UNIVERSITY OF CALIFORNIA

Los Angeles

Exploiting Regularities to Recover 3D Scene Geometry

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Alex King Lap Wong

2019

ABSTRACT OF THE DISSERTATION

Exploiting Regularities to Recover 3D Scene Geometry

by

Alex King Lap Wong
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2019
Professor Stefano Soatto, Chair

Recovering three-dimensional (3D) scene geometry from images is an ill-posed problem due to the loss of the extra dimension in the process of projection. Hence, the solution hinges on the choice of regularization or prior assumptions about the scene. We study the effects of various regularization schemes on the 3D reconstruction problem under two problem settings, single image depth prediction and sparse depth completion. Obtaining the 3D scene from a single image is literally an impossible task as there are infinitely many 3D scenes compatible with the given image – making both problem settings great candidates for evaluating the influence of a given regularizer. We begin by examining the relation between data fidelity residual and the degree of regularization to form a spatially and temporally varying adaptive weighting scheme for single image depth prediction. We additionally explore the use of gravity, as a supervisory signal, to induce a prior on the pose of objects populating a scene. To extend the use case to real world applications, we develop visioning systems to infer dense depth from an image with associated sparse depth measurements. We leverage the abundance of synthetic data to obtain a learned prior for guiding the learning process. Conscious of the limitations of current depth completion methods in processing sparse depth and their growth in parameters, we propose a two-stage approach that approximates the scene with a "scaffolding" and refines the approximation with a simple light-weight network. The result is a small and fast, but accurate visioning system that fits in an embedded system. To enable an agent to continuously learn, our systems are completely unsupervised and learn by exploiting geometry and known regularities.

The dissertation of Alex King Lap Wong is approved.

Alan L. Yuille

Quanquan Gu

Wei Wang

Stefano Soatto, Committee Chair

University of California, Los Angeles

2019

To my parents and my brother.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

To begin, I would like to thank my Ph.D. advisor, Professor Stefano Soatto, for sparking my interest in 3D vision and introducing me to geometric problems. I appreciate the patience and trust he has given me over the past two years and I thank him for his generosity and the gamble he took when I approached him about joining the Vision Lab. I have learned many invaluable lessons through our discussions, namely, the importance of impact and understanding a problem outside of pure academics. More importantly, I have learned to approach a problem from the first principles. To solve a problem, one must devise the solution after a thorough understanding of the problem. However intuitive that may sound, it is a common temptation to first have a solution and to then seek a problem that fits the solution. Lastly, I thank Professor Soatto for his continual support and the numerous opportunities he has given me.

I would also like to thank my Master's and Ph.D. advisor, Professor Alan Loddon Yuille, for introducing me to one-shot learning and providing me with the freedom and support to explore an array of ideas, ultimately leading to our study on protrusions and their semantic significance on object parts. What began as a spontaneous question about hand-written characters became a series of intriguing discussions and projects over the four years in the Computational Cognition, Vision, and Learning group (CCVL).

I would like to express my gratitude towards my committee members, Professor Alan Loddon Yuille, Professor Quanquan Gu, Professor Wei Wang and Professor Stefano Soatto for their support and advice, and for accommodating my schedule.

I want to thank Professor Byung-Woo Hong, who mentored me and introduced me to adaptive regularization. As I was unfamiliar with the topic, I was grateful for his patience and the many late night discussions about various research topics. Byung-Woo would spend hours to step through a problem with me. I also appreciate his open-mindedness to my spontaneous ideas and his continual support and encouragement. I would also like to thank him for all of his advice on life and career choices.

I am grateful for Professor Michael Shindler, who along with Dr. Adam Meyerson kick-

started my career in research. Had I never taken algorithms with Michael as the teaching assistant, I would never have gone down this career path. While I was never his best student, Michael mentored and introduced me to the streaming k-means problem, for which our work won the Outstanding Student Paper at NeurIPS. To this day, we both claim the other had came up with the idea for the paper.

There were many individuals I would like to thank during my time at UCLA. Firstly, I want to thank my good friend and collaborator Xiaohan Fei for sparking my interest in multi-sensor fusion. Coming from different background, Xiaohan and I exchanged many ideas and fused visual-inertial odometry with learning based 3D scene geometry. Together, we co-authored several papers on learning-based sensor fusion, one of which won the Best Paper Award in Robot Vision at ICRA. Xiaohan is technically sound with an exceptional work ethic. Our collaboration brought out some of my best work. Secondly, I would like to thank another good friend and collaborator Brian Richard Taylor for encouraging me to pursue a Ph.D. My first exposure to computer vision was seeing a set of figures that Brian generated for his project. I am grateful for his support and advice over the years. Additionally, I would like to thank Brian for proofreading my manuscripts the day before the submission deadline on numerous occasions. I would like to thank Safa Cicek for introducing me to semi-supervised learning and domain adaption. Our discussions led to our collaboration on several projects on domain adaption for learning 3D geometry. I would also like to thank Yanchao Yang for our numerous discussions, they gave clarity. I am grateful for our collaboration on our work on depth completion.

I would like to acknowledge other members of the UCLA Vision Lab: Alessandro Achille, Kareem Ahmed, Alper Ayvaci, Xinzhu Bei, Pratik Chaudhari, Isaac Deutsch, Shay Deutsch, Jingming Dong, Alhussein Fawzi, Aditya Golatkar, Tong He, Vasiliy Karasev, Nikos Karianakis, Brian Lai, Konstantine Tsotsos, Alexandre Tiard, Stephanie Tsuei, Albert Zhao and Peng Zhao. I thank them for welcoming me and for their support and helpful discussions. I would like to acknowledge Kareem Ahmed, Xinzhu Bei, Safa Cicek, Shay Deutsch, Xiaohan Fei, Tong He, Alexandre Tiard, Stephanie Tsuei, Yanchao Yang, Albert Zhao and Peng Zhao for proofreading my conference submission manuscripts.

I would also like to acknowledge members of the Computational Cognition, Vision, and Learning (CCVL) group: Boyan Bonev, Liang-Chieh Chen, Xianjie Chen, Xiaochen Lian, Chenxi Liu, Junhua Mao, Vittal Premachandran, Weichao Qiu, Zhou Ren, Jianyu Wang, Peng Wang, Fangting Xia, Cihang Xie, Lingxi Xie, Zhishuai Zhang, Zhuotun Zhu, and Jun Zhu. I would also like them for their support and proofreading my manuscripts. I specifically want to thank Boyan Bonev, Liang-Chieh Chen, Xianjie Chen, Xiaochen Lian, Junhua Mao, Zhou Ren, Jianyu Wang, Peng Wang, Fangting Xia, and Jun Zhu for their mentorship during my time in CCVL.

I am grateful for my time spent at LinkedIn. I thank my managers Cindy Chen, Thomas Feng, Sanjay Kshetramade, Saung Li, James Margatan, Aviad Pinkovezky, Warren Quach and Melanie Wong for the thoughtful projects and career advice. I would like to acknowledge my colleagues Collin Adams, Brian Chang, Bryan Cheng, You Cheng, Khanh Dao, Jackson Dean, Viman Deb, Brent Dimapilis, Alvin Huynh, Kevin Jia, Ashley Jin, Victor Korshun, Lanhui Long, Hao Liu, Sourav Maji, Raul Rivero, Edilberto Ruvalcaba, Krissa Santos, Tim Santos, Shivam Sharma, Hong Tam, Cynthia Tsai, Yiheng Wang, Haochen Wei, KC Winz, Jeffrey Wong, Feiyu Yu, Chenhui Zhai and Ping Zhu. They have given me invaluable advice and have taught me almost everything I know about engineering.

Finally, I would like to thank my parents and my brother for their continual encouragement and support. I am grateful for their patience and their willingness to support my decisions and forgive my mistakes. Their patience have in turn taught me patience and their support have allowed me to be curious and to see my failures as an opportunity for growth. They are my rock, and I dedicate this thesis to them.

VITA

2007–2012	B.S. in Computer Science, University of California, Los Angeles
2012–2015	M.S. in Computer Science, University of California, Los Angeles
2015–Present	Graduate Researcher at the University of California, Los Angeles

PUBLICATIONS

A. Wong, X. Fei, B.W. Hong, and S. Soatto. *An Adaptive Framework for Learning Unsupervised Depth Completion.* Preprint. November 15, 2019.

A. Wong†, X. Fei†, and S. Soatto. *VOICED: Depth Completion from Inertial Odometry and Vision.* University of California, Los Angeles Technical Report #190001. March 22, 2019.

A. Wong, B.W. Hong, and S. Soatto. *Bilateral Cyclic Constraint and Adaptive Regularization for Unsupervised Monocular Depth Prediction.* In the Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR). June 2019.

Y. Yang, **A. Wong**, and S. Soatto. *Dense Depth Posterior (DDP) from Single Image and Sparse Range.* In Proceedings of Computer Vision and Pattern Recognition (CVPR). June 2019.

X. Fei, **A. Wong**, and S. Soatto. *Geo-Supervised Visual Depth Prediction.* In the Robotics and Automation Letters (RA-L) 2019 and the Proceedings of International Conference on

Robotics and Automation (ICRA). May, 2019. **Best Paper Award in Robot Vision** (ICRA).

