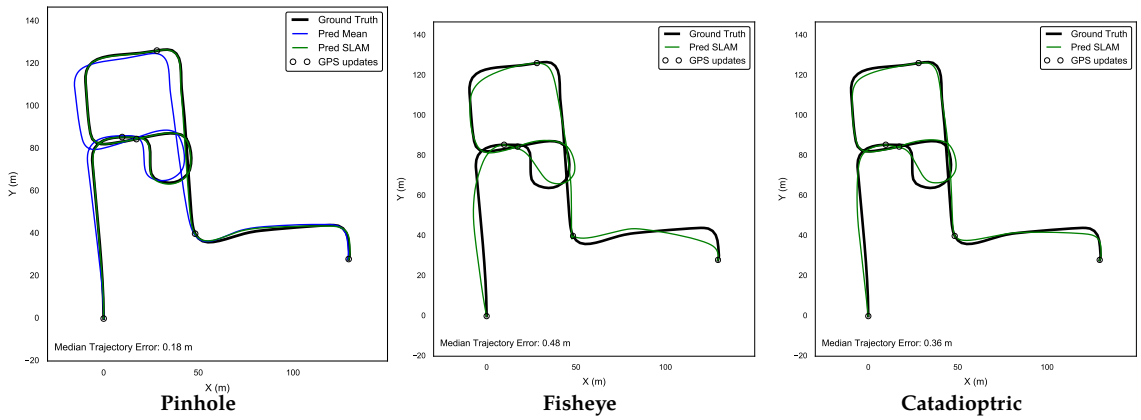


Figure 5-11: **Varied camera optics ►** An illustration of the performance of our general-purpose approach for varied camera optics (pinhole, fisheye, and catadioptric lenses) on the Multi-FOV synthetic dataset (Zhang et al. 2016). Without any prior knowledge on the camera optics, or the mounting configuration (extrinsics), we are able to robustly and accurately recover the full trajectory of the vehicle (with intermittent GPS updates every 500 frames).

5.5.4 Implementation Details

In this section we describe the details of our proposed model, training methodology and parameters used. The input $\mathbf{x} = (\boldsymbol{x}, \Delta\boldsymbol{x})$ to the density-based ego-motion estimator are feature tracks extracted via (Kanade-Lucas-Tomasi) KLT feature tracking over the raw camera image sequences. The input feature positions and flow vectors are normalized to be the in range of $[-1, 1]$ using the dimensions of the input image. We evaluate sparse LK (Lucas-Kanade) optical flow over 7 pyramidal scales with a scale factor of $\sqrt{2}$. As the features are extracted, the corresponding robot pose (either available via GPS or GPS/INS/wheel odometry sensor fusion) is synchronized and recorded in $SE(3)$ for training purposes. The input KLT features, and the corresponding relative pose estimates used for training are parameterized as $\mathbf{z} = (\mathbf{t}, \mathbf{r}) \in \mathbb{R}^6$, with a Euclidean translation vector $\mathbf{t} \in \mathbb{R}^3$ and an Euler rotation vector $\mathbf{r} \in \mathbb{R}^3$. We once again refer the reader to Figure 5-12 illustrating the deployment of the learned ego-motion model fused with GPS.

Network and training: The proposed architecture consists of a set of fully-connected stacked layers (with 1024, 128 and 32 units) followed by a Mixture Density Network with 32 hidden units and 5 mixture components (K). Each of the initial fully-connected layers implement *tanh* activation after it, followed by a dropout layer with a dropout rate of 0.1. The final output layer of the MDN (a^{π}, a^{μ}, a^{σ}) consists of $(O + 2) * K$ outputs where O is the desired number of states estimated.

The network is trained (in Stage 1) with loss weights of 10, 0.1, 1 corresponding to the losses $\mathcal{L}_{MDN}, \mathcal{L}_{TRAJ}, \mathcal{L}_{CVAE}$ described in previous sections. The training

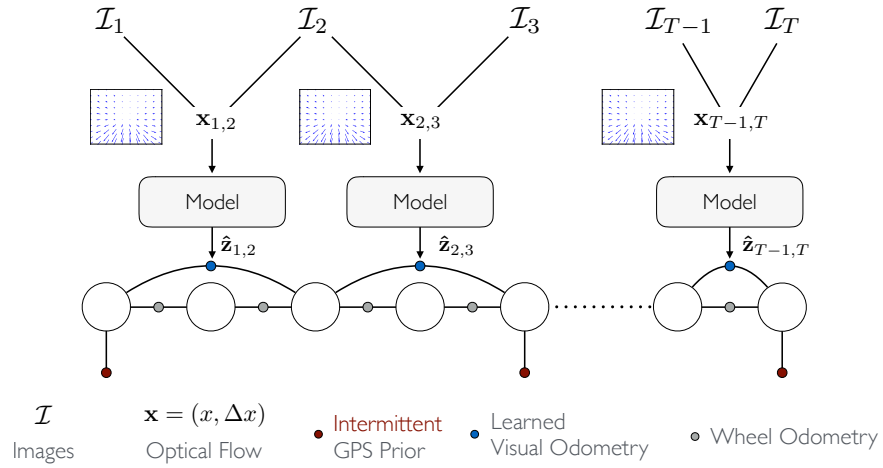


Figure 5-12: **Learned Ego-motion Deployment** ▶ During model deployment, the learned visual-egomotion model provides valuable relative pose constraints to augment the standard navigation-based sensor fusion (GPS/INS and wheel encoder odometry fusion).

data is provided in batches of 100 frame-to-frame subsequent image pairs, each consisting of approximately 50 randomly sampled feature matches via KLT. The learning rate is set to $1e{-}3$ with Adam (Kingma and Ba 2014) as the optimizer. On the synthetic Multi-FOV dataset and the KITTI dataset, training for most models took roughly an hour and a half (3000 epochs) independent of the KLT feature extraction step. For most datasets including KITTI and the Oxford 1000km, we train on 2-5 data sessions collected from the vehicle, and test on a completely new session. However on the synthetic Multi-FOV dataset, we train and test on the same set to validate our proposed method.

Two-stage optimization: We found the one-shot joint optimization of the *local* ego-motion estimation and *global* trajectory optimization to have sufficiently low convergence rates during training. One possible explanation is the high sensitivity of the loss weight parameters that is used for tuning the local and global losses into a single objective. As previously addressed in Section 5.4.2, we separate the training into two stages thereby alleviating the aforementioned issues, and maintaining fast convergence rates in Stage 1. Furthermore, we note that during the second stage, it only requires a few tens of iterations for sufficiently accurate ego-motion trajectories. In order to optimize over a larger time-window in stage 2, we set the batch size to 1000 frame-to-frame image matches, again randomly sampled from the training set as before. Due to the large integration window and memory limitations, we train this stage purely on the CPU for only 100 epochs each taking roughly 30s per epoch. Additionally, in stage 2, the loss weights for $\mathcal{L}_{TRAJ}$ are increased to 100 in order to have faster convergence to the *global* trajectory. The remaining loss weights

are left unchanged.

Trajectory fusion: We use GTSAM[3] to construct the underlying factor graph for pose-graph optimization. Odometry constraints obtained from the frame-to-frame ego-motion are incorporated as a 6-DOF constraint parameterized in $SE(3)$ with $1 * 10^{-3}$ rad rotational noise and $5 * 10^{-2}$ m translation noise. As with typical autonomous navigation solutions, we expect measurement updates in the form of GPS (absolute reference updates) in order to correct for the long-term drift incurred in open-loop odometry chains. We incorporate absolute prior updates only every 150 frames, with a weak translation prior of 0.01 m. The constraints are incrementally added and solved using iSAM2 (Kaess et al. 2012) as the measurements are streamed in, with updates performed every 10 frames.

While the proposed MDN is parameterized in Euler angles, the *trajectory integration module* parameterizes the rotation vectors in quaternions for robust and unambiguous long-term trajectory estimation. All the rigid body transformations are implemented directly in Tensorflow for pure-GPU training support.

Run-time performance: We are particularly interested in the run-time / test-time performance of our approach on CPU architectures for mostly resource-constrained settings. Independent of the KLT feature tracking run-time, we are able to recover ego-motion estimates at roughly 3ms on a consumer-grade Intel(R) Core(TM) i7-3920XM CPU @ 2.90GHz.

Source code and Pre-trained weights: We implemented the MDN-based ego-motion estimator with Keras and Tensorflow, and trained our models using a combination of CPUs and GPUs (NVIDIA Titan X). All the code was trained on an server-grade Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz and tested on the same consumer-grade machine as mentioned above to emulate potential real-world use-cases. The source code and pre-trained models used will be made available shortly[4].

5.6 Discussion and Future Work

The initial results in bootstrapped learning for visual ego-motion has motivated new directions towards life-long learning in autonomous robots. While our visual ego-motion model architecture is shown to be sufficiently powerful to recover ego-motions for non-linear camera optics such as fisheye and catadioptric lenses,

[3] http://collab.cc.gatech.edu/borg/gtsam

[4] See https://github.com/spillai/learning-egomotion

we continue to investigate further improvements to match existing state-of-the-art models for these lens types. Our current model does not capture distortion effects yet, however, this is very much a future direction we would like to take.

Active Calibration and Monitoring One of the benefits of self-supervised learning in the context of visual ego-motion is the ability to actively monitor calibration errors in sensors, and their task-related accuracy with more experience gathered. This same framework also allows us to incorporate active monitoring and fault tolerance mechanisms with little modification to our proposed solution, and make it feasible to correct for these calibration errors on-the-fly. We envision that over time, robots and their sensors will need to be constantly monitored and re-calibrated, and that these self-supervised solutions that require minimal human supervision can be especially valuable in ensuring robust and long-term operation.

Keyframe-based Tracking While sequential frame-to-frame feature tracking and pose estimation are sufficient for egomotion estimation, one of the major disadvantages of performing sequential (*filtering-based*) VO is the unbounded uncertainty propagation with every erroneous measurement. As with any open-loop odometry chain, the pose uncertainty is compounded with every subsequent relative pose measurement, while the same features are being tracked across multiple frames. In order to avoid frame-to-frame matching and growing pose uncertainty, certain frames can be skipped until the camera has observed considerable movement, before a new reference frame, called *keyframe*, is selected. During operation, features from the previously instantiated keyframe is matched against the current frame to determine the relative 6-DOF pose. This continues until a new keyframe is added and the relative pose measurement is maintained and propagated to provide the VO estimate. Due to its increased accuracy and robustness, keyframe-based methods has become a standard step in VO and vSLAM applications (Scaramuzza and Fraundorfer 2011). In the future, we intend to investigate this concept, and expect treating keyframe-instantiation as a learning problem in itself, further reducing the overall pose-estimation error incurred in visual odometry estimation.

Resource-Constrained Estimation Another consideration is the resource-constrained setting, where the optimization objective incorporates an additional regularization term on the number of parameters used, and the computation load consumed. This ties again to keyframe-selection strategies, where we want to minimize computational resources on tracking every frame, while maintaining sufficient accuracy in pose estimation. Moreover, we expect these models for ego-motion can be further fine-tuned, and compressed for computational efficiency while being tailored to the

desired operational regime of the robot. We hope for this resource-aware capability to transfer to real-world limited-resource robots and to have a significant impact on the adaptability of robots for long-term autonomy.

5.7 Chapter Summary

While many visual ego-motion algorithm variants have been proposed in the past decade, we envision that a fully end-to-end trainable algorithm for generic camera ego-motion estimation shall have far-reaching implications in several domains, especially autonomous systems. Furthermore, we expect our method to seamlessly operate under resource-constrained situations in the near future by leveraging existing solutions in model reduction and dynamic model architecture tuning. With the availability of multiple sensors on these autonomous systems, we also foresee our approach to bootstrapped task (visual ego-motion) learning to potentially enable robots to learn from experience, and use the new models learned from these experiences to encode redundancy and fault-tolerance all within the same framework.

Chapter 6

Self-Supervised Visual Place Recognition in Robots

Both visual odometry estimation and place recognition are two core competencies in any Visual SLAM front-end. As addressed earlier, place recognition is an essential capability that allows mobile robots to disambiguate and identify previously visited locations, and thereby significantly improve their model of where they are in the world. More generally, we are interested in understanding what makes two images "similar" in the context of mobile robots, in that they are captured from a very similar vantage point and location. In the previous chapter, we proposed a bootstrapping mechanism that leverages a GPS-aided SLAM solution to supervise the task of visual ego-motion estimation. Following this similar concept, we explore the ability to *self-supervise place recognition* in mobile robots. In this work, we consider the space of all image-based descriptors and provide a *bootstrapped mechanism* to gauge the similarity between image descriptions specifically tailored for the task of place-recognition in mobile robots.

6.1 Introduction

Visual place recognition, or more commonly referred to as loop-closure recognition, is a critical component in robot navigation that enables it to visually reestablish previously visited locations and simultaneously use this information to correct the drift that was incurred in the dead-reckoned estimate. Loop-closure recognition is a well-studied topic with several approaches leveraging sensor-specific features for the task at hand. However, even state-of-the-art methods today use

hand-tuned features and matching techniques to implement their vision-based loop-closure mechanisms. With the growing sensor modalities on robotic systems, maintaining several variants of the hand-engineered front-ends becomes increasingly tedious and difficult. The results are less than optimal since certain feature representations such as Convolutional Neural Networks (CNNs) extract (Zhou et al. 2014b; 2016b) for example, are generally optimized for the image classification task. Alternatively, we could learn a metric of similarity for the purposes of localization i.e. identifying a mapping where features extracted from identical locations lie closer to each other, and those extracted from dissimilar places lie farther away from each other. Furthermore, we would like to determine a calibrated distance metric that provides a probabilistic measure of similarity such that they can be readily deployed in safety-critical systems where modeling these probabilities can be especially valuable. To alleviate this growing concern, we envision robots to *self-supervise* the task of visual loop-closure recognition in newer sensors by bootstrapping their existing localization and mapping capabilities.

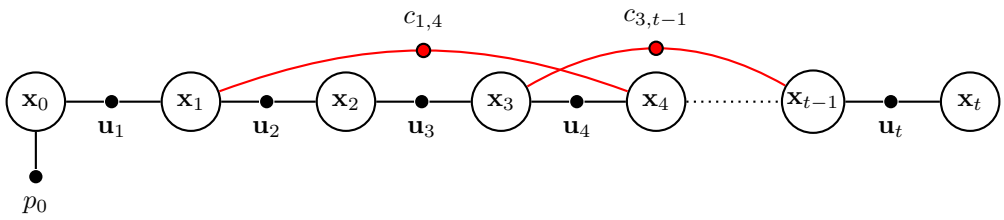


Figure 6-1: **Visual Loop-Closure Recognition Learning** ▶ In a typical factor-graph formulation of Pose-Graph SLAM, the vision-based loop-closure recognition contributes to relative-pose constraints $c_{j,k}$ (*in red*) between temporally-distant nodes. This chapter focuses on identifying these loop-closure constraints by describing and indexing images in an embedded feature space that can be self-supervised to perform accurate loop-closure retrieval.