A. Wong, B. Taylor, and A. Yuille. *Exploiting Protrusion Cues for Fast and Effective Shape Modeling via Ellipses.* In the Proceedings of British Machine Vision Conference (BMVC). September 2017.

A. Wong, and A. Yuille. *One Shot Learning via Compositions of Meaningful Patches.* In the Proceedings of International Conference on Computer Vision (ICCV). December 2015

X. Dong, Y. Zhu, W. Li, L. Xie, **A. Wong**, and A. Yuille. *Fidelity-Naturalness Evaluation of Single Image Super Resolution.* Preprint. November 21, 2015.

M. Shindler, **A. Wong**, and A. Meyerson. *Fast and Accurate k-Means for Large Datasets.* In the Proceedings of Neural Information Processing Systems (NeurIPS). December 2011. **Outstanding Student Paper Award**.

[†] Equal contributors.

CHAPTER 1

Introduction

We, human beings, as intelligent agents, are constantly interacting with our environment. From fetching a cup of water to driving a car, we take in an immense amount of information from our visual world to accomplish these tasks. We see colors, and from that we are able to identify shapes, gauge distance and even determine the scale of far-view objects. This is made possible not only because our brains dedicate hundreds of millions of neurons for visual processing alone, but also because we have prior knowledge about the construct of our visual world (e.g. the road is flat, an average person is about 1.5 to 1.8 meters tall), allowing us to infer properties of the scene, even if we are given novel imagery.

An autonomous agent, like us, needs to successfully navigate an environment in order to accomplish a task. Instead of our eyes, a common visioning system consists of cameras and light sensors. For an agent to navigate an environment, it is given images (for simplicity) and is required to reason about numerous questions that may include, but are not limited to: whether or not there is an object, whether an object is a chair, and how far away is the object. We are interested in recovering the shape of the scene (e.g. where are the surfaces and how far are they from the agent).

An agent may learn to perceive its environment in many ways. One may give an agent images of a scene and, for every pixel in the images, tell the agent how far (depth) the agent is from the corresponding point in space. Such is a supervised learning framework and requires per-pixel ground-truth annotations, that is often unavailable, for each image. Moreover, the ground-truth annotations, when available, are expensive to acquire and are, often times, the result of heavy data-processing and aggregation of information from many frames – still the quality may be poor and annotations may only over cover a portion of the image. This type of supervision is not scalable. We instead exploit the seemingly infinite amount of unlabeled imagery to train an agent to infer depth in an *unsupervised* manner by exploiting geometric relationships between the given images and regularities of the scene. In this thesis, we will illustrate several methods to train an agent to perceive without the use of ground-truth annotations, and show the benefits of leveraging known regularities in our visual world.

1.1 Motivation

A sequence of images contain rich information about the world (scene) around us. Specifically, they inform us about the three-dimensional (3D) geometry (shape) of the environment and our (the agent's) motion within the said environment. To begin, let's consider a sequence of *two* images captured by a monocular (single) camera. Given the camera calibration, one can infer the motion (pose) of the camera up to a scale in a global Euclidean reference frame, provided that there is sufficient parallax between the two frames. Given we have pose from one frame to the other, we can then recover the scene geometry represented as depth or distance from the camera (again, up to a scale) based on the principles of structure from motion (SFM). We do so by finding the set of corresponding pixels (correspondences) between the image pair by projecting the pixels from one image frame to the other based on the predicted depth. The same can be done with a pair of images captured by a stereo rig (two cameras separated by a fixed distance), in which case the relative pose between the camera is known. After rectifying the pair of stereo images, a pair of correspondences will lie along the same scan line; hence, the search for a given pair of correspondences is no longer over the entire image space, but becomes a one dimensional search to find the disparity (displacement) between a pair of pixels along a given scan line – one can then synthesize depth from disparity by applying the focal length and the baseline (distance between the optical center of the cameras). In both cases, once all of the correspondences have been found, we will have effectively recovered the geometry of the scene.

To compute the said correspondences, one may find pairs of pixels that are of similar intensities (minimizing the photometric discrepancy). However, there exists a correspondence ambiguity when matching these pixels. When registering two frames, both are subjected to occlusions and disocclusions (visibility phenomena) where a pixel may appear in one but not the other image. As the 3D reconstruction problem is ill-posed, the solution (the set of correspondences) hinges on the choice of regularization or prior assumptions on the scene (e.g. local connectivity or that surfaces are locally piece-wise smooth). This can be formulated as an additive loss function of two terms: data fidelity and regularization. Data fidelity measures discrepancy between the intensity of two pixels while the regularization (assuming local connectivity) measures the discontinuities between a prediction and its neighbors. A learning framework iteratively finds the correspondences by minimizing the loss function without knowing if a pixel of interest is actually co-visible between the two frames. To further complicate matters, we need to impose regularity. As one do not know the geometry of the scene and hence do not know where to impose regularity, one would commonly apply regularization uniformly across the entire image domain, in hopes that it has been applied to the right place with just the right amount – such is the motivation for our work. In this thesis, we explore various priors (regularization), both generic and learned, and we show the benefits of applying regularization with the right amount, at the right place and at the right time. Our work is realized through four visioning systems that take images (and when available, sparse depth measurements) as input and produce dense depth maps that associates each pixel in the images with a depth value – effectively reconstructing the scene.

1.2 Overview

To evaluate the effectiveness of a prior, we choose the extreme case of 3D reconstruction from a single image, commonly known as single image depth prediction. Due to the loss of one dimension in the projection process, the estimation of the 3D scene geometry from a single image is literally an impossible task as there exists infinitely many 3D scenes that can produce (or is compatible with) the given single image. As mentioned before, to recover a 3D scene (the correspondence problem), we require *multiple* images. Hence, any system that can produce a point estimate of the 3D scene *must* rely heavily on priors. We will, from this point onward, refer to the resulting point estimate as a *hypothesis*, or a *prediction.* We will first illustrate two methods to exploit known regularities based on the data fidelity residual (Chapter 2) and the natural pose and orientation of objects (Chapter 3).

Single image depth prediction is a great problem setting for demonstrating the influence of a regularizer. However, the use case of such is limited to applications such as 2D to 3D "pop-up" and computational photography. To extend our work to deployable real-world applications, we consider the depth completion problem, where in addition to a single image, we are also given sparse depth measurements produced by a lidar (outdoor driving scenarios), or a simultaneous localization and mapping (SLAM) or visual inertial odometry [JS11] (VIO) system (indoor scenarios). A lidar measures distance to a target by projecting a laser and measuring the reflected light with a sensor. The sparse depth measurements produced by a lidar generally correspond to horizontal scan lines along the image plane. The density of such depends on the number of lines produced by the lidar and can cover approximately 5% of the image domain concentrated on the lower third of the image. SLAM and VIO systems localize themselves by tracking a number of visually discriminative Lambertian regions (e.g. corners and edges). The position of such regions defines the Euclidean reference frame, with respect to which motion of the system is estimated. As these regions are visually distinctive they tend to be sparse (typically in the order of hundreds to thousands), but are sufficient to support a point-estimate of motion. In both cases, the sparse depth measurements are a poor representation of shape as they do not reveal the topology of the scene. The missing points between the sparse depth measurements could be empty, or occupied by a surface. The given sparse depth measurements and image can then be combined to restrict the possible scenes compatible with the input. To this end, we will demonstrate the effects of a learned prior (obtained by training a system to predict the compatibility of a given scene) in Chapter 4 and propose a novel refinement technique in Chapter 5 that takes a coarse approximation of a scene produced by a prior and outputs a detailed depth map.

1.3 Organization of Thesis

In Chapter 2 we will study the change in residuals over training time in relation to the amount of regularization applied. We begin with a commonly used generic prior, local smoothness, as our regularizer. Local smoothness discourages perturbations in the local neighborhood of the predictions. When applied uniformly to the solution (image) domain, local smoothness prevents large discontinuities, which generally occurs along object boundaries in the 3D reconstruction problem. Moreover, local smoothness does not consider the correctness of predictions and hence may propagate incorrect predictions to neighbors. We devised a spatially (image domain) and temporally (training time) varying weighting scheme that considers the data fidelity residual as a signal for adapting the degree of regularization based on the fitness of model to data. Initially, our weighting scheme imposes very little regularization, allowing the data fidelity term to dominate and explore the solution space. As the model learns to predict depth correctly, we begin to increase the amount of regularization. We show, that without any additional trainable parameters, our approach is able to achieve state-of-the-art in single image (monocular) depth prediction.

In Chapter 3 we will study the effect of a gravity-induced pose prior on selective objects in the scene. In this work, we use a ubiquitous setup of a single camera with an inertial measurement unit (IMU). Given that the poses of many objects in the scene are influenced by gravity (e.g. roads are perpendicular to the direction of gravity, and buildings are built parallel to gravity), we use this prior to selectively impose a simple shape prior (vertical and horizontal planes) on the objects in the scene. To achieve this, we obtain the direction of gravity from the IMU. However, as mentioned, not all objects are affected by gravity the same way (e.g. pose may be parallel or perpendicular to gravity) and the pose of some (generally deformable objects) are not affected. Hence, we need the semantics of the scene (produced by a semantic segmentation network) to selectively apply the prior. Our shape model is simple, but effective – we encourage the surfaces of objects to follow a horizontal or a vertical plane at training time to bias the predictions. At testing, we directly predict depth and remove the semantic segmentation network. We validate our approach on several recent monocular depth prediction methods and show that our prior can consistently boost worse performing methods above the state-of-the-art. When apply to the state-of-the-art, we further improve their performance to achieve new state-of-the-art.

In Chapter 4, we consider the depth completion problem. Although sparse depth (where available) restricts the possible scenes that can produce the image, the areas of the image where sparse depth is not present is still compatible with with infinitely many scenes. Rather than using a generic prior such as local smoothness, we explore the use of a learned prior by training a conditional prior network [YS18a] to train a novel depth completion architecture that we proposed. The conditional prior network takes an image and a degraded depth map as input and produces a depth map compatible with the input. As the early steps of training generally produces incorrect predictions (degraded depth map), the conditional prior network guides the predictions towards a more compatible one. The result of which is the state-of-the-art in depth completion. Yet, because we require a conditional prior network (essentially the size of a depth completion network), training requires large amounts of computational resources. Although at test time we can forgo the conditional prior network, the depth completion network itself is still very large and deep to compensate for the sparse inputs. These drawbacks motivates us to pursue a different direction.

In Chapter 5, we re-approach the depth completion problem with usability and computation in mind. Given that the use case for depth completion includes densification of sparse points produced by SLAM and VIO system (applicable to embedded systems such as drones and robots), we need to be mindful of the energy and memory usage of our model. A standard Nvidia Jetson only has 8 gigabytes of memory with much less computational power than a standard desktop graphical processing unit (GPU) such as a GTX 1080Ti. To begin, the challenge of sparse depth completion is precisely the sparsity. Much of the literature is dedicated to producing novel architecture (generally deeper and larger networks or specialized network operations) to deal with this problem. Hence, to alleviate the burden of processing sparse depth, we propose to approximate the scene by exploiting local connectivity to compute a Delaunay Triangulation. The missing values are linearly interpolated within the Barycentric coordinates, effectively propagating the sparse depth values. To refine the approximation, we propose a light-weight depth completion network that constitutes less than 50% of the parameters used by our network in Chapter 4. Although we have reduced the number of parameters, we received a large performance gain in both speed and accuracy of predictions. This is mainly due to our two-stage approach of approximate and refine. With the size reduction and computational improvements, our model fits comfortably in a Jetson chipset, enabling it to be deployed in real-world applications. Our model currently holds the state-of-the-art in unsupervised depth completion.

Finally, we will discuss some limitations to each of the proposed systems and possible improvements for future work in Chapter 6. Furthermore, as we will see in Chapter 5, the fusion of classical and deep learning methods is able to produce strong results in depth completion. We believe this can extend to other applications. we make a few remarks regarding this possible direction to conclude the thesis.

CHAPTER 2

Bilateral Cyclic Constraint and Adaptive Regularization for Unsupervised Monocular Depth Prediction

2.1 Introduction

Estimating the 3-dimensional geometry of a scene is a fundamental problem in machine perception with a wide range of applications, including autonomous driving [JGB17], robotics [LLS15, SSP10], pose-estimation [SSK13], localization [HS19], and scene object composition [HHY19, KSH14]. It is well-known that 3-d scene geometry can be recovered from multiple images of a scene taken from different viewpoints, including stereo, under suitable conditions. Under no conditions, however, is a single image sufficient to recover 3-d scene structure, unless prior knowledge is available on the shape of objects populating the scene. Even in such cases, metric information is lost in the projection, so at best we can use a single image to generate hypotheses, as opposed to estimates, of scene geometry.

Recent works [CFY16, EPF14, LRB16, LSL15, LSL16, XRO17, XWT18] sought to exploit such strong scene priors by using pixel-level depth annotation captured with a range sensor (e.g. depth camera, lidar) to regress depth from the RGB image. Cognizant of the intrinsic limitations of this endeavor, we exploit stereo imagery to train a network without ground-truth supervision for generating depth hypotheses, to be used as a reference for 3-d reconstruction. We evaluate our method against ground-truth depths via two benchmarks from the KITTI dataset [GLU12] and show that it generalizes well by applying models trained on KITTI to Make3d [SSN09].

Rather than attempting to learn a prior by associating the raw-pixel values with depth, we recast depth estimation as an image reconstruction problem [GBC16, GMB17] and exploit the epipolar geometry between images in a rectified stereo pair to train a deep fully convolutional network. Our network learns to predict the dense pixel correspondences (disparity field) between the stereo pair, despite only having seen one of them. Hence, our network implicitly learns the relative pose of the cameras used in training and hallucinates the existence of a second image taken from the same relative pose when given a single image during testing. From the disparity predictions, we can synthesize depth using the known focal length and baseline of the cameras used in training.

While [GBC16, GMB17, XGF16] follow a similar training scheme, [XGF16] does not scale to high resolution, and [GBC16] uses a non-differentiable objectives. [GMB17] proposed using two uni-directional edge-aware disparity gradients and left-right disparity consistency as regularizers. However, edge-awareness should inform bidirectionally and left-right consistency suffers from occlusions and dis-occlusions. Moreover, regularity should not only be data-driven, but also model-driven.

Our contributions are three-fold: (i) A model-driven adaptive weighting scheme that is both space- and training-time varying and can be applied generically to regularizers. (ii) A bilateral consistency constraint that enforces the cyclic application of left and right disparity to be the identity. (iii) A two-branch decoder that specifically learns the features necessary to maximize data fidelity and utilizes such features to refine an initial prediction by enforcing regularity. We formulate our contributions as an objective function that, when realized even by a generic encoder-decoder, achieves state-of-the-art performance on two KITTI [GLU12] benchmarks and exhibits generalizability to Make3d [SSN09].

2.2 Related Works

Supervised Monocular Depth Estimation. [SCN06] proposed a patch-based model that combined local estimates with Markov random fields (MRF) to obtain the global depth. Similarly, [HEH07, KLK12, KWI13, SSN09] exploited local monocular features to make global predictions. However, local methods lack the global context needed to generate accurate depth estimates. [LSL16] instead employed a convolutional neural network (CNN). [LSP14] further improved monocular methods by incorporating semantic cues into their model.

[EF15, EPF14] introduced a two scale network. [LRB16] proposed a residual network with up-sampling modules to produce higher resolution depth maps. [CFY16] learned depth using crowd-sourced annotations and [FGW18] learned the ordinal relations using atrous spatial pyramid pooling. [RT16] used image patches with neural forests. [KPS16, XRO17, XWT18] used conditional random fields (CRF) jointly with a CNN.

Unsupervised Monocular Depth Estimation. Recently, [FNP16] introduced novel view synthesis by predicting pixel values based on interpolation from nearby images. [XGF16] minimized an image reconstruction loss to hallucinate the existence of a right view of a stereo pair given the left by producing the distribution of disparities for each pixel.

[GBC16] trained a network for monocular depth prediction by reconstructing the right image of a stereo pair with the left and synthesizing disparity as an intermediate step. Yet, their image formation model is not fully differentiable, making their objective function difficult to optimize. Unsupervised methods [GMB17, PHC15, ZKA16, ZTS16] utilized a bilinear sampler modeled after the Spatial Transformer Network [JSZ15] to allow for a fully differentiable loss and end-to-end training of their respective networks. Specifically, [GMB17] used SSIM [WBS04] as a loss in addition to the image reconstruction loss. Also, [GMB17] predicted both left and right disparities and used them for regularization via a left-right consistency check along with an edge-aware smoothness term. [ATP18] trains a Generative Adversarial Network (GAN) [GPM14] to constrain the output to reconstruct a realistic image to reduce the artifacts seen from stereo reconstruction. This class of method is also employed in depth completion [YWS19].

Self-supervised methods [MWA18, UZU17, ZBS17, ZLH18] used a pose network to learn ego-motion and depth from monocular videos, while [WBZ18a, YWS18] leveraged visual odometry from off-the-shelf methods [EKC18, SSC11] and [FWS19] gravity as supervisors. [ZGW18] followed both unsupervised and self-supervised paradigms by using stereo video streams and proposed a feature reconstruction loss. While additional supervision and data are used to improve predictions, [GMB17] still remains as the state-of-the-art in the unsupervised setting. Our method follows the unsupervised paradigm and we show that it not only outperforms [GMB17], but also [ZGW18] who leveraged techniques from both unsupervised and self-supervised domains.

Adaptive Regularization. A number of computer vision problems can be formulated as energy minimization in a variational framework with a data fidelity term and a regularizer weighted by a fixed scalar. The solution found by the minimal energy involves a trade-off between data fidelity and regularization. Finding the optimal parameter for regularity is a long studied problem as [GK92] explored methods to determine the regularization parameter in image de-noising, while [NMG01] used cross-validation as a selection criterion for the weight. [GMB17, WCP09, WTP09] used image gradients as cues for a data-driven weighting scheme. [YS18a] learned regularity conditioned on an image. Recently, [HKB17, HKD17] proposed that regularity should not only be data-driven, but also model driven. The amount of regularity imposed should adapt to the fitness of the model in relation to the data rather than being constant throughout the training process.