6.2 Related Work

Visual place recognition in the context of vision-based navigation is a well studied problem in the robotics and computer vision literature (Lowry et al. 2016). In order to identify previously visited locations the system needs to be able to extract salient cues from an image that describes the content contained within it. Extracting an appropriate set of cues can be especially challenging when building robust systems that operate for extremely long periods of time. Typically, the same place may be significantly different from its previous appearance due to various factors such as variations in lighting (e.g. sunny, cloudy, rainy etc), observed viewpoint

(e.g. viewing from opposite directions, viewing from significantly different vantage points), or perceptual aliasing (e.g. facing and seeing a brick-wall elsewhere). *These properties make it extremely challenging to hand-engineer solutions that robustly operate in a wide range of scenarios*. For a detailed overview of existing visual place recognition methods and their capabilities, we refer the reader to a recent literature survey (Lowry et al. 2016) on visual place recognition.

Local and Global methods Some of the earliest forms of visual place recognition entailed directly observing pixel intensities in the image and measuring their correlation with an existing set of images maintained in an efficient database. In order to be invariant to viewpoint changes, subsequent works (Angeli et al. 2008; Churchill and Newman 2012; Cummins and Newman 2011; Konolige and Agrawal 2008; Košecká et al. 2005; Mei et al. 2010; Sünderhauf and Protzel 2011) proposed using low-level *locally-invariant* descriptors such as SIFT (Lowe 1999), SURF (Bay et al. 2006), ORB (Rublee et al. 2011) and others (Tuytelaars et al. 2008). These descriptions are aggregated into a single high-dimensional feature vector for the entire image via Bag-of-Visual-Words (BoVW) (Philbin et al. 2007; Sivic and Zisserman 2003), VLAD (Arandjelovic and Zisserman 2013; Jégou et al. 2010) or Fisher Vectors (Jegou et al. 2012; Perronnin et al. 2010a) embedded in real-space $\mathbb{R}^D$. Other works (Milford 2013; Singh and Kosecka 2010; Sünderhauf and Protzel 2011; Sünderhauf et al. 2013) directly modeled whole-image statistics and hand-engineered *global* descriptors such as GIST (Oliva and Torralba 2001; 2006) to determine an appropriate feature representation for an image. The features extracted from each of the images are advantageously chosen to lie in a high-dimensional space that makes it especially powerful for large-scale image search. In order to keep the computational and memory complexity nominal with the growing database size, and dimensionality, the features are further efficiently indexed (Nister and Stewenius 2006), and compressed (Jegou et al. 2011) for efficient instance-level retrieval.

Sequence-based, Time-based or Context-based methods While image-level feature descriptions are convenient in identifying and tagging places, it however becomes less reliable when the dataset grows in size. This approach can become especially difficult when the appearance does not significantly change for large portions of dataset. For example, differentiating feature descriptions when driving along an empty highway can be difficult, and would lead to misidentifying and matching two considerably different locations as the same place. Furthermore, dynamic objects that may appear for a short period of time in the robot's viewing periphery. They are also undesirably encoded into the database that may potentially lead to

false positive classification. These concerns led to further work (Galvez-Lopez and Tardos 2012; Lynen et al. 2014; Maddern et al. 2012; Milford and Wyeth 2012) in matching whole sequences of consecutive images that effectively describes a place. In SeqSLAM, the authors (Milford and Wyeth 2012) identify potential loop closures by casting it as a sequence alignment problem, and solving it via dynamic programming. (Galvez-Lopez and Tardos 2012), on the other hand, rely on temporal consistency checks across long image sequences in order to robustly propose loop closures. Mei et al. (2010) finds cliques in the pose graph to define places. This representation can be powerful when objects are not fully seen in a single view and an aggregation or pooling step over the co-visibility graph enables richer scene-level descriptions. Lynen et al. (2014) proposed a "placeless" place recognition scheme where they match features on the level of individual descriptors, avoiding the need to build a vocabulary for BoVW projection. By identifying high-density regions in the distance matrix computed from feature descriptions extracted across a large sequence of images, the system can propose swaths of potentially matching places.

Learning-based methods The aforementioned methods relied heavily on hand-engineered feature extraction schemes coupled with appropriate distance metrics to match places accurately. The hyper-parameters and accompanying models were manually optimized for the target datasets they were applied to. The promise of machine learning-based methods saw widespread adoption when increasing dataset sizes suddenly rendered hand-engineered optimizations tedious. In one of the earliest works in learning-based methods, Kuipers and Beeson (2002) proposed a mechanism to identify distinctive features in a location relative to those in other nearby locations. FABMAP (Cummins and Newman 2011; 2010), one of the most relevant recent works on appearance-based mapping, models the space of visual words and their joint probability distribution. The authors in their work approximate the joint probability distribution via the Chow-Liu tree decomposition to develop an information-theoretic measure for place-recognition. The observational likelihood is probabilistically modeled by taking into account the distinctiveness of each visual word during a training phase. Through this model, one can sample from the conditional distribution of visual word occurrence, in order to appropriately weight the likelihood of having seen identical visual words before. This drastically reduces the overall rate of false positives, thereby significantly increasing the precision of the system. Latif et al. (2013) re-cast place-recognition as a sparse convex L_1 minimization problem with efficient homotopy methods that enable robust loop-closure hypothesis. In similar light, experience-based learning methods (Churchill and Newman 2012; Lowry et al. 2016) take advantage of the

robot's previous experiences to learn the set of features to match, incrementally adding more details to the description of a place if an existing description is insufficient to match a known place.

Deep Learning methods Recently, the advancements in Convolutional Neural Network Architectures (Krizhevsky et al. 2012; Simonyan and Zisserman 2014; Szegedy et al. 2014; 2016; 2017) have drastically changed the landscape of algorithms used in vision-based recognition tasks such as object recognition (Girshick 2015; Girshick et al. 2014a; Gupta et al. 2014; He et al. 2017; Redmon et al. 2016; Ren et al.; 2015) or place recognition (Zhou et al. 2014a; 2016a;b). Their adoption in vision-based place recognition for robots (Chen et al. 2017; Sunderhauf et al. 2015) has been considerably successful due to their striking transferable properties (Sharif Razavian et al. 2014). Typically, transferring the task to a new domain involves fine-tuning a pre-trained network (AlexNet (Krizhevsky et al. 2012), and VGG Net (Simonyan and Zisserman 2014)) with data gathered in the new domain. Due to the modular representation in the convolutional network stack, only the last few layers (mid-level layers, and fully connected layers) are allowed to be fine-tuned keeping the first few layers fixed (Sharif Razavian et al. 2014). More recently, Sunderhauf et al. (2015) leveraged object proposal techniques (Zitnick and Dollár 2014) combined with mid-level CNN features from Places205 (Zhou et al. 2014a) to extract feature descriptors to describe a place reliably. In order to match and identify loop-closures, however, they still resorted to a hand-specified cosine metric that enabled strong recognition performance. In recent work, Kendall et al. (2015) introduced PoseNet, where the re-localization task is cast as a regression problem with the 6-DOF pose as a target variable. The authors show some preliminary results in re-localization by utilizing Structure-from-Motion estimates to self-supervise the location regression. However, it is still unclear if such approaches generalize well to complex and larger scenes. To our knowledge, the most promising results for place recognition was proposed by Arandjelovic et al. (2016), where the authors leverage the rich representational capacity of CNN architectures coupled with a differentiable VLAD (Jégou et al. 2010; Jegou et al. 2012) layer to extract feature descriptors for large-scale instance-based retrieval. More importantly, they are able to *weakly-supervise* the training of these models with corresponding GPS information making it especially amenable for scaling up the learning task.

Current Limitations All these methods, in some way or the other, require a hand-engineered *metric* for matching the visual descriptors extracted. The choice of feature extraction needs to be tightly coupled with the right distance metric in order to retrieve similar objects appropriately. An extensive literature review in

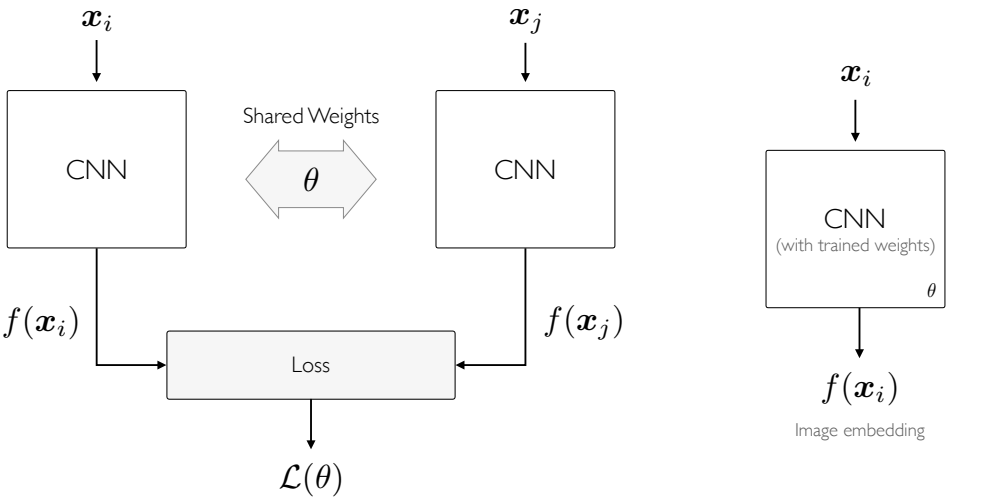
studying this problem reveals further pre-processing or post-processing steps are required to improve the overall retrieval performance. These are typically either in form of varied feature encoding techniques (Chatfield et al. 2011) such as Vector Quantization (VQ), Vector of Locally Aggregated Descriptors (VLAD), Fisher Vectors (FK), Super Vector (SV), Locality-constrained Linear encoding (LLC) etc or in the form of appropriately chosen distance metrics (Arandjelovic and Zisserman 2013; Chatfield et al. 2011; Sunderhauf et al. 2015) (L_1, L_2, Cosine, Hamming distances etc) for matching. This adds yet another level of complexity in designing and tuning reliable systems that are fault tolerant and robust to operating in varying appearance regimes. Furthermore, these approaches do not provide a mechanism to optimize for specific appearance regimes (e.g. learn to ignore fog/rain in those specific conditions). In similar light to (Chen et al. 2015b; 2017), we envision that the distance metric can be learned in an unsupervised manner (Kuipers and Beeson 2002), with the distances between features describing the same place to be well calibrated for the typical scenes that the robot may experience during its life-time. This fine-tuned behavior can be especially advantageous since the robot chooses to operate in a similar environment, and can quickly adapt to variations in the data observed.

6.3 Background

6.3.1 Metric Learning

In this work we rely on metric learning to identify a metric space where the pairs of similar sensor data (such as camera images taken of roughly the same scene, or laser point clouds of the same location) lie closer to each other, while those that are dissimilar (sensor data from different locations) are pushed away from each other in some high-dimensional space. The problem of metric learning was first introduced as *Mahalanobis metric learning* in (Xing et al. 2003), and subsequently explored with various dimensionality-reduction (Cunningham and Ghahramani 2015; Kulis 2012), information-theoretic (Davis et al. 2007) and geometric (Zadeh et al. 2016) lenses.

More abstractly, metric learning seeks to learn a non-linear mapping $f(\cdot;\theta) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that takes in input data $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^n$, where the Euclidean distance in the new target space $\|f(\boldsymbol{x}_i;\theta) - f(\boldsymbol{x}_j;\theta)\|_2$ is an approximate measure of *semantic* distance in the original space $\mathbb{R}^n$. Unlike in the supervised learning paradigm where



(a) Training: Siamese Network Architecture

(b) Testing: Feature embedding

Figure 6-2: **Training and testing architectures for Siamese Networks ►** (a) In a typical siamese network training architecture, the CNN weights (θ) are shared between the two parallel instantiated graphs, with a common loss (such as the contrastive loss) defined between them. (b) Once the CNN weights are learned, new features/images are mapped into the learned and task-appropriate embedding space $f(\boldsymbol{x}_i)$ at test time.

the loss function is evaluated for individual samples, here, we consider the loss over pairs of samples $\mathcal{X} = \mathcal{X}_S \cup \mathcal{X}_D$. We define sets of similar and dissimilar paired examples $\mathcal{X}_S$, and $\mathcal{X}_D$ respectively as follows

$$\mathcal{X}_S := \{(\boldsymbol{x}_q, \boldsymbol{x}_s) \mid \boldsymbol{x}_q \text{ and } \boldsymbol{x}_s \text{ are in the } \textit{same} \text{ class}\} \tag{6.1}$$

$$\mathcal{X}_D := \{(\boldsymbol{x}_q, \boldsymbol{x}_d) \mid \boldsymbol{x}_q \text{ and } \boldsymbol{x}_d \text{ are in } \textit{different} \text{ classes}\} \tag{6.2}$$

and define an appropriate loss function that captures the aforementioned properties.

Contrastive Loss The contrastive loss (Chopra et al. 2005) optimizes the distances between positive pairs $(\boldsymbol{x}_q, \boldsymbol{x}_s)$ such that they approach close to each other, while preserving the distances between negative pairs $(\boldsymbol{x}_q, \boldsymbol{x}_d)$ at or above a fixed margin α. Intuitively, the overall loss (Equation 6.3) is expressed as the sum of two terms.

$$\mathcal{L}(\theta) = \underbrace{\sum_{(\boldsymbol{x}_q, \boldsymbol{x}_s) \in \mathcal{X}_S} \ell_p(\boldsymbol{x}_q, \boldsymbol{x}_s)}_{\substack{\text{Penalize similar examples} \\ \text{that are far away}}} + \underbrace{\sum_{(\boldsymbol{x}_q, \boldsymbol{x}_d) \in \mathcal{X}_D} \ell_n(\boldsymbol{x}_q, \boldsymbol{x}_d)}_{\substack{\text{Penalize dissimilar examples} \\ \text{that are nearby}}} \tag{6.3}$$

$$\text{where} \quad \ell_p(\boldsymbol{x}_q, \boldsymbol{x}_s) = \|f(\boldsymbol{x}_q;\theta) - f(\boldsymbol{x}_s;\theta)\|_2^2 \tag{6.4}$$

$$\ell_n(\boldsymbol{x}_q, \boldsymbol{x}_d) = \max(0, \alpha - \|f(\boldsymbol{x}_q;\theta) - f(\boldsymbol{x}_d;\theta)\|_2)^2 \tag{6.5}$$

In the above equation (6.3), the first term penalizes positive pairs that are far away from each other, and the second term penalizes negative pairs that are nearby while ensuring a minimum margin of α between them. More generally, this reduces to the following equation with y being the indicator variable in identifying positive examples from negative ones.

$$\mathcal{L}(\theta) = \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{X}} yD(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 + (1-y)\Big[\alpha - D(\boldsymbol{x}_i, \boldsymbol{x}_j)\Big]_+^2 \tag{6.6}$$

$$\text{where} \;\; D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|f(\boldsymbol{x}_i;\theta) - f(\boldsymbol{x}_j;\theta)\|_2 \tag{6.7}$$

$$\text{and} \;\; y = \begin{cases} 1 & \text{if } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{X}_S, \\ 0 & \text{if } (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{X}_D \end{cases} \tag{6.8}$$

The margin α defines the radius around $f(\boldsymbol{x})$; the contribution to the overall loss comes from either the dissimilar pairs that are separated by a distance less than α, or from similar pairs that are separated by a large distance.

Training with Siamese Networks Learning is then typically performed with a Siamese architecture (Bromley et al. 1994; Chopra et al. 2005), consisting of two parallel networks $f(\boldsymbol{x};\theta)$ that share weights θ amongst each other (See Figure 6-2a). The contrastive loss is then defined between the two parallel networks $f(\boldsymbol{x}_i;\theta)$ and $f(\boldsymbol{x}_j;\theta)$ given by Equation 6.6. The inputs to this architecture are sets of similar $(\boldsymbol{x}_q, \boldsymbol{x}_s) \in \mathcal{X}_S$ or dissimilar samples $(\boldsymbol{x}_q, \boldsymbol{x}_d) \in \mathcal{X}_D$, with labels $y = 1$ for similar samples, and $y = 0$ otherwise. The scalar output loss computed from batches of similar and dissimilar samples are then used to update the parameters of the siamese network θ via Stochastic Gradient Descent (SGD). Typically, batches of positive and negative samples are provided in alternating fashion during training.

Learned Feature Embedding Once the parameters θ of the Siamese network are sufficiently learned for the desired task, we strip the parallel network architecture and only consider one of the networks for embedding the input feature $\boldsymbol{x}_i$ in its task-appropriate feature space (See Figure 6-2b). The distances (L_2) in this new embedding space are considered to be more appropriate for the task, and is especially amenable to high-dimensional indexing and querying for image retrieval

purposes. For a comprehensive overview of metric learning and its various forms, we refer the reader to (Kulis et al. 2013).

6.4 Self-Supervised Metric Learning for Place Recognition

With the growing experiences that robots log today, we recognize the need for *fully automatic solutions* for learning of and improving their performance in tasks such as place recognition or loop-closure identification. Inspired by NetVLAD (Arandjelovic et al. 2016), we cast place recognition in robots as a *self-supervised* metric learning problem. However, unlike their work, we instead focus on learning the distance function that is optimal for the desired task. Most previous works (Lowry et al. 2016; Milford 2013; Sunderhauf et al. 2015) use hand-engineered image representations such as SIFT/ORB followed by a pooling to describe the whole image as a single feature vector. However, in their work, they assume that the feature descriptions are well-separated in the Euclidean space. Certain other works (Sunderhauf et al. 2015), explicitly model the distance metric based on the choices of image representation embedding. In this work, however, we realize that certain choices in explicit image feature representation may be application-specific to admit certain representational capacity (Lowry et al. 2016; Sunderhauf et al. 2015).

Convolutional Neural Networks (CNNs) have been shown to be remarkably powerful in various tasks including instance retrieval, image and object classification, and saliency detection. Part of their success has been attributed to their the enormous representational capacity that enable these methods to extract rich, high-level and descriptive semantic features from images. However, most of these systems require large amounts of training data in order to achieve their highly touted performance capabilities. Moreover, most domain-specific tasks require further fine-tuning of these large-scale networks in order to perform well in their application-specific tasks. Despite the ready availability of training models and weights, we foresee the data collection and its supervision being a predominant source of friction for fine-tuning models for application-specific tasks such as place recognition. Due to the rich amount of cross-modal information that robots typically collect, we envision that robots to be able to self-supervise tasks such as place recognition by fine-tuning existing CNN models on the experience they have accumulated.

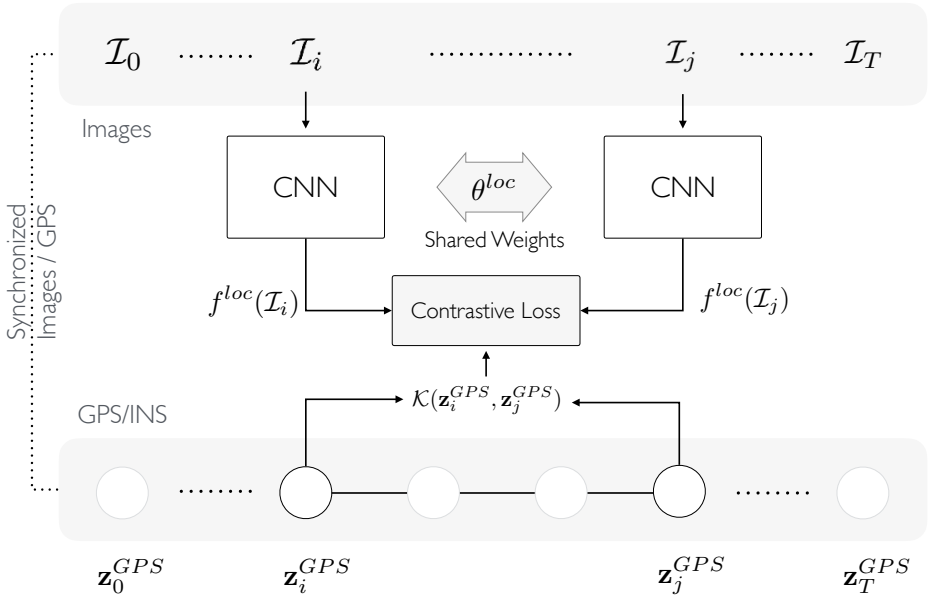


Figure 6-3: **Self-Supervised Metric Learning for Localization ►** The illustration of our proposed self-supervised Siamese Net architecture. The model bootstraps synchronized cross-modal information (Images and GPS) in order to learn an appropriate similarity metric between pairs of images in an embedded space, that implicitly learns to predict the loop-closure detection task. The key idea is the ability to sample and train our model on positive and negative pairs of examples of similar and dissimilar places by taking advantage of corresponding GPS location information.

To this end, we bolster these CNN-based feature representations with a fully trainable and optimized distance metric that significantly improves the precision-recall performance. Furthermore, we propose a completely *self-supervised* approach to learning the distance metric thereby avoiding the need for any external supervisory signal besides the data collected from the robot's experience. We refer the reader to Algorithm 6 for a high-level description of our proposed self-supervised metric learning procedure for place-recognition.

6.4.1 Self-supervised Dataset Generation

Multi-camera systems and navigation modules have more-or-less become ubiquitous in modern autonomous systems today. Typical systems log this sensory information in an asynchronous manner, providing a treasure of cross-modal information that can be readily used for machine learning purposes. We foresee robots in the near future to be able to teach itself representations for newly introduced sensors by *bootstrapping* these task capabilities from other sensory channels. Here, we focus on the task of vision-based place recognition via a forward-looking camera, by leveraging synchronized information collected via standard navigation modules

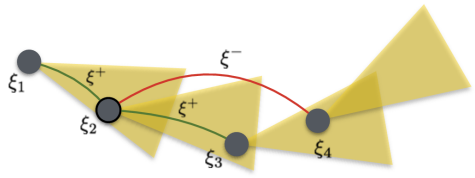


Figure 6-4: **Self-Supervision from camera frustum overlap ►** The camera frustum overlap in subsequent views allows us to confidently sample positive and negative examples of visual loop-closures directly from the keyframe similarity described in Equation 6.9.

(GPS/IMU, Visual-Inertial etc.).

Sensor Synchronization In order to formalize the notation used in the following sections, we shall refer to $(\mathcal{I}_t, \mathbf{z}_t^{GPS})$ as the *synchronized* tuple of camera image $\mathcal{I}$, and GPS measurement $\mathbf{z}^{GPS}$, captured at approximately the same time t. In typical systems however, these sensor measurements are captured in an asynchronous manner, and the synchronization needs to be carried out carefully in order to ensure clean and reliable measurements for the bootstrapping procedure. It is important to note that for the specific task of place recognition, $\mathbf{z}$ can also be sourced from various external sensors including, but not limited to, standard INS, or even LiDAR-based localization systems that these autonomous systems are already equipped with.

Key-frame Sub-sampling While we could consider the full set of synchronized image-GPS pairs, it can be especially efficient to extract and learn only from a diverse set of viewpoints. We expect that learning from this strictly smaller, yet sufficiently diverse set, can substantially speed up the training process while being able to achieve the same performance characteristics when trained with the original dataset. While it is unclear what this sampling function may look like for image descriptions, we can easily provide this measure to determine a diverse set of GPS measurements. We incorporate this via a standard keyframe-selection strategy (Klein and Murray 2007; Strasdat et al. 2010) where the poses are sampled from a continuous stream whenever the relative pose has exceeded a certain translational or rotational threshold from its previously established keyframe. We set these translational and rotational thresholds to 5m, and $\frac{\pi}{6}$ radians respectively to allow for efficient sampling of diverse keyframes. For an illustration of the sampling based on GPS or viewing frustum, we refer the reader to Figure 6-4.

Key-frame Similarity The self-supervision is enabled by defining a viewing frustum that applies to both the navigation-view $\mathbf{z}_t$ and the image-view. We define

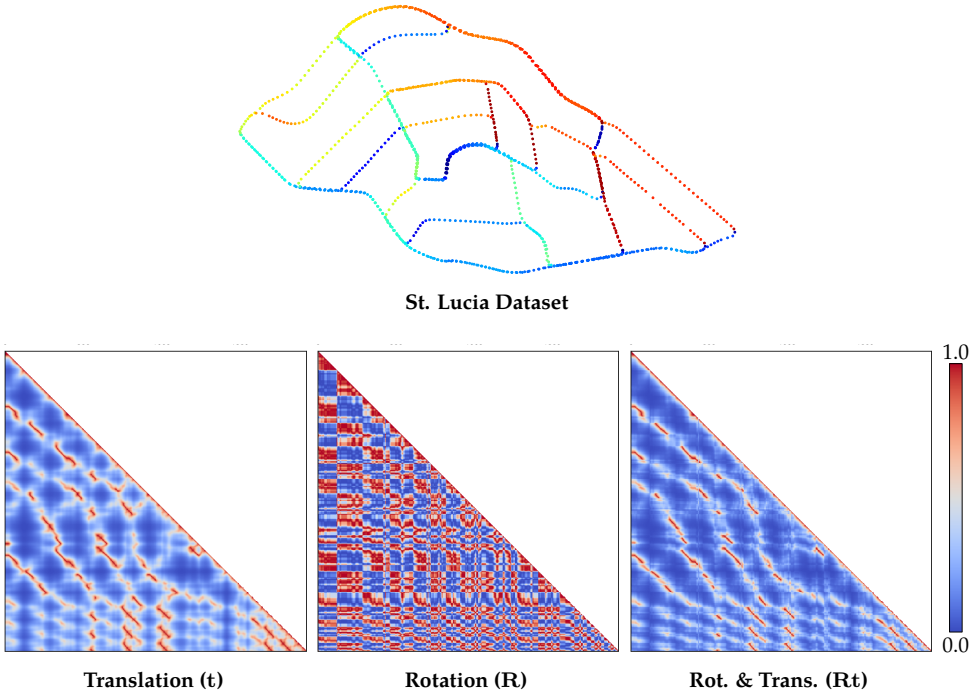


Figure 6-5: **Bootstrapped learning using cross-modal information** ▶ *Top row* ▶ An illustration of the vehicle path traversed in the St. Lucia dataset (`100909_1210`) with synchronized Image and GPS measurements. The colors correspond to the vehicle bearing angle (Rotation $\mathbf{R}$) inferred from the sequential GPS measurements. *Bottom row* ▶ The self-similarity matrix determined from the translation ($\mathbf{t}$), rotation ($\mathbf{R}$) and their combination ($\mathbf{Rt}$) on the St. Lucia Dataset using the assumed ground-truth GPS measurements. Each row and column in the self-similarity matrix corresponds to key-frames sampled from the dataset as described in Section 6.4.1. The sampling scheme ensures a time-invariant (aligned) representation where loop-closures appear as off-diagonal entries that are a fixed-offset from the current sequence (main-diagonal). We use a Gaussian kernel (Equation 6.9) to describe the similarity between key-frames and sample positive/negative samples from the combined $\mathbf{Rt}$ similarity matrix.

a Gaussian similarity kernel $\mathcal{K}$ between two instances of GPS measurements $\mathbf{z}_i^{GPS}$ and $\mathbf{z}_j^{GPS}$ given by:

$$\mathcal{K}(\mathbf{z}_i^{GPS}, \mathbf{z}_j^{GPS}) = \underbrace{\exp(-\gamma^{\mathbf{t}} \left\| \mathbf{z}_i^{\mathbf{t}} - \mathbf{z}_j^{\mathbf{t}} \right\|_2^2)}_{\text{Translation similarity}} \cdot \underbrace{\exp(-\gamma^{\mathbf{R}} \left\| \mathbf{z}_i^{\mathbf{R}} \ominus \mathbf{z}_j^{\mathbf{R}} \right\|_2^2)}_{\text{Rotation similarity}} \tag{6.9}$$

where $\mathbf{z}_i^{\mathbf{t}}$ is the GPS translation measured in metric-coordinates at time i, and $\mathbf{z}_i^{\mathbf{R}}$ is the corresponding rotation or bearing determined from the sequential GPS coordinates for the particular session (See Figure 6-5). Here, the only hyper-parameter required is the choice of the bandwidth parameters $\gamma^{\mathbf{R}}$ and $\gamma^{\mathbf{t}}$, and generally depends on the viewing frustum of the camera used. The resulting similarity matrix for the translation (using GPS translation $\mathbf{t}$ only), and the rotation (using established bearing $\mathbf{R}$ only) is illustrated in Figure 6-5. Sampling is illustrated in Figure 6-6.

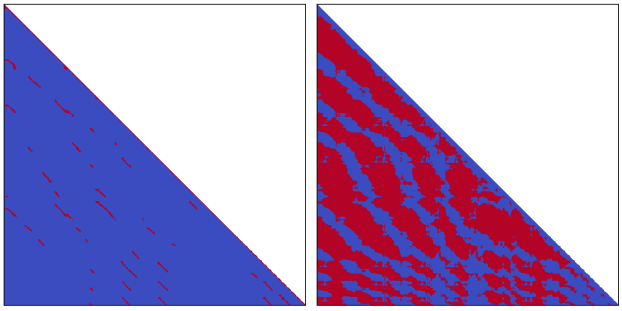


Figure 6-6: **Self-Supervised sampling ►** The $\mathcal{K}$ kernel computed in equation 6.9 is used to "supervise" the sampling procedure. **Left figure:** Samples whose kernel $\mathcal{K}(\mathbf{z}^{GPS}, \mathbf{z}'^{GPS})$ evaluates to higher than $\tau_p^{\mathbf{Rt}}$ are considered as positive samples (in red). **Right figure:** Samples whose kernel $\mathcal{K}(\mathbf{z}^{GPS}, \mathbf{z}'^{GPS})$ evaluates to lower than $\tau_n^{\mathbf{Rt}}$ are consider as negative examples (in red).