We propose a novel objective function using bilateral cyclic consistency constraint along with a spatial and temporal varying regularization modulator. We show that despite using the fewer parameters than [GMB17], we outperform [GMB17] and other unsupervised methods. We detail our loss function with adaptive regularization, in Sec. 2.3, present a two-branch decoder architecture in Sec. 2.4, and specify hyper-parameters and data augmentation procedures used in Sec. 2.5. We evaluate our model on the KITTI 2015, KITTI Eigen Split, and Make3d benchmarks in Sec. 2.6. Lastly, we end with a discussion of our work in Sec. 2.7.

2.3 Method Formulation

We learn a model to hypothesize or "estimate" the disparity field d compatible with an image I^0 by exploiting the availability of stereo pairs (I^0, I^1) during training. We then synthesize the depth $z = FB/d$ of the scene using the focal length F and baseline B during test time. Given I^0, we estimate a function $d \in \mathbb{R}_+$ that represents the disparity of I^0, which we formulate as a loss function L (Eqn. 2.1), comprised of data terms and adaptive regularizers.

Our network, parameterized by ω, takes a single image I^0 as input and estimates a function $d = f(I^0; \omega)$, where d represents the disparity (which is monotonically related to inverse-depth) corresponding to I^0. We drive the training process with I^1, which is only used in the loss function, by a surrogate loss that minimizes the reprojection error of I^0 to I^1 and vice versa. We will refer to the disparity estimated by L as d^0 and d^1 for I^0 and I^1, respectively. Interested readers may refer to Supplementary Materials (Supp. Mat.) for more details on our formulation.

$$L = \underbrace{w_{ph} l_{ph} + w_{st} l_{st}}_{\text{data fidelity}} + \underbrace{w_{sm} l_{sm} + w_{bc} l_{bc}}_{\text{regularization}} \tag{2.1}$$

where each individual term l will be described in the next sections and their weights w in Sec. 2.5.

2.3.1 Data Fidelity

Our data fidelity terms seek to minimize the discrepancy between the observed stereo pair (I^0, I^1) and their reconstructions $(\hat{I}^0, \hat{I}^1)$. We generate each $\hat{I}$ term by applying a 1-d horizontal disparity shift to I at each position (x, y):

$$\hat{I}^0{}_{xy} = I^1_{xy-d^0_{xy}} \text{ and } \hat{I}^1{}_{xy} = I^0_{xy+d^1_{xy}} \tag{2.2}$$

We do so by using a 1-d horizontal bilinear sampler modeled after the image sampler from the Spatial Transformer Network [JSZ15] – instead of applying an affine transformation to activations, we warp an image to the domain of its stereo-counterpart using disparities. Our sampler is locally fully differentiable and each output pixel is the weighted sum of two (left and right) pixels. We propose to minimize the reprojection residuals as a two-part loss, which measures the standard color constancy (photometric) and the difference in illumination, contrast and image quality (structural).

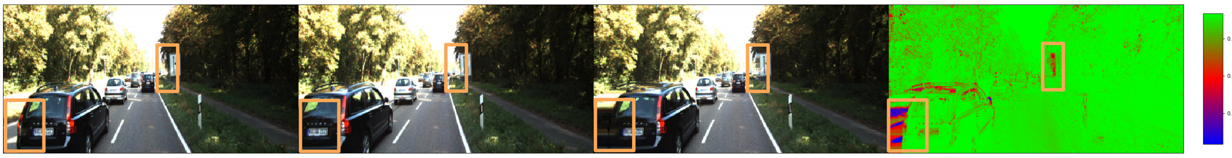

Figure 2.1: *Examples of our adaptive weighting scheme as images.* Left to right: left image, right image, left reconstruction, adaptive weights. The adaptive weights reduce regularization at regions of high residual; hence, they discount dis-occlusions and occlusions as in the highlighted regions.

Photometric loss. We model the image formation process via a photometric loss l_{ph}, which measures the $L1$ penalty of the reprojection residual for each I and $\hat{I}$ on each channel at every (x, y) position in the image space Ω:

$$l_{ph} = \sum_{(x,y)\in\Omega} |I^0_{xy} - \hat{I}^0_{xy}| + |I^1_{xy} - \hat{I}^1_{xy}| \tag{2.3}$$

Structural loss. In order to make inference invariant to local illumination changes, we use a perceptual metric (SSIM) that discounts such variability. We apply SSIM (ϕ) to image patches of size 3×3 at corresponding (x, y) in I and $\hat{I}$. Since two similar images give a SSIM score close to 1, we subtract 1 by the score to represent a distance:

$$l_{st} = \sum_{(x,y)\in\Omega} 2 - (\phi(I^0_{xy}, \hat{I}^0_{xy}) + \phi(I^1_{xy}, \hat{I}^1_{xy})) \tag{2.4}$$

2.3.2 Residual-Based Adaptive Weighting Scheme

A point estimate d can be obtained by maximizing the Bayesian criterion with a data fidelity term (energy) $\mathcal{D}(d)$ and a Bayesian or Tikhonov regularizer $\mathcal{R}(d)$ in the form:

$$\mathcal{D}(d) + \alpha\mathcal{R}(d) \tag{2.5}$$

where the weight α is a pre-defined positive scalar parameter that controls the regularity to impose on the model, leading to a trade-off between data fidelity and regularization.

The weight α modulates between data-fidelity and regularization, constraining the solution space. Yet, subjecting the entire solution, a dense disparity field, to the same regularity fails to address cases where the assumptions do not hold. Suppose one enforces a smoothness constraint to the output disparity field by simply taking the disparity gradient ∇d. This constraint would incorrectly penalize object boundaries (regions of high image gradients) and hence [GMB17, HKJ13] apply an edge-aware term to reduce the effects of regularization on edge regions. Although the edge-awareness term gives a data-driven approach on regularization, it is still static (the same image will always have the same weights) and independent of the performance of the model. Instead, we propose a space- and training-time varying weighting scheme based on the performance of our model measured by reprojection residuals.

Model-driven adaptive weight. We propose an adaptive weight α_{xy} that varies in space and training time for every position (x, y) of the solution based on the local residual $\rho_{xy} = |I_{xy} - \hat{I}_{xy}|$ and the global residual, represented by the average per-pixel residual, $\sigma = \dfrac{1}{\frac{1}{|\Omega|} \sum\limits_{(x,y)\in\Omega} |I_{xy} - \hat{I}_{xy}|}$:

$$\alpha_{xy} = \exp\left(-\frac{c\rho_{xy}}{\sigma}\right) \tag{2.6}$$

α is controlled by the local residual between an image I and its reprojection $\hat{I}$ at each position while taking into account of the global residual σ, which correlates to the training time step and decreases over time. c is a scale factor for the range of α. α is naturally small when residuals are large and tends to 1 as training converges.

Local adaptation. Consider a pair of poorly matched pixels, $(I_{xy}, \hat{I}_{xy})$, where the residual $|I_{xy} - \hat{I}_{xy}|$ is large. By reducing the regularity on the solution d_{xy}, we effectively allow for exploration in the solution space to find a better match and hence a d_{xy} that minimizes the data fidelity terms. Alternatively, consider a pair of perfectly matched pixels, $(I_{xy}, \hat{I}_{xy})$, where $|I_{xy} - \hat{I}_{xy}| = 0$. We should apply regularization to decrease the scope of the solution space such that we can allow for convergence and propagate the solution. Hence, a spatially adaptive α_{xy} must vary inversely to the local residual ρ_{xy} such that we impose regularization when the residual is small and reduce it when the residual is large.

Global adaptation. Consider a solution d_{xy} proposed at the first training time step $t = 1$. Imposing regularity effectively reduces the solution space based on an assumption about d_{xy} and biases the final solution. We propose that a weighting scheme $\alpha_{xy} \rightarrow 1$ as $t \rightarrow \infty$. However, if α_{xy} is directly dependent on the t, then α_{xy} will change if we continue to train even after convergence – causing the model to be unstable. Instead, let α_{xy} be inversely proportional to the global residual σ such that α_{xy} is small when the σ is large (generally corresponding to early time steps) and $\alpha_{xy} \rightarrow 1$ as $\sigma \rightarrow 0$. When training converges (i.e. the global residual has stabilized), α_{xy} likewise will be stable. This naturally lends to an annealing schedule where $\alpha_{xy} \rightarrow 1$ as time progresses in training steps.

2.3.3 Adaptive Regularization

Our regularizers assume local smoothness and consistency between the left and right disparities estimated. We propose to minimize the disparity gradient (smoothness) and the disparity reprojection error (bilateral cyclic consistency) while adaptively weighting both with α (Sec. 2.3.2).