Distance-Weighted Sampling With key-frame based sampling considerably reducing the dataset to a diverse, yet representative one for efficient training, we now focus on sampling positive and negative pairs in order to ensure speedy convergence of the proposed contrastive loss function. We first consider the key-frame similarity matrix between all pairs of keyframes for a given dataset, and sample positive pairs whose similarity exceeds a specified threshold $\tau_p^{\mathbf{Rt}}$. Similarly, we sample negative pairs whose similarity is below $\tau_n^{\mathbf{Rt}}$. For each of the positive and negative sets, we further sample uniformly by their inverse distance following (Wu et al. 2017) closely, to encourage faster convergence.

6.4.2 Learning an Appropriate Distance Metric for Localization

Our proposed self-supervised place recognition architecture is realized with a Siamese network (Figure 6-2) with an appropriate contrastive loss (given by Equation 6.10), that simultaneously finds a reduced dimensional metric space where the relative distances between features in the embedded space are *well-calibrated*. In this context, well-calibrated refers to the property that negative samples are separated at least by a known margin α, while positive samples are likely to be separated by a distance less than the margin. Following the terminology in Section 6.3.1, we consider tuples $(\mathcal{I}_i, \mathbf{z}_i^{GPS}) \in \mathcal{X}$ of similar (positive) $\mathcal{X}_S \subset \mathcal{X}$ and dissimilar (negative) examples $\mathcal{X}_D \subset \mathcal{X}$ for learning an appropriate embedding function $f^{loc}(\cdot; \theta^{loc})$. Intuitively, we seek to find a "*semantic measure*" of distance given by $D(\mathcal{I}_i, \mathcal{I}_j) = \left\| f^{loc}(\mathcal{I}_i; \theta^{loc}), f^{loc}(\mathcal{I}_j; \theta^{loc}) \right\|_2$ in a target space of $\mathbb{R}^m$ such that they are "*similar*" to those defined in the metric space of GPS measurements (in this case) given by $D(\mathbf{z}_i^{GPS}, \mathbf{z}_j^{GPS}) = \left\| \mathbf{z}_i^{GPS} - \mathbf{z}_j^{GPS} \right\|_2$. Since we are particularly interested in

identifying potential images that may be taken from within a fixed distance of each other, we make the appropriate modification to the objective such that the original task of loop-closure recognition can be performed with a probabilistic interpretation.

Let $(\mathcal{I}, \mathbf{z}^{GPS}) \in \mathcal{X}$ be the input data and $\mathbb{1}_{GPS} \in 0, 1$ be the indicator variable representing dissimilar ($\mathbb{1}_{GPS} = 0$) and similar ($\mathbb{1}_{GPS} = 1$) pairs of examples within $\mathcal{X}$. We seek to find a kernel $f^{loc}(\cdot; \theta^{loc}) : \mathcal{I} \mapsto \Phi$ that maps the input image $\mathcal{I}$ to an embedding $\Phi \in \mathbb{R}^m$ whose *distances between similar places are low*, while the *distances between dissimilar places are high*. We take advantage of availability of synchronized Image-GPS measurements $(\mathcal{I}, \mathbf{z}^{GPS})$ to provide an indicator for place similarity, thereby rendering this procedure fully automatic and *self-supervised*. Re-writing equation 6.6 for our problem, we get Equation 6.10 where $D(\mathcal{I}_i, \mathcal{I}_j)$ measures the "*semantic distance*" between images (Equation 6.11). The indicator variable $\mathbb{1}_{GPS}$ in Equation 6.12 determines whether the sample belongs to the similar $\mathcal{X}_S$ ($\mathbb{1}_{GPS} = 1$, if so) or the dissimilar set $\mathcal{X}_D$ ($\mathbb{1}_{GPS} = 0$).

$$\mathcal{L}(\theta^{loc}) = \sum_{((\mathcal{I}_i, \mathbf{z}_i), (\mathcal{I}_j, \mathbf{z}_j)) \in \mathcal{X}} (\mathbb{1}_{GPS}) \cdot D(\mathcal{I}_i, \mathcal{I}_j)^2 + (1 - \mathbb{1}_{GPS}) \cdot \Big[\alpha - D(\mathcal{I}_i, \mathcal{I}_j)\Big]_+^2 \tag{6.10}$$

$$\text{where} \;\; D(\mathcal{I}_i, \mathcal{I}_j) = \left\| f^{loc}(\mathcal{I}_i; \theta^{loc}) - f^{loc}(\mathcal{I}_j; \theta^{loc}) \right\|_2 \tag{6.11}$$

$$\text{and} \;\; \mathbb{1}_{GPS} = \begin{cases} 1 & \text{if } \mathcal{K}(\mathbf{z}_i^{GPS}, \mathbf{z}_j^{GPS}) > \tau_p^{\mathbf{Rt}} \\ 0 & \text{if } \mathcal{K}(\mathbf{z}_i^{GPS}, \mathbf{z}_j^{GPS}) < \tau_n^{\mathbf{Rt}} \end{cases} \tag{6.12}$$

For brevity, we omit θ^{loc} and use $f^{loc}(\mathcal{I}_i)$ instead of the full expression $f^{loc}(\mathcal{I}_i; \theta^{loc})$. We pick the thresholds for $\tau^{\mathbf{Rt}}$ based on a combination of factors including convergence rate and overall accuracy of the final learned metric. Nominal values of $\tau_p^{\mathbf{Rt}}$ range from 0.8 to 0.9 that indicate the tightness of the overlap between viewing frustums of positive examples, with $\tau_n^{\mathbf{Rt}}$ for negative examples set to 0.4.

Figure 6-7 illustrates the visual self-similarity matrix of the feature embedding at various stages during the training process. Initially (at epoch 0), the feature embedding is equivalent to the original feature description, where the distances are not well-calibrated. As training progresses, the similarity metric learning draws positively labeled examples of loop-closure image pairs closer together in the embedded space, while pushing the negative examples farther from each other. As the training converges, we start to notice a few characteristics in the learned embedding that make it especially powerful in identifying loop-closures: (i) The red diagonal bands in the visual self-similarity matrix are well-separated from the blue

Algorithm 6 Self-Supervised Metric Learning for Place Recognition

Input: $\mathcal{X} = \{(\mathcal{I}_1, \mathbf{z}_1^{GPS}), \ldots, (\mathcal{I}_t, \mathbf{z}_t^{GPS})\}$: Image sequence and corresponding GPS measurements
Output: f^{loc}, θ^{loc}: Improved feature embedding model for place recognition task

▷ Compute Key-frame similarity matrix (Equation 6.9)
1: $\mathcal{K}^{GPS} \leftarrow$ PairwiseKeyframeSimilarity($\mathbf{z}^{GPS}$)

▷ Generate positive and negative examples for learning the similarity metric (Section 6.4.1)
2: $\mathcal{X}_S, \mathcal{X}_D \leftarrow$ SelfSupervisedDatasetGeneration($\mathcal{X}, \mathcal{K}^{GPS}$)

▷ Discriminative Similarity Metric Learning via Contrastive Loss (Section 6.4.2)
▷ Learned invariant mapping $f^{loc} : \mathcal{I} \mapsto \Phi$, where $\mathcal{I} \in \mathbb{R}^n$ and $\Phi \in \mathbb{R}^m$
3: $f^{loc}, \theta^{loc} \leftarrow$ SceneSimilarityMetricLearning($\mathcal{X}_S, \mathcal{X}_D$)

background indicating that the learned embedding has identified a more separable function for the purposes of loop-closure recognition; and (ii) The visual self-similarity matrix starts to resemble the target self-similarity matrix computed using the GPS measurements (as shown in Figure 6-8). Furthermore, the t-SNE embedding[1] (whose values are visualized in the RGB colorspace) of the learned features extracted at identical locations are strikingly similar, indicating that the learned feature embedding $f(\cdot; \theta^{loc})$ has implicitly learned a metric space that is better suited for the task of place-recognition in mobile robots (See Figure 6-8).

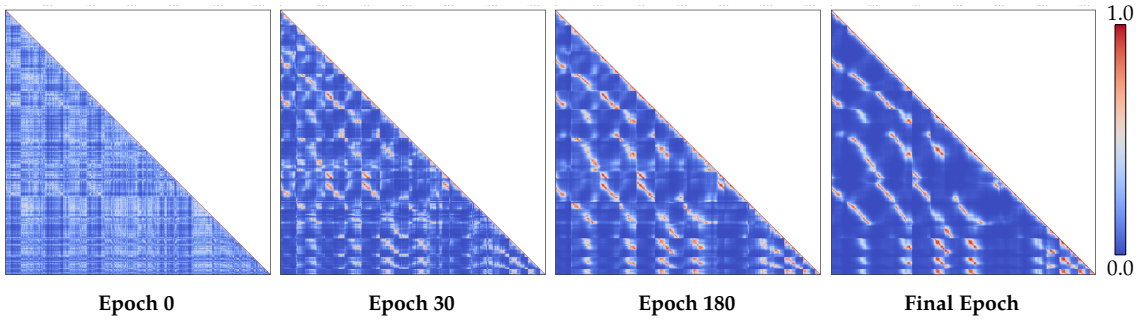


Figure 6-7: **Self-Supervised learning of a visual-similarity metric ►** An illustration of the similarity matrix at various stages of training. At *Epoch 0*, the distances between features extracted at identical locations are not well-calibrated requiring hand-tuned metrics for reliable matching. With more positive and negative training examples, the model at *Epoch 30* has learned to draw positive features closer together (strong red off-diagonal sequences indicating loop-closures), while pushing negative features farther apart (strong blue background). This trend continues with *Epoch 180* where the loop-closures start to look well-defined, while the background is consistently blue indicating a reduced likelihood for false-positives.

[1] t-SNE (Maaten and Hinton 2008) is a non-linear dimensionality reduction technique that is especially tailored to embedding high-dimensional data on a lower dimensional manifold, typically in $\mathbb{R}^2$ or $\mathbb{R}^3$. This makes it particularly valuable in visualizing high-dimensional data. In our case, we embed the high-dimensional features onto a 3-dimensional manifold via t-SNE and visualize the data as if they sit in a 3-dimensional RGB-colorspace. This allows us to identify similar feature embeddings by their color, where features with similar color indicate that they lie closer to each other in the original higher-dimensional space.

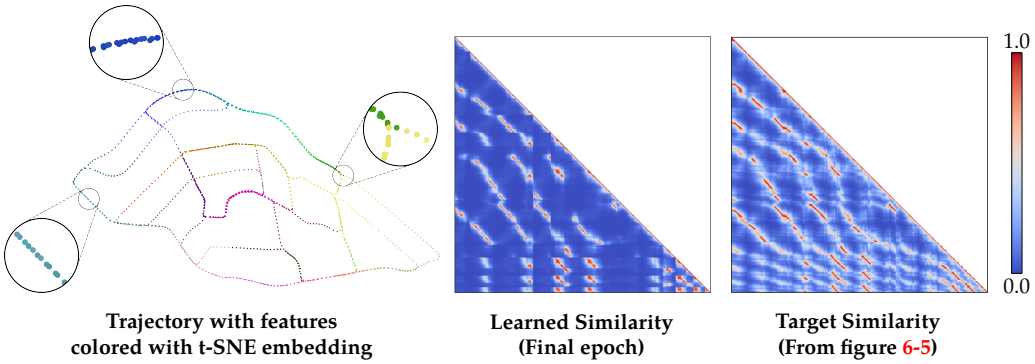


Figure 6-8: **Qualitative results of self-supervised metric learning on the St. Lucia Dataset** ▶ *Left*: An illustration of the path traversed (`100909_1210`) with the colors indicating the 3-D t-SNE embedding of the learned features Φ extracted at those corresponding locations. The visual features extracted across multiple traversals along the same location are consistent, as indicated by their similar color embedding. The colors are plotted in the RGB colorspace. *Columns 2 and 3*: Comparison of the learned visual-similarity metric against the target or ground truth similarity metric (obtained by determining overlapping frustums using GPS measurements). As expected, the distances in the learned model tend to be *well-calibrated* enabling strong precision-recall performance. Furthermore, the model can be qualitatively validated when the learned similarity matrix starts to closely resemble the target similarity matrix (comparing columns 2 and 3 in the figure).

6.4.3 Efficient Scene Indexing, Retrieval and Matching

One of the critical requirements for place-recognition is to ensure high recall in loop-closure proposals while maintaining sufficiently high precision in candidate matches. This however requires probabilistic interpretability of the matches proposed, with accurate measures of confidence in order to incorporate these measurements into the back-end pose graph optimization. Typically, similarities or distances measured in the image descriptor space are not well-calibrated, making these measures only amenable to distance-agnostic matching such as k-nearest neighbor search. Moreover, an indexing and matching scheme such as k-nn also makes it difficult to filter out false positives as the distances between descriptors in the original embedding space is practically meaningless. Calibrating distances by learning a new embedding has the added advantage of avoiding these false positives, while being able to recover confidence measures for the matches retrieved.

Once features Φ are extracted and mapped to an appropriate target space in $\mathbb{R}^m$, we require a mechanism to insert and query these embedded descriptors from a database. We use a KD-Tree in order to incrementally insert features into a balanced tree data structure, thereby enabling $\mathcal{O}(\log N)$ queries. While other works resort to efficient encodings such as Product Quantization (PQ) (Jegou et al. 2011) to speed up querying, our model learns already a reduced dimensionality target space that is especially conducive for efficient indexing and querying.

6.5 Towards Self-Supervised Visual-SLAM Front-Ends

Modern SLAM systems today (Engel et al. 2014; Mur-Artal et al. 2015) typically consist of a vision-based front-end component to construct the set of constraints for optimization performed by a back-end factor-graph based solver (Kaess et al. 2008; Kümmerle et al. 2011). The vision-based front-end typically consists of a visual-odometry module that is responsible for frame-to-frame camera ego-motion tracking, and a vision-based place-recognition module that identifies potential loop-closure constraints that may be added to the overall graph-based optimization. We envision robots in the near future to be able to bootstrap, and learn to perform both these critical tasks as they collect more relevant experience.

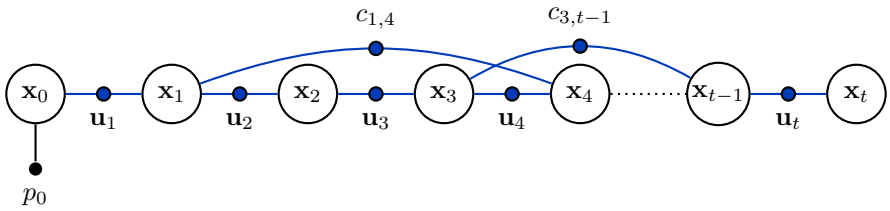


Figure 6-9: **Self-Supervised Vision-based Front-End ►** With the methods developed thus far, we propose one of the very first *self-supervised* visual-SLAM front-ends. Again, in the typical factor-graph formulation, the visual ego-motion estimator provides odometry measurement factors ($u_{i-1,i}$), while the vision-based loop-closure module provides relative-pose constraints $c_{j,k}$ between temporally distant nodes.

Self-supervised Visual Ego-motion Following work from Chapter 5, we learn the implicit visual ego-motion function in autonomous systems that are equipped with standard sensors such as imaging and navigation modules. By leveraging GPS-aided SLAM as a bootstrapping mechanism, we learn to perform frame-to-frame odometry from sequential camera imagery.

Self-supervised Visual Loop-Closure Identification As described earlier, we leverage image-GPS pairs typically collected in autonomous vehicles to learn a metric for accurate loop-closure identification. By learning a new embedding for the task of loop-closure recognition, we are able to develop probabilistic and interpretable systems that are more reliable and tolerant to hyperparameter tuning.

We refer the reader to Algorithm 7 for a high-level overview of how the self-supervised visual-SLAM front-end component is realized. $f^{vo}(\cdot;\theta^{vo})$ and $f^{loc}(\cdot;\theta^{loc})$ refer to the model and parameters learned for each of the Visual-SLAM front-end tasks, namely visual-odometry (*vo*) and vision-based localization (*loc*).

Algorithm 7 Deployment of Learned Visual-SLAM Front-End (See Section 6.5)

Inputs: Input image sequence ($\mathcal{I}$)
Learned Visual-Egomotion Model and Parameters (f^{vo}, θ^{vo})
Learned Visual-Place Similarity Model (f^{loc}, θ^{loc})
Outputs: Optimized robot trajectory ($\hat{\mathbf{x}}_{1:t}$)

▷ Ego-motion estimation (Section 5.4)
$\mathbf{u}_t \leftarrow f^{vo}(\mathcal{I}_{t-1}, \mathcal{I}_t; \theta^{vo})$
▷ Odometry factor insertion
$\mathcal{G} \leftarrow \{\mathcal{G}, \mathbf{u}_t\}$
▷ Extract visual place description (Section 6.4)
$\Phi_t \leftarrow f^{loc}(\mathcal{I}_t; \theta^{loc})$
▷ Identify loop-closure
▷ (query index: q, prob: p_q)
$q, p_q \leftarrow$ QueryClosestSceneDescription(Φ_t)
▷ Loop-closure factor insertion
▷ $c_{q,t}$: Zero-translation relative-pose constraint with large Σ
if $p_q > 0.9$ **then**
    $\mathcal{G} \leftarrow \{\mathcal{G}, c_{q,t}\}$
end if
▷ Pose-graph optimization
$\hat{\mathbf{x}}_{1:t} \leftarrow$ PoseGraphIncrementalUpdate($\mathcal{G}, \hat{\mathbf{x}}_{1:t-1}$)

6.6 Experiments and Results

We evaluate the performance of the proposed self-supervised localization method on the KITTI (Geiger et al. 2012) and St. Lucia Dataset (Glover et al. 2010). For each of the datasets, we train the localization component on all sessions but the test set. We compare our approach against the image descriptions obtained from extracting the activations from several layers in the Places365-AlexNet pre-trained model (Zhou et al. 2016b) (*conv3*, *conv4*, *conv5*, *pool5*, *fc6*, *fc7* and *fc8* layers). While we take advantage of the pre-trained models developed in (Zhou et al. 2016b) for the following experiments, we remind the reader that the proposed framework could allow us to learn relevant task-specific embeddings from any image-based feature descriptor. The implementation details of our proposed method is described in detail in section 6.6.4.

6.6.1 Learned Feature Embedding Characteristics

While pre-trained models can be especially powerful image descriptors, they are typically trained on publicly-available datasets such as the ImageNet (Russakovsky

et al. 2015), PASCAL VOC (Everingham et al.), COCO (Chen et al. 2015a) etc. that have strikingly different natural image statistics. Moreover, some of these models are trained for the purpose of image or place classification that specializes their feature description capabilities to the desired task. As with most pre-trained models, we expect some of the descriptive performance of Convolutional Neural Networks to generalize, especially in its lower-level layers (*conv1*, *conv2*, *conv3*). However, the descriptive capabilities in its mid-level and higher-level layers (*pool4*, *pool5*, *fc* layers) start to specialize to the specific data regime and recognition task it is trained for. This has been addressed quite extensively in the literature, arguing the need for domain adaptation and fine-tuning these models on more representative datasets to improve task-specific performance (Ganin and Lempitsky 2015; Gopalan et al. 2011; Khosla et al. 2012; Oquab et al. 2014).

Similar to previous domain adaptation works (Ganin and Lempitsky 2015; Glorot et al. 2011; Gopalan et al. 2011), we are interested in adapting existing models to newer task domains such as place-recognition with minimal human supervision involved. We argue for a self-supervised approach to model fine-tuning, and emphasize the need for a well-calibrated embedding space, where the features embedded in the new space can provide measures for both similarity and the corresponding confidence associated in matching.

Comparing performance between the original and learned embedding space In Figure 6-10, we compare the precision-recall performance in loop-closure recognition using the original and learned feature embedding space. For various thresholds of localization accuracy (20 and 30 meters), our learned embedding shows a considerable performance boost over the pre-trained Places365-AlexNet model. In the figures, we also illustrate the noticeable drop in performance with the descriptive capabilities in the higher-level layers (*fc6*, *fc7*, *fc8*) as compared to the lower-level layers (*conv3*, *conv4*, *conv5*) in the Places365-AlexNet model. This is as expected, since the higher layers in the CNN (*pool5*, *fc6*, *fc7*) are more tailored to the original classification task they were trained for.

Embedding distance calibration As described earlier, our approach to learning an appropriate similarity metric for visual loop-closure recognition affords a probabilistic interpretation of the matches proposed. These accurate measures of confidence can be later used to incorporate these measurements into the back-end pose graph optimization. Figure 6-11 illustrates the interpretability of the proposed learned embedding metric compared to the original feature embedding distance metric. The histograms for the L_2 embedding distance separation are illustrated for

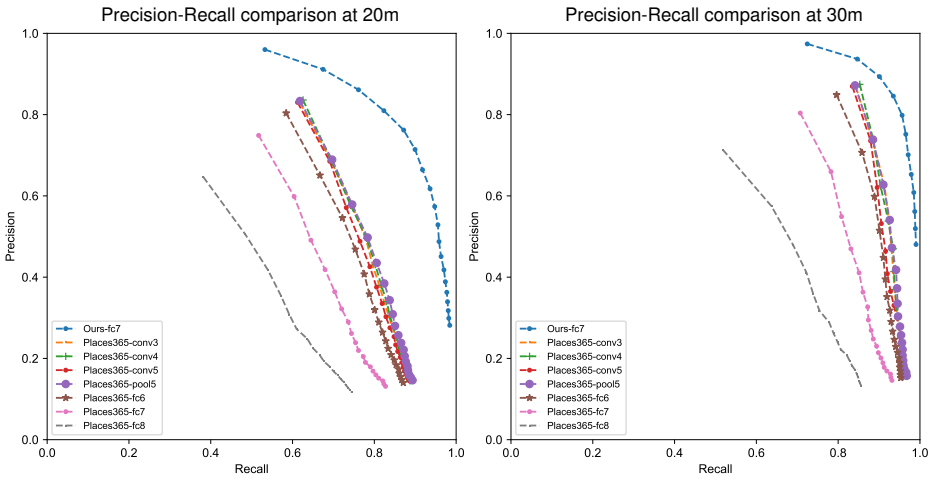


Figure 6-10: **Precision-Recall performance in loop-closure recognition using the original and learned feature embedding space ►** The figures show the precision-recall (P-R) performance in loop-closure recognition for various feature descriptors using the pre-trained Places365-AlexNet model and the learned embedding (*Ours-fc7*). Our learned embedding is able to significantly outperform the pre-trained Places365-AlexNet model for all feature layers, by *self-supervising* the model on a more representative dataset.

both positive (in green) and negative (in blue) pairs of features. Here, a positive pair refers to feature descriptions of images taken at identical locations, while the negative pairs refer to those pairs of feature descriptions that were taken from at least 50 meters apart from each other. The figure clearly illustrates how the learned embedding (*Ours-fc7*) is able to tease apart positive pairs, from those between the negative pairs of features, enabling an improved classifier (with a more obvious separator) for place-recognition. Intuitively, the histogram overlap between the positive and negative probability masses measures the ambiguity in loop-closure identification, with our learned feature embedding (*Ours-fc7*) demonstrating the least amount of overlap.

ε-NN search in the learned feature embedding space Once the distances are calibrated in the feature embedding space, even a naïve fixed-radius nearest neighbor strategy, that we shall refer to as ε-NN, can be surprisingly powerful. In Figure 6-12, we show that our approach is able to achieve high-recall, with considerably strong precision performance for features that lie within distance α (contrastive loss margin as described in Section 6.4.2) from each other.

Furthermore, the feature embedding can also be used in the context of image retrieval with strong recall performance via naïve k-Nearest Neighbor (k-NN) search. Figure 6-13 compares the precision-recall performance of the k-NN strategy on the original and learned embedding space, and shows a considerable performance gain in the learned embedding space. Furthermore, the recall performance also tends to

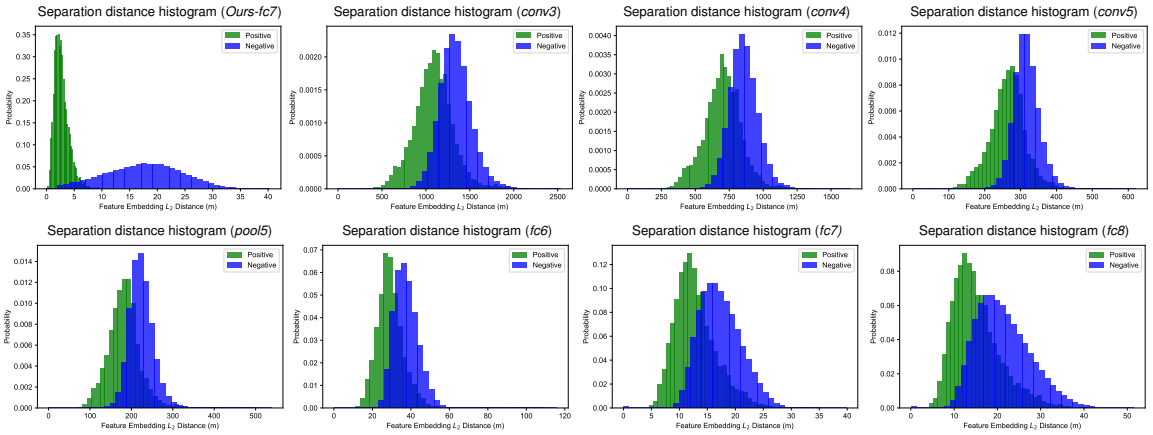


Figure 6-11: **Separation distance calibration ►** The histograms of L_2 distances between positive and negative examples are shown for the various feature descriptions with the pre-trained Places365-AlexNet model. Our learned model is able to fine-tune intermediate layers and distort the feature embedding such that the distances between positive and negative examples (similar and dissimilar places) are well-calibrated. This is seen especially in the first plot (top row, far left *Ours-fc7*), where the probability mass for positive and negative examples are better separated with reduced overlap, while the other histograms are not well-separated in the feature embedding space.

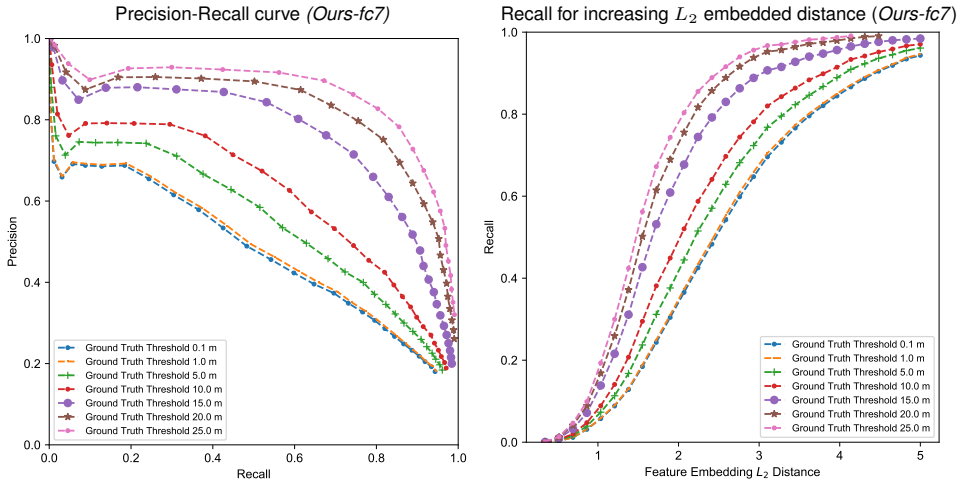


Figure 6-12: **Precision-Recall performance for loop-closure recognition in the original and learned feature embedding space using fixed-radius neighborhood search (ε-nn) ►** The first column convincingly shows that our learned feature embedding space is able to maintain strong Precision-Recall performance by using ε-nn (fixed-radius search). The plot on the second column shows the recall performance with increasing feature embedding L_2 distance considered for each query sample. The Siamese network was trained with a contrastive loss margin of $\alpha = 10$, which distorts the embedding space such that positive pairs are encouraged to only be separated by an L_2 distance of 10 or lower. The figure on the *right* shows that in the learned feature embedding space (*Ours-fc7*), we are able to capture most candidate loop-closures within an L_2 distance of 5 from the query sample, as more matching neighbors are considered.

be higher for the learned embedding space as compared to the original descriptor space.