Smoothness loss. We encourage the predicted disparities to be locally smooth by applying an $L1$ penalty to the disparity gradients in the x (∂_X) and y (∂_Y) directions. However, such an assumption does not hold at object boundaries, which generally correspond to regions of high changes in pixel intensities; hence, we include an edge-aware term λ to allow for discontinuities in the disparity gradient. We also weigh this term adaptively with α:

$$\begin{aligned} l_{sm} = \sum_{(x,y)\in\Omega} & \alpha^0_{xy}(\lambda^0_{xy}|\partial_X d^0_{xy}| + \lambda^0_{xy}|\partial_Y d^0_{xy}|) + \\ & \alpha^1_{xy}(\lambda^1_{xy}|\partial_X d^1_{xy}| + \lambda^1_{xy}|\partial_Y d^1_{xy}|) \end{aligned} \tag{2.7}$$

where $\lambda_{xy} = e^{-|\nabla^2 I_{xy}|}$ and the ∇^2 operator denotes the image Laplacian. We use the image Laplacian over the first order image gradients because it allows the disparity gradients to be aware of intensity changes in both directions. However, we regularize the disparity field using the disparity gradient so that we can allow for independent movement in each direction. Prior to computing the image Laplacian for λ, we smooth the image with a Gaussian kernel to reduce noise.

Bilateral cyclic consistency loss. A common regularization technique in stereo-vision is to maintain the consistency between the left (d^0) and right (d^1) disparities by reconstructing each disparity through projecting its counter-part with its disparity shifts:

$$d^{0p}_{xy} = d^1_{xy-d^0_{xy}} \text{ and } d^{1p}_{xy} = d^0_{xy+d^1_{xy}} \tag{2.8}$$

However, in doing so, the projected disparities suffer from the unresolved correspondences of both the disparity ramps, occlusions and dis-occlusions. We, propose a bilateral cyclic consistency check that is designed to specifically reason about occlusions while removing the effects of stereo dis-occlusions. We follow the intuition that the disparities d should have an identity mapping when projected to the domain of its stereo-counterpart and back-projected to the original domain as a reconstruction $\hat{d}$ so reconstruction of dis-occlusion is ignored.

$$\hat{d}^0_{xy} = d^0_{xy+d^1_{xy}-d^0_{xy}} \text{ and } \hat{d}^1_{xy} = d^1_{xy-d^0_{xy}+d^1_{xy}} \tag{2.9}$$

By applying an $L1$ penalty on the disparity field and its reconstruction, we are constraining that the cyclic transformations should be the identity transform, which keeps d^0 and d^1 consistent with each other in co-visible regions. If there exists an occluded region, the region in the reconstruction would be inconsistent with the original – yielding reprojection error. To avoid penalizing a model for an unresolvable correspondence due to the nature of the data, we propose to adaptively regularize the bilateral cyclic constraint using our residual-based weighting scheme (Eqn. 2.6). Unsurprisingly, local regions of high reprojection residual often correspond to occluded regions.

$$l_{bc} = \sum_{(x,y)\in\Omega} \alpha^0_{xy}|d^0_{xy} - \hat{d}^0_{xy}| + \alpha^1_{xy}|d^1_{xy} - \hat{d}^1_{xy}| \tag{2.10}$$

2.4 A Two-Branch Decoder

As our adaptive weighting scheme (Sec. 2.3.2) is function of the data fidelity residuals, we seek to ensure that the network learns a sufficient representation to minimize the data fidelity loss (Sec. 2.3.1). We propose a two-branch decoder (Fig. 2.2) with one branch (prefixed with '`i`') dedicated to learning the features, `iconv`, necessary to make a prediction that minimizes

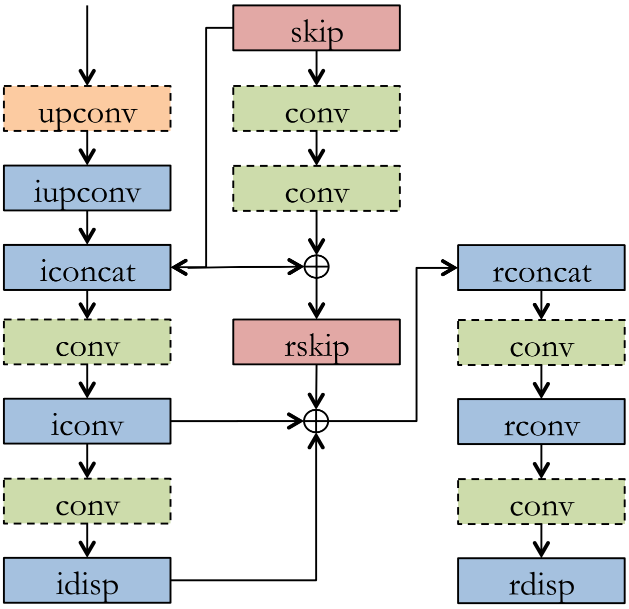

Figure 2.2: *Two-branch decoder.* `idisp` produces an initial prediction based only on the data terms and `rdisp` produces a refined prediction using the entire loss function (Eqn. 2.1). By minimizing just the data terms (Eqn. 2.11) in `idisp`, we force `iconv` to learn sufficient information for the reconstruction task such that `rdisp` can utilize such features along with the residual learned from the skip connection to refine a prediction that satisfies data fidelity by imposing regularity based on the data fidelity residual.

data fidelity loss:

$$L^0 = w_{ph} l_{ph} + w_{st} l_{st} \tag{2.11}$$

using the reconstructed features via up-convolution and the corresponding `skip` connection from the encoder. We use a residual block [HZR16] to learn the skip connection residual, `rskip`, necessary to minimize Eqn. 2.1 – both data fidelity and regularity loss. By concatenating `iconv` and `rskip` with the initial prediction (`idisp`) as features for the second branch (prefixed with ‘`r`‘), we have provided the decoder branch with a prediction that satisfies data fidelity along with features necessary to impose regularity. The branch can now utilize such information to refine the initial prediction by adaptively applying regularization based on the data fidelity residual. To maintain a similar network size and run-time, we reduce the depth of the network by 1 and added a single convolution as the first layer to enable a skip connection to the last layer. This, in fact, resulted in our network having ≈ 10 million fewer parameters than [GMB17]. We show qualitative results in Fig. 2.3 and 2.4 where we observe the benefits of learning the features that satisfy data fidelity as we recover more details about the scene geometry. Quantitatively, we show in Table 2.3, Table 2.2, and 2.4 that this structure improves over the state-of-the-art performance on all metrics achieved by our generic encoder with a single branch decoder, where the final predictions of both decoders minimize our objective function (Eqn. 2.1).

2.5 Implementation Details

Our approach was implemented using TensorFlow [ABC16]. There are ≈ 31 million trainable parameters in the generic encoder-decoder [GMB17] and ≈ 21 million in our proposed structure (more details can be found in Supp. Mat. Table 2 and 3). Training takes ≈ 18 hours using an Nvidia GTX 1080Ti. Inference takes ≈ 32 ms per image. We used Adam [KB14] to optimize our network with a base learning rate of 1.8×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We then increase the learning rate to 2×10^{-4} after 1 epoch, decrease it by half after 46 epochs and by a quarter after 48 epochs for a total of 50 epochs. We use a batch size of 8 with a 512×256 resolution and 4 levels in our loss pyramid. We are able to achieve our results using the following set of weights for each term in our loss function: $w_{ph} = 0.15$, $w_{st} = 0.425$, $w_{sm} = 0.10$ and $w_{bc} = 1.05$. We choose the scale factor $c = 5.0$ for the adaptive weight α. For our smoothness term, we decrease it by a factor of 2^r for each r-th resolution in the loss pyramid where $r = 0$ refers to our highest resolution at 512×256 and $r = 3$ the lowest. Data augmentation is performed online during training. We perform a horizontal flip (with a swap to maintain correct relative positions) on the stereo pairs with 50% probability. Color augmentations on brightness, gamma and color shifts of each channel also occur with 50% chance. We uniformly sample from $[0.5, 1.5]$ for brightness, and $[0.8, 1.2]$ for gamma and each color channel separately.

Metric	Definition
AbsRel	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\frac{\lvert z_{xy}-z_{xy}^{\text{gt}}\rvert}{z_{xy}^{\text{gt}}}$
SqRel	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\frac{\lvert z_{xy}-z_{xy}^{\text{gt}}\rvert^2}{z_{xy}^{\text{gt}}}$
RMS	$\sqrt{\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert z_{xy}-z_{xy}^{\text{gt}}\rvert^2}$
logRMS	$\sqrt{\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert\log z_{xy}-\log z_{xy}^{\text{gt}}\rvert^2}$
$\log_{10}$	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert\log z_{xy}-\log z_{xy}^{\text{gt}}\rvert$
Accuracy	% of z_{xy} s.t. $\delta \doteq \max\left(\frac{z_{xy}}{z_{xy}^{\text{gt}}},\frac{z_{xy}^{\text{gt}}}{z_{xy}}\right) <$ threshold

Table 2.1: *Error and accuracy metrics.* z_{xy} is the predicted depth at $(x, y) \in \Omega$ and z_{xy}^{gt} is the corresponding ground truth. Three different thresholds $(1.25, 1.25^2$ and $1.25^3)$ are used in the accuracy metric as a convention in the literature.

2.6 Experiments and Results

We present our results on the KITTI dataset [GLU12] under two different training and testing schemes, the KITTI 2015 split [GMB17] and the KITTI Eigen split [EPF14, GBC16]. The KITTI dataset contains 42,382 rectified stereo pairs from 61 scenes with approximate resolutions of 1242 × 375. We evaluate our method on the monocular depth estimation task on KITTI Eigen split and compare our approach with similar variants on a disparity error metric as an ablation study using the KITTI 2015 split. We show that our method outperforms state-of-the-art unsupervised monocular approaches and even supervised approaches on KITTI benchmarks, while generalizing to Make3d [SSN09].

2.6.1 KITTI Eigen Split

We evaluate our method using the KITTI Eigen split [EPF14], which has 697 test images from 29 scenes. The remaining 32 scenes contain 23,488 stereo pairs, of which 22,600 pairs are used for training and the rest for validation, following [GBC16]. We project the velodyne

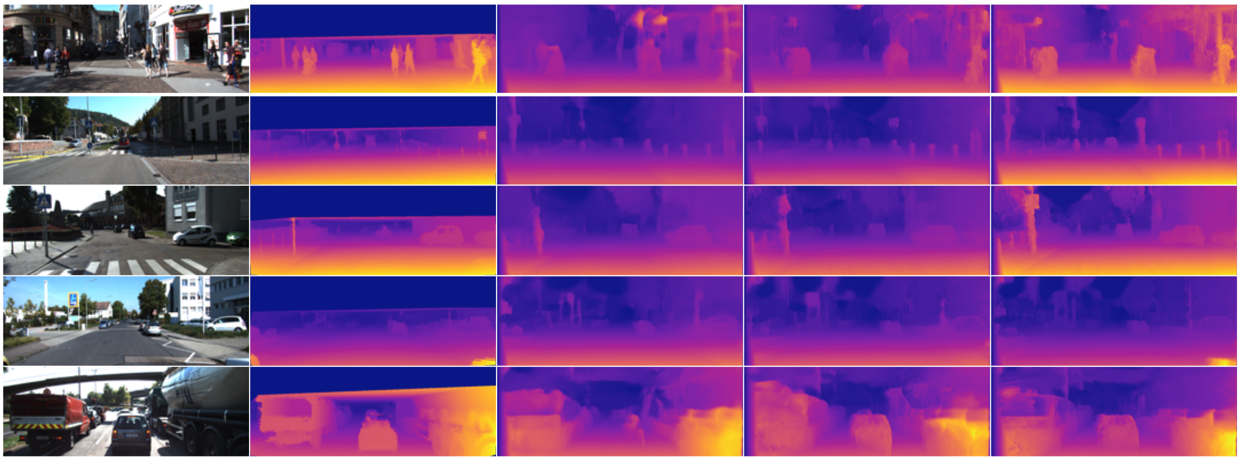

Figure 2.3: *Qualitative results on KITTI Eigen split.* From left to right: input images, ground-truth disparities, results of Godard et al. [GMB17], our results with a generic decoder and our results with the proposed decoder. Our method under both decoders recovers more scene structures (row 2, 3: street signs, row 5: car in middle). Moreover, the predictions of the proposed two-branch structure are more realistic (row 1: pedestrian on right, row 4: tail of another car at bottom right corner, row 5: hollow trunk of truck on left, where both [GMB17] and the generic decoder predicted as a surface).

points into the left input color camera frame to generate ground-truth depths. The ground-truth depth maps are sparse ($\approx 5\%$ of the entire image) and prone to errors from rotation of the velodyne and motion of the vehicle and surrounding objects along with occlusions. As a result, we use the cropping scheme proposed by [GBC16], which contains approximately 58% in height and 93% in width of the image dimensions.

We compare our approach with the recent monocular depth estimation methods at 50 and 80 meters caps in Table 2.3 and Table 2.2. Fig. 2.3 provides a qualitative comparison between our method and the baseline. We note that [ZGW18] trained two networks using stereo video streams (as opposed to a single network with stereo pairs like ours and [GMB17]), which allows their networks to learn a depth prior in both spatial and temporal domains. Using the network of [GMB17] (generic encoder with a single branch decoder), we outperforms all competing methods in all metrics under both depth caps except for $\delta < 1.25^3$ where we are comparable to [ZGW18]. We improve consistently over [GMB17] and [ZGW18] by an

		Error Metrics				Accuracy Metrics		
Method	Dataset	Abs Rel	Sq Rel	RMS	logRMS	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou et al. [ZBS17]	K	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Mahjourian et al. [MWA18]	K	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Garg et al. [GBC16]	K	0.152	1.226	5.849	0.246	0.784	0.921	0.967
Godard et al. [GMB17]	K	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Zhan et al. [ZGW18] (w/ video)	K	0.144	1.391	5.869	0.241	0.803	0.928	**0.969**
Ours (Full Model)	K	0.135	1.157	5.556	0.234	0.820	0.932	0.968
Ours (Full Model)*	K	**0.133**	**1.126**	**5.515**	**0.231**	**0.826**	**0.934**	**0.969**
Zhou et al. [ZBS17]	CS+K	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian et al. [MWA18]	CS+K	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Godard et al. [GMB17]	CS+K	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Ours (Full Model)*	CS+K	**0.118**	**0.996**	**5.134**	**0.215**	**0.849**	**0.945**	**0.975**

Table 2.2: *Quantitative results*[2.1] *on the KITTI [GLU12] Eigen split [EPF14] benchmark.* Depths are capped at 80 meters. K denotes training on KITTI. CS+K denotes pretraining on Cityscape [COR16] and fine-tuning on KITTI. Our full model using a generic encoder-decoder consistently outperforms other methods across all metrics with the exception of $\delta < 1.25^3$ where [ZGW18], which used temporal information (sequences of stereo-pairs), marginally beats our us by 0.1%. Our proposed decoder (*) improves over our encoder-decoder model across all metrics and is the state-of-the-art.

average of 8.7% and 5.75% in AbsRel, 13.1% and 10.5% in SqRel and even 5.25% and 2.55% in logRMS, respectively. Furthermore, we score significantly higher in $\delta < 1.25$ (the hardest accuracy metric), which suggests that our model produces more correct and realistically detailed depths than all competing methods. In addition, our two-branch decoder improves over the said results across all metrics and depth caps and is the current state-of-the-art. Table 2.2 shows that our model also beats [GMB17] when pretraining on Cityscape [COR16] and fine-tuning on KITTI. An ablation study on Eigen Split examining the effects of each of our contributions (Sec. 2.3.3) can be found in our Supp. Mat.

		Error Metrics				Accuracy Metrics		
Method	Dataset	Abs Rel	Sq Rel	RMS	logRMS	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou et al. [ZBS17]	K	0.201	1.391	5.181	0.264	0.696	0.900	0.966
Garg et al. [GBC16]	K	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard et al. [GMB17]	K	0.140	0.976	4.471	0.232	0.818	0.931	0.969
Zhan et al. [ZGW18] (w/ video)	K	0.135	0.905	4.366	0.225	0.818	0.937	**0.973**
Ours (Full Model)	K	0.128	0.856	4.201	0.220	0.835	0.939	0.972
Ours (Full Model)*	K	**0.126**	**0.832**	**4.172**	**0.217**	**0.840**	**0.941**	**0.973**

Table 2.3: *Quantitative results*[2.1] *on the KITTI [GLU12] Eigen split [EPF14] benchmark.* Depths are capped at 50 meters. K denotes training on KITTI. Our full model using a generic encoder-decoder consistently outperforms other methods, including [ZGW18] who trained on sequences of stereo-pairs, across all metrics. Our proposed decoder (*) improves over our encoder-decoder model is the state-of-the-art.

2.6.2 KITTI 2015 Split

We evaluate our method on 200 high quality disparity maps provided as part of the official KITTI training set [GLU12]. These 200 stereo pairs cover 28 of the total 61 scenes. From 30,159 stereo pairs covering the remaining 33 scenes, we choose 29,000 for training and the rest for validation. While typical training and evaluation schemes project velodyne laser values to depth, we choose to use the provided disparity maps as they are less erroneous than velodyne data points. In addition, we also use the official KITTI disparity metric of end-point-error (D1-all) to measure our performance as it is a more appropriate metric on our class of approach that outputs disparity and synthesizes depth from the output using camera focal length and baseline.