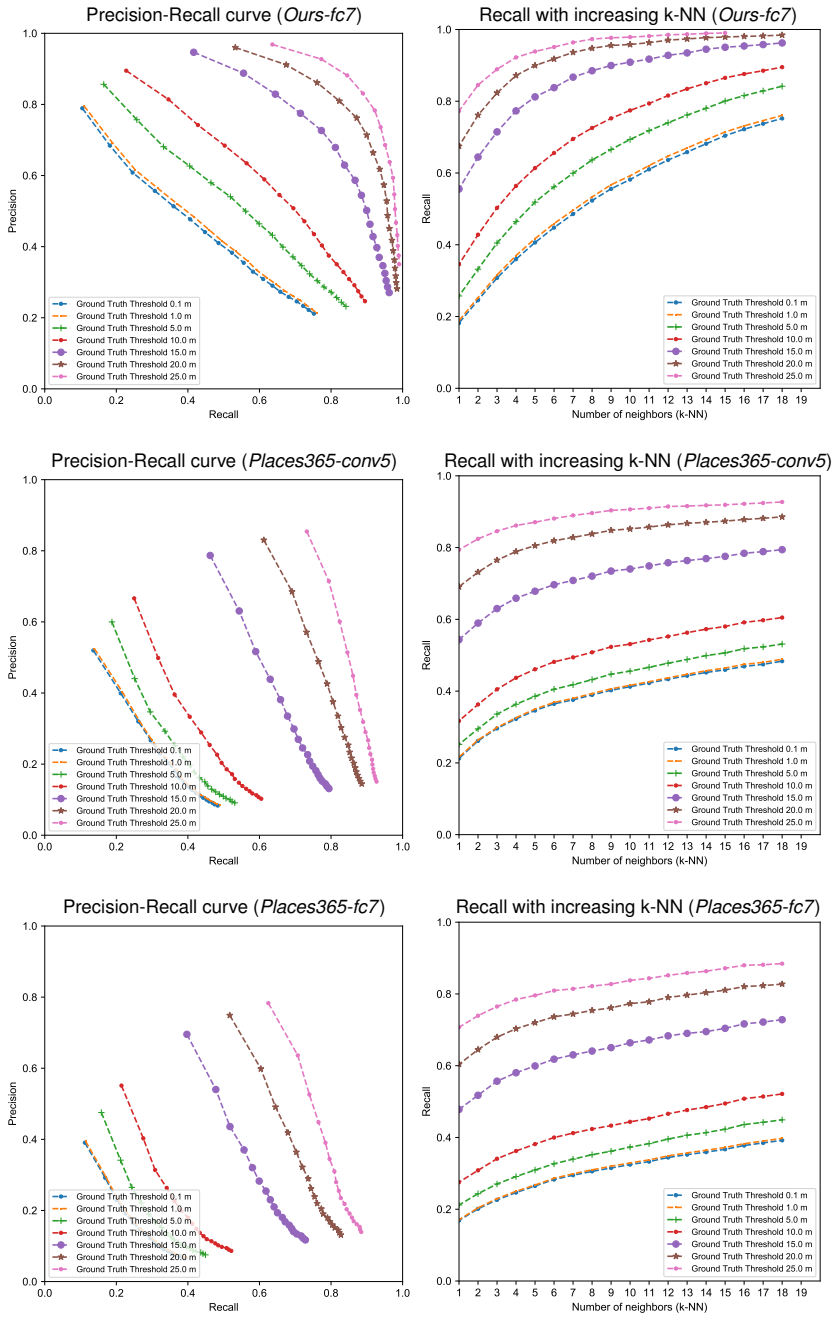


Figure 6-13: **Precision-Recall performance for loop-closure recognition in the original and learned feature embedding space using k-Nearest Neighbors ►** The first column shows that our learned feature embedding space is able to capture more Precision-Recall performance than the pre-trained layers (Places365-AlexNet *conv5* and *fc7*). The plot on the second column shows the recall performance with increasing set of neighbors considered for each query sample. Using the learned feature embedding space (Ours-fc7), we are able to capture more candidate loop-closures within the closest 20 neighbors of the query sample.

6.6.2 Qualitative Results on Loop Closure Recognition

As previously described, we rely on metric learning to determine an appropriate image embedding, whose distances are well-calibrated to the place-recognition task. More specifically, we are interested in minimizing human intervention even in the training phase, and only leverage the data collected from the more representative dataset to learn the feature embedding. Qualitatively, one can assess the representational capacity of the learned feature embedding space over the original feature description space by simply performing a naïve fixed-radius neighborhood search (ε-nn) to identify images that are potentially captured from identical locations. Figure 6-14 compares the qualitative localization performance using ε-nn search between a pre-trained Places365-AlexNet descriptor via its *fc7* layer (*Places365-fc7*) and our learned feature embedding (*Ours-fc7*).

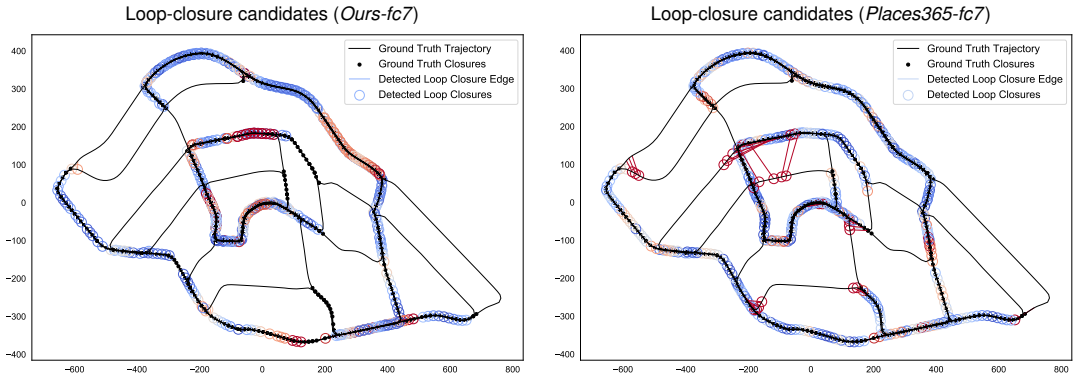


Figure 6-14: **Qualitative comparison of loop-closure identification in the original and learned feature embedding space ►** The figures show the qualitative performance of our proposed localization method using the learned embedding (*Ours-fc7*), compared to the pre-trained Places365-AlexNet model (*Places365-fc7*). Despite using a naive nearest-neighbor strategy, our method is able to minimize false positives (crossed-edges shown in red).

6.6.3 Localization Performance within Visual-SLAM Front-Ends

Recall in Figure 6-9, we illustrate the underlying factor graph instantiated to recover the optimized vehicle trajectories. Figure 6-15 shows the trajectory of the optimized pose-graph leveraging the constraints proposed by our learned loop-closure proposal method. The visual place-recognition module determines constraints between temporally distant nodes in the pose-graph that are likely to be associated with the same physical location. To evaluate the localization module independently, we simulate drift in the odometry chains by injecting noise in the individual ground truth odometry measurements.

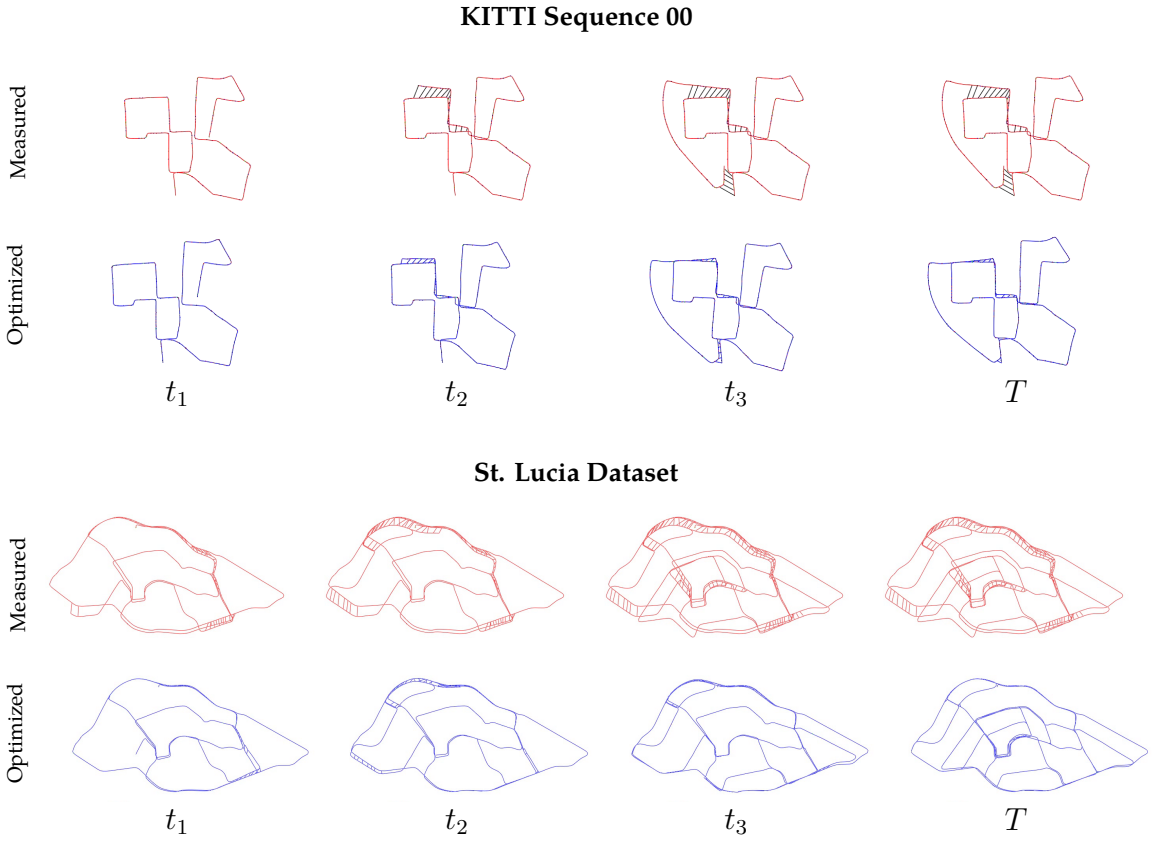


Figure 6-15: **Vision-based Pose-Graph SLAM with our learned place-recognition module ►** The two sets of plots show the measured (in red) and optimized (in blue) pose-graph for a particular KITTI and St Lucia session. The crossed edges in the measured pose-graph correspond to loop-closure candidates proposed by our learned place-recognition module. As more measurements are added and loop-closures are proposed ($t_1 < t_2 < t_3 < T$), the pose-graph optimization accurately recovers the true trajectory of the vehicle across the entire session. For both sessions, we inject odometry noise to simulate drift in typical odometry estimates.

The trajectory recovered from sequential noisy odometry measurements are shown in red, as more measurements are added ($t_1 < t_2 < t_3 < T$). With every new image, the self-supervised localization module describes the image in its learned embedding space, and queries the database to recover a similar embedding that may indicate a potential loop-closure. The loop-closures are realized as weak zero rotation and translation relative pose-constraints connecting the query node and the matched node. The recovered trajectories after the pose-graph optimization (in blue) shows consistent long-range and drift-free trajectories that the vehicle traversed.

6.6.4 Implementation Details

Network and Training We take the pre-trained Places205 AlexNet (Zhou et al. 2014b; 2016b), and set all the layers before and including *pool5* layer to be fixed, while the rest of the fully-connected layers are allowed to be fine-tuned. The resulting network is used as a base network to construct the Siamese Network with shared weights (See Section 6.4.2). We follow the distance-weighted sampling scheme as proposed by Wu et al. (2017), and sample 10 times more negative examples as positive examples. The class weights are scaled appropriately to avoid any class imbalance during training. In all our experiments, we set the sampling threshold $\tau^{\mathbf{Rt}}$ to 0.9, that ensures that identical places have considerable overlap in their viewing frustums. We train the model for 3000 epochs, with each epoch roughly taking 10s on an NVIDIA Titan X GPU. For most datasets including KITTI and St. Lucia Dataset, we train on 2-5 data sessions collected from the vehicle, and test on a completely new session.

Pose-Graph Construction and Optimization We use GTSAM[2] to construct the resulting odometry and loop-closure measurement factors proposed by our Visual-SLAM front-end for pose-graph optimization. Odometry constraints obtained from the frame-to-frame ego-motion are incorporated as a 6-DOF constraint parameterized in $SE(3)$ with $1\mathrm{e}{-3}$ rad rotational noise and $5\mathrm{e}{-2}$ m translation noise. We incorporate the loop-closure constraints as zero translation and rotation relative-pose constraint in $SE(3)$ with a weak translation and rotation covariance of 3 m and 0.3 rad respectively. The constraints are incrementally added and solved using iSAM2 (Kaess et al. 2012) as the measurements are recovered.

6.7 Discussion and Future Work

Scene Context Modeling We shall now consider a potential extension to our scene-level similarity metric learning model using an LSTM (Hochreiter and Schmidhuber 1997). By modeling sequence of scene descriptions as a sequence learning problem, we can choose to reformulate the metric learning objective with a temporal component that finds a similar invariant mapping fixed-length windows of scene descriptions. This can be particularly advantageous in improving the overall precision-recall performance of the algorithm by reducing erroneous false-positives commonly observed in these tasks. We train a trajectory similarity metric, similar to

[2] http://collab.cc.gatech.edu/borg/gtsam

the scene similarity metric previously defined, using an LSTM as the building block to model temporally evolving scene sequences. Again, due to the ready availability of cross-modal information, we train the trajectory similarity metric in an end-to-end fashion while being fully self-supervised. Figure 6-16 illustrates the scene-level similarity compared to the sequence-level similarity recovered using LSTMs.

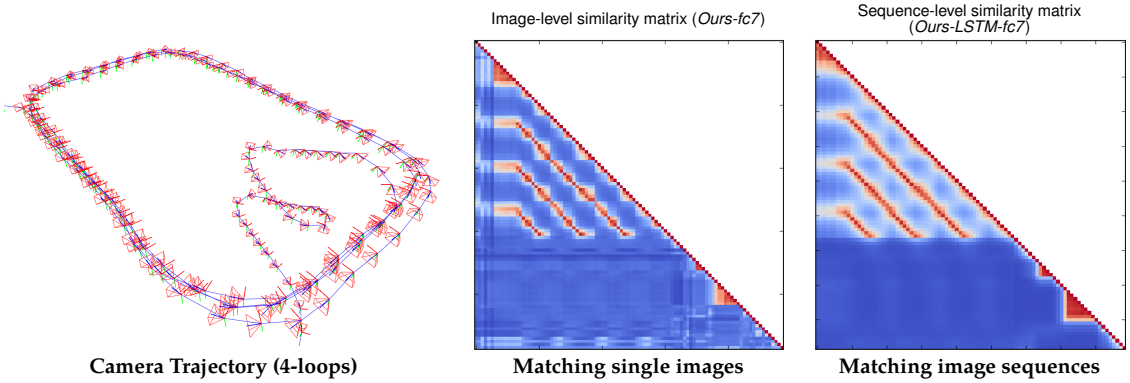


Figure 6-16: **Simultaneous trajectory and scene similarity metric learning ►** By modeling the sequence of images and their associated embeddings, our self-supervised approach to descriptor metric learning also allows us to consider scene context. This can be particularly useful in minimizing false-positives that occur in typical single image embedding and matching.

While it is convenient to describe trajectories as fixed-length sequences, the presumed chain-like topology can be limiting especially when the robot traverses the world with cyclic loops in the pose-graph topology. In fact, we would like to consider general graph topologies of robot motion, where a scene is described by pooling semantic features from the node and its kth-order neighbors. We find this to be the graph analog of a bag-of-words-based approach where the words within a document are pooled into a single order-less histogram that describes it as a whole. Motivated by this analog, we notice that a new-class of Convolutional Neural Networks called *Graph Convolutional Networks* (Kipf and Welling 2016a;b) aim to shed some light on exactly this capability. We hope to leverage some of these recent techniques for the task of self-supervised place recognition, with the potential scope for rich and hierarchical scene context modeling.

Weak-Supervision from SLAM In the context of bootstrapped learning, we envision SLAM to be especially valuable in acting as a correspondence engine for spatial and geometric understanding. We take advantage of this property and investigate the use of the SLAM solution in providing weak supervision to semantic scene understanding tasks. With a Visual-Inertial Navigation System (VINS) such as the Google Project Tango Tablet, we first extract semantic descriptions from images (using the pre-trained Places365-AlexNet model as described in the previous

section), along with its corresponding 3D pose. Since we are particularly interested in the consistency of the semantic model, we leverage the VINS solution and its associated uncertainty to identify images captured at spatially identical locations and vantage points. By leveraging the uncertainty modeled in VINS systems, we are also able to confidently sample positive samples from the experiences collected without any external supervision.

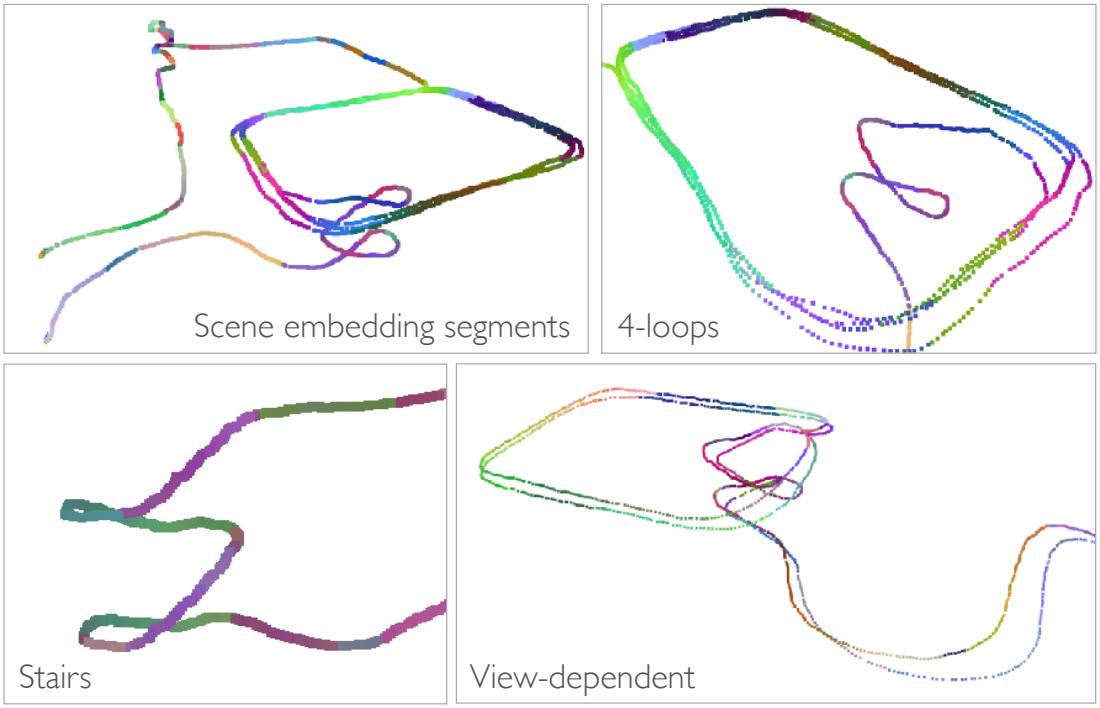


Figure 6-17: **Weakly-Supervised scene embedding for indoor localization ►** By combining Visual-Inertial Odometry (VIO) capabilities coupled with weak supervision (via ground truth fiducial markers), we envision that the resulting optimized state estimates can be re-purposed to provide useful supervisory-signals to the proposed metric learning approach.The figures show various trajectories collected with a VIO-capable device, with the colors indicating the "embedding" of the semantic objects contained in the image collected at those locations. The learned semantic embedding of images (as represented by its color) are consistent across multiple passes along the same corridor. Furthermore, the scene embedding is also able to discern scenes from each other based on the set of objects contained within those scenes.

As elaborated in this chapter, this SLAM-supervised bootstrapped learning capability allows us to learn the *semantic measure* between images enabling strong and robust scene understanding performance in our trained models. We illustrate initial results in this line of work in Figure 6-17. Via weak-supervision obtained from a VIO sensor, we are able to learn a feature embedding space (represented by the colors of nodes) that is able to extract consistent semantic embeddings (similar colors) for spatially identical vantage points.

Resource-Aware Computation None of the existing learning-based approaches

for place recognition model their task performance in a resource-aware manner. In their work (Milford 2013), the model is explicitly reduced in order to determine place-recognition under severe resource-constraints (memory, computation etc). We envision that robots one day can learn to perform the same tasks with significantly fewer resources, without having to be explicitly modeled to perform such optimized behavior. To the best of our knowledge, no such mechanism exists for automatically tuning and refining place recognition for the desired resource constraints. We find this capability to be particularly valuable, and hope to pursue future work along these directions.

6.8 Chapter Summary

In this chapter, we developed a self-supervised approach to place recognition in robots. By leveraging the synchronization between sensors, we propose a method to transfer and learn a metric for image-image similarity in an embedded space by sampling corresponding information from a GPS-aided SLAM solution. Furthermore, we show that the newly learned embedding can be particularly powerful for the task of visual place-recognition as the embedded distances are well-calibrated for efficient indexing and accurate retrieval. Effectively, the methods developed in this chapter and in the previous chapter enable scalable training of Visual-SLAM front-ends without having to develop human-devised visual feature descriptors and hand-tuned hyper-parameters during deployment.

Chapter 7

Future Directions

7.1 Spatially and Semantically-Aware Robot Databases

Given the large volume of mixed information robots collect, there exists a strong need for data persistence and query capabilities to enable the vision of life-long learning and autonomy. Moreover, rich geometric and semantic relationships that mobile robots structurally encode can be directly embedded into a *context-aware* database. In some initial work (Fourie et al. 2017), we advocate for a graph-database based abstraction that affords various SLAM-aware capabilities. These include massively parallelizable inference schemes for experience-based learning, querying, and multi-robot mapping, that are typically difficult to encode in a monolithic architecture. Furthermore, it is especially appealing that this extensible star-architecture abstraction could be leveraged to potentially decouple light-weight robot front-ends from their elastic heterogeneous computing back-ends with many-core, many-GPU, many-machine or hybrid configurations. With the advent of GPU-based databases and accessible Domain-Specific Languages (DSLs) for high-performance computing, the scope for a next-generation computational engine for large-scale robot learning seems extremely promising.

7.2 Expressive Language for Robot Data Querying

Domain-Specific Languages (DSLs) provide a decoupled abstraction between an algorithm and its computation schedule. Their compilers are able to synthesize high-performance and optimized implementations for several hardware architec-

Figure 7-1: **Semantic foveation with SLAM-aware backends ►** We illustrate the potential of SLAM-aware databases that allow for semantically relevant and efficient queries in a scene. In this example, we are able to query the database for various views of the same piece of artwork that has previously been recognized. By making the database, both semantically and spatially-aware, we are able to query the semantics in a scene given its spatial location or query the different spatial locations of a given semantic entity. Furthermore, these databases can be temporally persistent that allows multi-session SLAM solutions to provide strong relational connectivity within the graph database.

tures, including multi-core, and heterogeneous architectures (x86, ARM, CUDA), all while maintaining the exact same source-code implementation. Inspired by these recent trends, we are particularly interested in developing similar domain-specific expressions and abstractions for large-scale robot data computation. Through some recent work (Moll et al. 2017), we hope to develop an expressive DSL for mobile robots that can allow complex machine-learning workflow specifications on high-volume robot data, while abstracting performance decisions to a massively parallel heterogeneous back-end. We expect these tools and abstractions to heavily leverage modern GPGPU hardware, and foresee it being especially valuable to the robotics community as we enter an era for petabyte-level machine learning.

7.3 Self-Supervised Cross-Modal Learning in Robots

Self-supervised learning provides a compelling solution to the life-long autonomy problem in robots. If robots are to constantly learn from their experiences, they need to be able to query their experiences and the physical world that they interact with, both from a spatial and semantic context. In the near future, we expect these

Figure 7-2: **Automatic labeling for an image-based trajectory planner from hindsight experience ►** By taking advantage of experience collected from human-supervised driving, we are able to project the vehicle's future trajectory onto the current camera image, and automatically recover ground truth trajectories and their associated image for self-supervising an image-based trajectory planner.

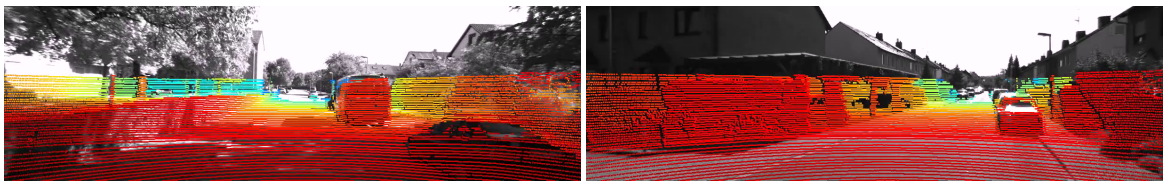


Figure 7-3: **Automatic labeling using LiDAR for camera-based scene reconstruction ►** Using the knowledge of Camera-LiDAR calibration, we are able to project the point cloud reconstructed by the LiDAR onto onto the camera's image, and automatically recover ground truth disparity maps for self-supervising image-based disparity estimation algorithms (Multi-view or Temporal reconstruction).

experience-databases to be *SLAM-aware* with a rich semantic understanding of the mapped environment it perceived in the past. Furthermore, these experiences need to be able to queryable both from a semantic and spatial context, taking advantage of both the representations simultaneously.

As robots explore their environment equipped with several sensors, they continuously collect a multitude of measurement modalities that may or may not be correlated. However, as task-driven learning continues to incorporate all sensor measurements in a sound manner, it becomes cumbersome to do so as we would expect the number of on-board sensors to grow rapidly in the years to come. With this in mind, we treat the problem of characterizing and learning to autonomously incorporate new sensor measurements into the robot's task without having to explicitly specify its measurement model.

We focus on the ability to jointly consider raw measurements from various sensors, and reason over their compatibility towards a task. We think that robots need to be able to self-supervise themselves in learning to accomplish certain tasks, such as ego-motion and place-recognition, with newly introduced sensors that have not been characterized yet. This bootstrapped technique allows for an autonomous agent to leverage its existing estimation, exploration, and navigation strategies to extract meaningful cues that may allow it to accomplish similar tasks in the future.

In figures 7-2 and 7-3, we illustrate two such examples of cross-modal learning: (a) automatic labeling for an image-based trajectory planner from hindsight experience, and (b) automatic depth labeling using LiDAR for camera-based scene reconstruction. Both these examples exemplify the potential of bootstrapped learning in the context of autonomous systems, where a rich set of cross-modal information is readily available for self-supervision.

7.4 Life-long Learning with Simulation

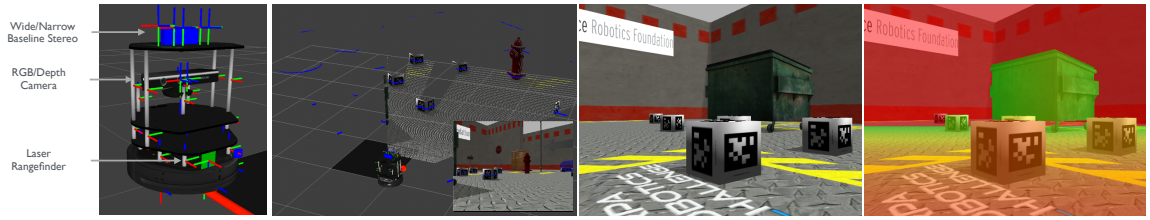


Figure 7-4: **Turtlebot simulation with ground truth depth, scene flow, odometry** ▶ Simulation environments such as Gazebo can be a valuable resource for cross-modal learning, and transfer learning in the context of self-supervised and life-long autonomy. Readily available ground truth information such as depth, scene semantics and geometry can be used to train models in simulation before they are fine-tuned for real-world sensory information.

Autonomous exploration and mapping strategies in robots admit a broad range of experiences that can be inevitably used towards learning new task-specific representations. We explore self-supervised learning in robots using both simulated and real-world robots. We introduce a dataset-generating pipeline that allows autonomous collection of a robot navigating an unknown space, along with ground truth pose information to help bolster the labeled data concern that most supervised (including and especially deep-learning) techniques suffer from. We model the simulation environment[1] as closely as possible to our physical Turtlebot in order to ensure the ease of transfer learning from a simulated robot environment to a stochastically-driven environment. For e.g. we would like to learn a range of tasks such as estimation of depth, flow, location, odometry from a simulated world with readily available ground truth (See Figure 7-4). Additionally, with the availability of a simulation engine that can emulate the robot's configuration space and sensor characteristics, we expect simulation to real-world transfer learning to be more realistic and manageable with minimal fine-tuning required once these mechanisms are deployed on the physical robot.

[1]Gazebo http://gazebosim.org

Chapter 8

Conclusion

SLAM is a fundamental capability in mobile robots, and has been typically considered in the context of aiding mapping and navigation tasks. Due to the memory and run-time complexity of the full SLAM formulation, most practical real-time solutions only implement a specialized variant of SLAM that is geared for the specific robot application. In this thesis, we advocate for the use of SLAM as a supervisory signal to further the perceptual capabilities in robots. Through the concept of *SLAM-supported object recognition*, we develop the ability for robots equipped with a single camera to be able to leverage their SLAM-awareness (via Monocular Visual-SLAM) to better inform object recognition within its immediate environment. Additionally, by maintaining a spatially-cognizant view of the world, we find our SLAM-aware approach to be particularly amenable to few-shot object learning. We show that a SLAM-aware, few-shot object learning strategy can be especially advantageous to mobile robots, and is able to learn object detectors from a reduced set of training examples. In the future, we expect that an object-level abstraction for semantic mapping can be particularly useful in *Recognition-supported SLAM*, where the SLAM objective is described over the set of objects contained in the environment. We expect this key insight to be imperative in scaling up visual-SLAM solutions to extremely long-term settings (hours, days, and months), while being considerably more robust to semi-static, and dynamic environments.

Implicit to realizing modern visual-SLAM systems is its choice of map representation. It is imperative that the map representation is crucially utilized by multiple components in the robot's perception stack, while it is constantly optimized as more measurements are available. Thus, we seek a unified intermediate representation for maps that can benefit the vision-based mapping and planning stacks in the

robot. State-of-the-art solutions to feature-based visual-SLAM reconstruct sparse 3D landmarks, but fail to recover much geometric structure from these scenes to afford vision-based navigation with these features. Other variants such as semi-dense reconstructions may potentially afford planning-in-the-loop, however, with the heavy-computational needs that these systems currently desire, it is still open to discussion if these representations may be suitable for scalable mapping purposes. Motivated by the need for a unified map representation in vision-based mapping and navigation, we develop an *iterative and high-performance mesh-reconstruction algorithm* from stereo imagery. We envision that in the future, these tunable mesh representations can potentially enable robots to quickly reconstruct their immediate surroundings while being able to directly plan in them and maneuver at high-speeds.

In order to robustly operate in dynamic and changing environments, robots need to be able to leverage their previous experiences and continuously adapt to their immediate surroundings, improving their overall task-performance while simultaneously optimizing for model efficiency. Visual SLAM implementations have been realized in a variety of ways, ranging from sparse, feature-based methods to semi-dense and dense, direct methods lately. While these methods have monotonically improved in their overall accuracy and robustness, a fair amount of hyper-parameter tuning is necessary to adapt these implementations to a new environment or operating regime. While most visual-SLAM front-ends explicitly encode application-specific constraints for accurate and robust operation, we advocate for a fully *self-supervised* solution to developing application-specific Visual-SLAM front-ends. Again, by taking advantage of GPS-aided SLAM as a supervisory signal, we leverage the fused trajectory estimates in vehicles to self-supervise the task of *visual ego-motion estimation* and *vision-based re-localization*.