We show qualitative comparisons in Fig. 2.4 and quantitative comparisons in Table 2.4. Table 2.4 also serves as an ablation study on variants belonging to the stereo unsupervised paradigm using different image formation model and regularization terms. We show that by simply applying our adaptive regularization to [GMB17], we achieve improvement over their model. We also study the effects of substituting our bilateral cyclic consistency with

	Error Metrics					Accuracy Metrics		
Method	Abs Rel	Sq Rel	RMS	logRMS	D1-all	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
[GMB17] w/ Deep3D [XGF16]	0.412	16.37	13.693	0.512	66.850	0.690	0.833	0.891
[GMB17] w/ Deep3Ds [XGF16]	0.151	1.312	6.344	0.239	59.640	0.781	0.931	0.976
$ph + st + \lambda^G sm$ [GMB17]	0.123	1.417	6.315	0.220	30.318	0.841	0.937	0.973
$ph + st + \lambda^G sm + lr$ [GMB17]	0.124	1.388	6.125	0.217	30.272	0.841	0.936	0.975
$ph + st + \alpha\lambda^G sm + \alpha lr$	0.120	1.367	6.013	0.211	30.132	0.849	0.942	0.975
Aleotti et al. [ATP18]	0.119	1.239	5.998	0.212	29.864	0.846	0.940	0.976
$ph + st + \lambda^L sm + bc$	0.117	1.264	5.874	0.207	29.793	0.851	0.944	0.977
$ph + st + \alpha\lambda^L sm + \alpha lr$	0.117	1.251	5.876	0.206	29.536	0.851	0.944	0.977
$ph + st + \alpha\lambda^G sm + \alpha bc$	0.115	1.211	5.743	0.203	28.942	0.852	0.945	0.977
$ph + st + \alpha\lambda^L sm + \alpha bc$	0.114	1.172	5.651	0.202	28.142	0.855	**0.947**	0.979
$ph + st + \alpha\lambda^L sm + \alpha bc$ *	**0.110**	**1.119**	**5.576**	**0.200**	**27.149**	**0.856**	**0.947**	**0.980**

Table 2.4: *Quantitative comparison*[2.1] *amongst variants of our model on KITTI 2015 split proposed by [GMB17].* Each variant is named according to its loss function. *ph* and *st* denote data terms, *sm* local smoothness, α our adaptive weights, λ^G image gradients [GMB17], λ^L image Laplacian, *lr* left-right consistency [GMB17], and *bc* our bilateral cyclic consistency. We show the effectiveness of our adaptive regularization (Sec. 2.3.3) by applying it to [GMB17] and improving their model. Our full model using a generic encoder-decoder outperforms all variants on every metric, including [ATP18] which predicts disparities that generate photo-realistic images. Our full model using our proposed two-branch decoder (*) further improves the state-of-the-art.

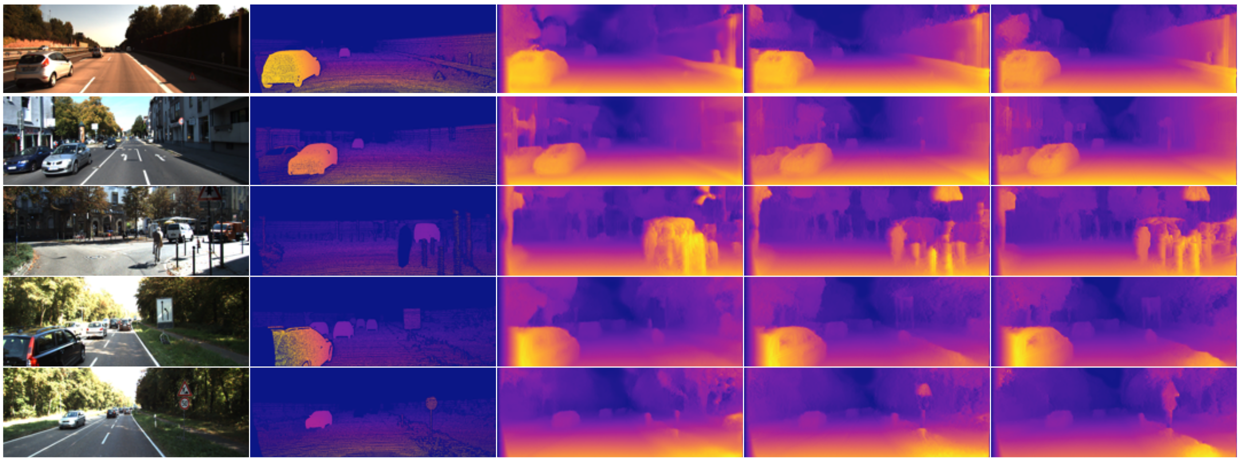

Figure 2.4: *Qualitative results on KITTI 2015 split.* From left to right: input images, ground-truth depths, results of Godard et al.[GMB17], our results using a generic decoder and our results the proposed decoder. Our approach generates more consistent depths (row 1: walls on right, row 2: building on left) and recovers more detailed structures (row 3: biker and poles on right, rows 4, 5: street signs), with the two-branch decoder recovering the most.

the left-right consistency regularizer [GMB17]. We also substitute image Laplacian with image gradients for edge-aware weights. In addition, we find that adaptive regularization and bilateral cyclic consistency contribute similarly to the improvements of the models. However, when combined they achieve significantly improvements over the baseline method (and all variants) in every metric. Furthermore, when using our proposed decoder, we again surpass all variants on every metric. We additionally outperform [ATP18], who uses a GAN to constrain the output disparities to produce photo-realistic images during reconstruction. This result aligns with our performance on accuracy metrics – our method produces accurate and realistic depths.

2.6.3 Generalizing to Different Datasets: Make3d

To show that our model generalizes, we present our qualitative results in Fig. 2.5 and and quantitative results in Table 2.5 on the Make3d dataset [SSN09] containing 134 test images with 2272 × 1707 resolution. Make3d provides range maps (resolution of 305 × 55) for ground-truth depths, which must be rescaled and interpolated. We use the central cropping

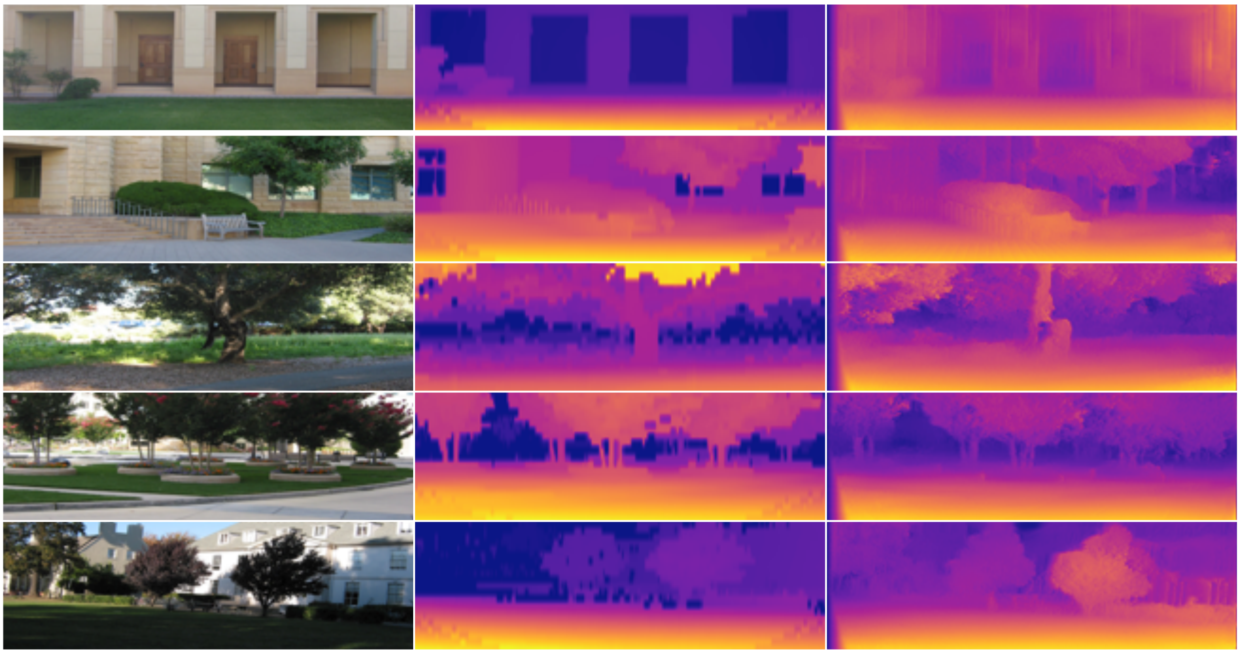

Figure 2.5: *Qualitative results*$^{2.1}$ *on Make3d [SSN09] with maximum depth of 70 meters.* Left to right: input images, ground-truth disparities, our results. Despite being trained on KITTI, we are still able to recover the 3d scene on Make3d.

		Error Metrics			
Method	Supervised	AbsRel	Sq Rel	RMS	$\log_{10}$
Karsch et al. [KLK12]	Yes	0.417	4.894	8.172	0.144
Liu et al. [LSL16]	Yes	0.462	6.625	9.972	0.161
Laina et al. [LRB16]	Yes	**0.198**	**1.665**	**5.461**	**0.082**
Godard et al. [GMB17]	No	0.468	9.236	12.525	0.165
Ours	No	0.454	8.470	12.211	0.163
Ours*	No	**0.427**	**8.183**	**11.781**	**0.156**

Table 2.5: *Quantitative results*$^{2.1}$ *on Make3d [SSN09] with maximum depth of 70 meters.* The unsupervised methods listed are all trained on KITTI Eigen split. Despite being trained on KITTI, we perform comparably to a number of supervised methods trained on Make3d.

proposed by [GMB17] where we generate a 852 × 1707 crop centered on the image. We use the standard $C1$ evaluation metrics[2.1] proposed for Make3d and limit the maximum depth to 70 meters. The results of the supervised methods are taken from [GMB17]. Because Make3d does not provide stereo pairs, we are unable to train on it. However, we find that despite having trained our model on KITTI Eigen split, our performance is comparable to that of supervised methods trained on Make3d and is better than the baseline across all metrics.