We envision that self-supervised and weakly-supervised solutions to task learning shall have far-reaching implications in several domains, especially in the context of life-long learning in autonomous systems. Furthermore, we expect these techniques to seamlessly operate under resource-constrained situations in the near future by leveraging well-studied solutions in model reduction and dynamic model architecture tuning. With the availability of multiple sensors on these autonomous systems, we also foresee *bootstrapped task learning* to potentially enable robots to learn from experience, while being able to deal with redundancy and fault-tolerance, all within the same framework.

Bibliography

A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, 2008.

R. Arandjelovic and A. Zisserman. All about VLAD. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013.

R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch. Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation. In *Embedded Computer Systems (SAMOS), 2010 International Conference on*. IEEE, 2010.

S. Y. Bao and S. Savarese. Semantic structure from motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011.

S. Y. Bao, M. Bagra, Y.-W. Chao, and S. Savarese. Semantic structure from motion with points, regions, and objects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable. Tracking and data association. *The Journal of the Acoustical Society of America*, 87(2):918–919, 1990.

A. J. Barry and R. Tedrake. Pushbroom stereo for high-speed navigation in cluttered environments. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 2015.

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006.

S. Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker, 2007.

C. M. Bishop. Mixture density networks. 1994.

M. Bleyer and C. Breiteneder. Stereo Matching - State-of-the-Art and Research Challenges. In *Advanced Topics in Computer Vision*, pages 143–179. Springer, 2013.

M. Bleyer, C. Rhemann, and C. Rother. Patchmatch Stereo - Stereo matching with slanted support windows. In *BMVC*, volume 11, pages 1–11, 2011.

L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

I. Bogun, A. Angelova, and N. Jaitly. Object recognition from short videos for robotic perception. *arXiv preprint arXiv:1509.01602*, 2015.

A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*. IEEE, 2007.

S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic SLAM. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1722–1729. IEEE, 2017.

J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.

Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010.

R. O. Castle, G. Klein, and D. W. Murray. Combining monoSLAM with object recognition for scene augmentation using a wearable camera. *Image and Vision Computing*, 28(11), 2010.

K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2011.

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

P. Cheeseman, R. Smith, and M. Self. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotic Research*, pages 467–474, 1987.

X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015a.

Z. Chen, S. Lowry, A. Jacobson, Z. Ge, and M. Milford. Distance metric learning for feature-agnostic place recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2556–2563. IEEE, 2015b.

Z. Chen, A. Jacobson, N. Sunderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. Deep learning features at scale for visual place recognition. *arXiv preprint arXiv:1701.05105*, 2017.

M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

W. Churchill and P. Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4525–4532. IEEE, 2012.

T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci. Evaluation of non-geometric methods for visual odometry. *Robotics and Autonomous Systems*, 62(12):1717–1730, 2014.

J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel. Towards semantic SLAM using a monocular camera. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2011.

J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization. In *Structure from Motion using the Extended Kalman Filter*, pages 33–63. Springer, 2012.

R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni. VINet: Visual-Inertial odometry as a sequence-to-sequence learning problem. AAAI, 2016.

A. Collet and S. S. Srinivasa. Efficient multi-view object recognition and full pose estimation. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 2010.

A. Concha and J. Civera. DPPTAM: Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

P. Corke, D. Strelow, and S. Singh. Omnidirectional visual odometry for a planetary rover. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 4007–4012. IEEE, 2004.

G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia. Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation. *IEEE Robotics and Automation Letters*, 1(1):18–25, 2016.

G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, 2004.

M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

M. J. Cummins and P. M. Newman. FAB-MAP: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.

J. P. Cunningham and Z. Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16:2859–2900, 2015.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005.

J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013.

J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the VLAD image representation. In *Proceedings of the 21st ACM international conference on Multimedia*, 2013.

F. Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

F. Dellaert, M. Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

L. Devroye. Sample-based non-uniform random variate generation. In *Proceedings of the 18th conference on Winter simulation*, pages 260–265. ACM, 1986.

J. Dong, X. Fei, and S. Soatto. Visual-inertial-semantic scene representation for 3d object detection. *arXiv preprint arXiv:1606.03968*, 2016.

A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. European Conf. on Computer Vision (ECCV)*. Springer, 2014.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *Int'l J. of Computer Vision*, 88(2):303–338, 2010.

L. Fe-Fei et al. A bayesian approach to unsupervised one-shot learning of object categories. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1134–1141. IEEE, 2003.

L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.

M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 15–22. IEEE, 2014.

D. Fourie, S. Claassens, S. Pillai, R. Mata, and J. Leonard. *SLAMinDB*: Centralized Graph Databases for Mobile Robotics. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017. (Link).

F. Fraundorfer and D. Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.

F. Fraundorfer, C. Engels, and D. Nistér. Topological mapping, localization and navigation using image collections. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3872–3877. IEEE, 2007.

F. Fraundorfer, D. Scaramuzza, and M. Pollefeys. A constricted bundle adjustment parametrization for relative scale estimation in visual odometry. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1899–1904. IEEE, 2010.

D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

D. Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5), October 2012. ISSN 1552-3098. doi: 10.1109/TRO.2012.2197158.

D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel. Real-time monocular object SLAM. *Robotics and Autonomous Systems*, 75:435–449, 2016.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

S. K. Gehrig, F. Eberli, and T. Meyer. A real-time low-power stereo vision engine using semi-global matching. In *Computer Vision Systems*. Springer, 2009.

A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Computer Vision–ACCV 2010*. Springer, 2011a.

A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3D reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011b.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

G. Georgakis, M. A. Reza, and J. Košecka. Rgb-d multi-view object detection with object proposals and shape context. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4125–4130. IEEE, 2016.

R. Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014a.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014b.

X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth. FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3507–3512. IEEE, 2010.

R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005.

A. Graves. Stochastic backpropagation through mixture density distributions. *arXiv preprint arXiv:1607.05690*, 2016.

J. J. Guerrero, R. Martinez-Cantin, and C. Sagüés. Visual map-less navigation based on homographies. *Journal of Robotic Systems*, 22(10):569–581, 2005.

S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *Proc. European Conf. on Computer Vision (ECCV)*. 2014.

B. Hariharan and R. Girshick. Low-shot visual object recognition. *arXiv preprint arXiv:1606.02819*, 2016.

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

K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *arXiv preprint arXiv:1703.06870*, 2017.

G. Hee Lee, F. Faundorfer, and M. Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, 2013.

H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2. IEEE, 2005.

H. Hirschmüller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31 (9):1582–1599, 2009.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

D. Honegger, H. Oleynikova, and M. Pollefeys. Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2014.

A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34 (3):189–206, 2013.

J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In M. Valstar, A. French, and T. Pridmore, editors, *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3946–3952. IEEE, 2008.

V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61 (8):721–738, 2013.

H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010.

H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011.

H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2012.

E. S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, page 0278364911430419, 2011.

M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–390. IEEE, 2003.

A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946, 2015.

A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. P. Kingma and M. Welling. Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016a.

T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016b.

B. Kitt, A. Geiger, and H. Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *Intelligent Vehicles Symposium*, pages 486–492, 2010.

G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

L. Kneip, P. Furgale, and R. Siegwart. Using multi-camera systems in robotics: Efficient solutions to the n-PnP problem. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3770–3776. IEEE, 2013.

K. Konda and R. Memisevic. Learning visual odometry with a convolutional network. In *International Conference on Computer Vision Theory and Applications*, 2015.

K. Konolige and M. Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

K. Konolige, M. Agrawal, and J. Sola. Large-scale visual odometry for rough terrain. In *Robotics research*, pages 201–212. Springer, 2010a.

K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *The International Journal of Robotics Research*, 29(8):941–957, 2010b.

K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent. Efficient sparse pose adjustment for 2d mapping. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 22–29. IEEE, 2010c.

J. Košecká, F. Li, and X. Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *AAAI/IAAI*, pages 174–180, 2002.

B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2012.

B. Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4): 287–364, 2013.

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g 2 o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE, 2011.

K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 2011.

K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3D scenes. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 2012.

K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3D scene labeling. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*. IEEE, 2014.

Y. Latif, C. Cadena, and J. Neira. Robust loop closing over time. *Robotics: Science and Systems VIII*, page 233, 2013.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2. IEEE, 2006.

V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

B. Liang and N. Pears. Visual navigation using planar homographies. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 205–210. IEEE, 2002.

R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–I. IEEE, 2002.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, pages 61–62, 1987.

D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.

B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

S. Lynen, M. Bosse, P. Furgale, and R. Siegwart. Placeless place-recognition. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 303–310. IEEE, 2014.

L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

W. Maddern, M. Milford, and G. Wyeth. CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory. *The International Journal of Robotics Research*, 31(4):429–451, 2012.

W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, page 0278364916679498, 2016.

L. H. Matthies. Dynamic stereo vision. 1989.

C. Mei, G. Sibley, and P. Newman. Closing loops without places. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3738–3744. IEEE, 2010.

M. Milford. Vision-based place recognition: how low can you go? *The International Journal of Robotics Research*, 32(7):766–789, 2013.

M. J. Milford and G. F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE, 2012.

O. Moll, A. Zalewski, S. Pillai, S. Madden, M. Stonebraker, and V. Gadepally. Exploring big volume sensor data with Vroom. In *Very Large Data Bases (VLDB), Demo*, 2017. (Link).

H. P. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.

A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Robotics and automation, 2007 IEEE international conference on*, pages 3565–3572. IEEE, 2007.

R. Mur-Artal and J. Tardos. Probabilistic semi-dense mapping from highly accurate feature-based monocular SLAM. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

R. Mur-Artal, J. Montiel, and J. D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *arXiv preprint arXiv:1502.00956*, 2015.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on robotics and automation*, 17(6):890–897, 2001.

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1180–1187. IEEE, 2006.

D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. Ieee, 2006.

D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652. IEEE, 2004.

D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.

A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

C. F. Olson, L. H. Matthies, H. Schoppers, and M. W. Maimone. Robust stereo ego-motion for long distance navigation. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 453–458. IEEE, 2000.

M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391. IEEE, 2010a.

F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. European Conf. on Computer Vision (ECCV)*. Springer, 2010b.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

S. Pillai and J. Leonard. Monocular SLAM Supported Object Recognition. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015. (Link).

S. Pillai and J. Leonard. Towards Visual Ego-motion Learning in Robots. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, Vancouver, Canada, 2017a. IEEE.

S. Pillai and J. Leonard. Self-Supervised Visual Place Recognition in Mobile Robots. In *Proc. of Workshop on Learning for Localization and Mapping, IEEE Intelligent Robots and Systems (IROS)*, 2017b. Accepted to appear.

S. Pillai, S. Ramalingam, and J. Leonard. High-Performance and Tunable Stereo Reconstruction. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016. (Link).

S. Ramalingam, M. Antunes, D. Snow, G. Hee Lee, and S. Pillai. Line-sweep: Cross-Ratio for Wide-Baseline Matching and 3D Reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Link).

F. Ramos and L. Ott. Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730, 2016.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

R. S. Ren, K. He, and R. Faster. Towards real-time object detection with region proposal networks, arxiv preprint. *arXiv preprint arXiv:1506.01497*.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

R. Roberts, C. Potthast, and F. Dellaert. Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 57–64. IEEE, 2009.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 430–443. Springer, 2006.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*. IEEE, 2011.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013.

D. Scaramuzza. 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International journal of computer vision*, 95(1):74–85, 2011.

D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011.

D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1413–1419. IEEE, 2009a.

D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4293–4299. IEEE, 2009b.

D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int'l J. of Computer Vision*, 47(1-3), 2002.

K. Schauwecker, R. Klette, and A. Zell. A new feature detector and stereo matching method for accurate high-performance sparse stereo matching. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2012.

M. Schönbein and A. Geiger. Omnidirectional 3d reconstruction in augmented manhattan worlds. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 716–723. IEEE, 2014.

A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

G. Singh and J. Kosecka. Visual loop closing using gist descriptors in manhattan world. In *ICRA Omnidirectional Vision Workshop*, 2010.

J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. Int'l. Conf. on Computer Vision (ICCV)*. IEEE, 2003.

K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2017.

H. Strasdat, J. Montiel, and A. J. Davison. Real-time monocular SLAM: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664. IEEE, 2010.

H. Strasdat, A. J. Davison, J. M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2352–2359. IEEE, 2011.

C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *arXiv preprint arXiv:1707.02968*, 2017.

N. Sünderhauf and P. Protzel. Brief-gist-closing the loop by simple means. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1234–1241. IEEE, 2011.

N. Sünderhauf, P. Neubert, and P. Protzel. Are we there yet? challenging SeqSLAM on a 3000 km journey across all four seasons. In *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*, page 2013, 2013.

N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.

N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid. Meaningful maps − object-oriented semantic mapping. *arXiv preprint arXiv:1609.07849*, 2016.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

K. Tateno, F. Tombari, and N. Navab. When 2.5 d is not enough: Simultaneous reconstruction, segmentation and recognition on dense slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2295–2302. IEEE, 2016.

A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool. Towards multi-view object class detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2. IEEE, 2006.

S. Thrun and J. J. Leonard. Simultaneous localization and mapping. In *Springer handbook of robotics*, pages 871–889. Springer, 2008.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

P. H. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International journal of computer vision*, 24(3):271–300, 1997.

P. H. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustmentâĂŤa modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

T. Tuytelaars, K. Mikolajczyk, et al. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 3(3):177–280, 2008.

J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *Int'l J. of Computer Vision*, 104(2), 2013.

I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1023–1029. Ieee, 2000.

K. E. van de Sande, C. G. Snoek, and A. W. Smeulders. Fisher and VLAD with FLAIR. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014.

O. Veksler. Dense features for semi-dense stereo correspondence. *Int'l J. of Computer Vision*, 47(1-3), 2002.

P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

H. Wang, K. Yuan, W. Zou, and Q. Zhou. Visual odometry based on locally planar ground assumption. In *2005 IEEE International Conference on Information Acquisition*, pages 6–pp. IEEE, 2005.

L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez. Not seeing is also believing: Combining object and metric spatial information. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1253–1260. IEEE, 2014.

L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez. Data association for semantic world modeling from partial views. *The International Journal of Robotics Research*, 34(7):1064–1082, 2015.

C.-Y. Wu, R. Manmatha, A. J. Smola, and P. KrÃďhenbÃijhl. Sampling Matters in Deep Embedding Learning, 2017.

Y. Xiang and D. Fox. Da-rnn: Semantic mapping with data associated recurrent neural networks. *arXiv preprint arXiv:1703.03098*, 2017.

E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15: 505–512, 2003.

Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-depth super resolution for range images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 151–158. Springer, 1994.

P. H. Zadeh, R. Hosseini, and S. Sra. Geometric mean metric learning. In *International Conference on Machine Learning (ICML)*, 2016.

J. Žbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui. Meshstereo: A global stereo model with mesh alignment regularization for view interpolation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2057–2065, 2015.

Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene CNNs. *arXiv preprint arXiv:1412.6856*, 2014a.

B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014b.

B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016a.

B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*, 2016b.

X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *Proc. European Conf. on Computer Vision (ECCV)*. Springer, 2010.

C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.