2.7 Discussion

In this work, we proposed an adaptive weighting scheme (Sec. 2.3.3) that is both spatially and time varying, allowing for not only a data-driven, but also model-driven approach to regularization. Moreover, we introduce a bilateral cyclic consistency constraint that not only enforces consistency between the left and right disparities, but also removes stereo dis-occlusions while discounting unresolved occlusions when combined with our weighting scheme. Finally, we propose a two-branch decoder that achieves the state-of-the-art by learning features to improve data residual for imposing our adaptive regularity. We achieve state-of-the-art performance on two KITTI benchmarks and show that our method generalizes to Make3d. Our two-branch decoder further improves over those results. Our experiments (Table 2.2, Table 2.3 and 2.4) show that our approach produces depth maps with more details while maintaining global correctness.

For future work, we plan to improve robustness to specular and transparent surfaces as these regions tend to produce inconsistent depths. We are also exploring more sophisticated regularizers in place of the simple disparity gradient. Finally, we believe that the task should drive the network architecture. Rather than using a generic network, finding a better architectural fit could prove to be ground-breaking and further push the state-of-the-art.

CHAPTER 3

Geo-Supervised Visual Depth Prediction

3.1 Introduction

The visual world is heavily affected by gravity, including the shape of many artifacts such as buildings and roads, and even natural objects such as trees. Gravity provides a globally consistent orientation reference that can be reliably measured with low-cost inertial sensors present in mobile devices from phones to cars. We call a machine learning system able to exploit global orientation, *geo-supervised*. Gravity can be easily inferred from inertial sensors without the need for dead-reckoning, and the effect of biases is negligible in the context of our application.

To measure the influence of gravity as a supervisory signal, we choose the extreme example of predicting depth from a single image. This is, literally, an impossible task in the sense that there are infinitely many three-dimensional (3D) scenes that can generate the same image. So, any process that yields a point estimate has to rely heavily on priors. We call the resulting point estimate a *hypothesis*, or *prediction*, and use public benchmark datasets to quantitatively evaluate the improvement brought about by exploiting gravity. Of course, only certain objects have a shape that is influenced by gravity. Therefore, our prior has to be applied *selectively*, in a manner that is informed by the semantics of the scene.

Our approach to geo-supervised Visual Depth Prediction is based on training a system end-to-end to produce a map from a single image and an estimate of the orientation of gravity in the (calibrated) camera frame to an inverse depth (disparity) map. In one mode of operation, the training set uses calibrated and rectified stereo pairs, together with a semantic segmentation module, to evaluate a loss function differentially on the images where geo-referenced objects are present. In a second mode, we use monocular videos instead and minimize the reprojection (prediction) error. Optionally, we can leverage modern visual-inertial odometry (VIO) and mapping systems that are becoming ubiquitous from hand-held devices to cars.

The key to our approach is a prior, or regularizer, that selectively biases certain regions of the image that correspond to geo-referenced classes such as roads, buildings, vehicles, and trees. Specifically, points in space that lie on the surface of such objects should have normals that either align with, or are orthogonal to, gravity. This is in addition to standard regularizers used for depth prediction, such as left-right consistency and piecewise smoothness.

While at training time a semantic segmentation map is needed to apply our prior selectively, it is never passed as input to the network. Therefore, at test time it is not needed, and an image is simply mapped to the disparity.

The ultimate test for a prior is whether it helps improve end-performance. To test our prior, we first incorporated it into two top-performing methods, one binocular (Sect. 3.5.2) and one monocular (Sect. 3.5.3), in the KITTI benchmark [GLU12], and showed consistent performance improvement in all metrics. To further challenge our prior, we took two other baselines which were not the top performers. We then added our prior and tested the results against the top performers in the latest benchmark. We also performed generalizability tests (Sect. 3.5.5), ablation studies (Sect. 3.5.4) and demonstrated our approach with VIO on hand-held devices (Sect. 3.5.6).

3.2 Related work

Early learning-based depth prediction approaches [SCN06, SSN09, KWI13, KLK12] predict depth using local image patches and then refine it using Markov random fields (MRFs). Recent works [EPF14, LRB16] leverage deep networks to directly learn a representation for depth prediction where the networks are typically based on the multi-scale fully convolutional encoder-decoder structure. These methods are fully supervised and do not generalize well outside the datasets on which they are trained. Latest self-supervised methods [GBC16, GMB17, ZBS17] have shown better performance on benchmarks with better generalization.

There is a large body of work [MWA18, YS18b, WBZ18b, ZGW18] on self-supervised monocular depth prediction following Godard *et al.* [GMB17] and Zhou *et al.* [ZBS17], which simply use the reprojection error as a learning criterion, as has been customary in 3D reconstruction for decades. Generic priors such as piecewise smoothness and left-right consistency are also encoded into the network as additional loss terms. Our work is in-line with these self-supervised approaches, but we also exploit class-specific regularizers beyond the generic ones.

In terms of exploiting the relation of different geometric quantities in an end-to-end learning framework, closely related works include [WSR16, QLL18, LYC18], where surface normals are explicitly computed by using either a network [WSR16] or some heuristics [QLL18]. While the former is computation intensive, the latter relies on heuristics and thus is suboptimal. In contrast, by using losses proposed in this paper, we directly regularize depth via the depth-gravity relation without a separate surface normal predictor. Besides, both [WSR16] and [LYC18] are supervised, while ours is self-supervised with the photometric loss and guided by global orientation and the semantics of the scene.

Earlier work on semantic segmentation [SJC08] relied on local features, and have been improved by incorporating global context using various structured prediction techniques [KK11, RKT09]. Starting from the work of Long *et al.* [LSD15], fully convolutional encoder-decoder networks have been a staple in semantic segmentation. Although we do not address semantic segmentation, we leverage per-pixel semantic labeling enabled by existing systems to aid depth prediction in the form of providing class-specific priors and an attention mechanism to selectively apply such priors, which is different from joint segmentation and depth prediction approaches [JGK17].

The idea of using class-specific priors to facilitate reconstruction is not new [HZC13, KLD14]. In [HZC13], class-specific shape priors in the form of spatially varying anisotropic smoothness terms are used in an energy minimization framework to reconstruct small objects. Though promising, this system does not scale well. An efficient inference framework [KK11] has been used with a CRF model over a voxel-grid to achieve real-time performance by [KLD14]. While all these methods explore class-specific priors in various ways, none has used them in an end-to-end learning framework. Also, all the methods above take range images as inputs, which are then fused with semantics during optimization, while ours exploits semantics at an earlier stage – when generating such range images which themselves can serve as priors for dense reconstruction and other inference tasks.

3.3 Methodology

In this section, we introduce our loss functions as regularizers added to existing models at training time, in addition to data terms (photometric loss) and generic regularizers (smoothness loss). We dub our loss semantically informed geometric loss (SIGL) because geometric constraints are selectively applied to certain image regions, where a semantic segmentation module informs the selection. Fig. 3.1 illustrates part of our training diagram. In Sect. 3.3.3, we review baseline models used in our experiments and show that the application of our losses on top of them improves performance (Sect. 3.5).

3.3.1 Semantically informed geometric loss

During training, we assume to be given a partition of the image plane into semantic classes $c \in C$ that have a consistent geometric correlate. For instance, a pixel with image coordinates $(x, y) \in \mathbb{R}^2$ and class $c(x, y) =$ "road" is often associated to a normal plane oriented along the vertical direction (direction of gravity), whereas $c =$"building" has a normal vector orthogonal to it. We also assume we are given the calibration matrix K of the camera capturing the images, so the pixel coordinates (x, y) on the image plane back-project to points in space via

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} Z(x, y) \tag{3.1}$$

where $Z(x, y)$ is the depth Z of the point along the projection ray determined by (x, y).

Any subset $\Omega \subset \mathbb{R}^2$ of the image plane that is the image of a spatial plane with normal vector $N \in \mathbb{R}^3$, at distance $\|N\|$ from the center of projection, satisfies a constraint of the form $\mathbf{X}_i^T N = 1$ for all i, assuming the plane does not go through the optical center. Stacking all the points into matrix $\bar{\mathbf{X}} \doteq [\mathbf{X}_1, \mathbf{X}_2 \cdots \mathbf{X}_M]^\top$, we have $\bar{\mathbf{X}} N = \mathbf{1}$, where $\mathbf{1}$ is a vector of M ones, and $M = |\Omega|$ is the cardinality of the set Ω. If the direction, but not the norm, of the vector N is known, a scale-invariant constraint can be easily obtained by removing the mean of the points, so that (details in Sect. 3.3.2)

$$(\mathbf{I} - \frac{1}{M}\mathbf{1}\mathbf{1}^\top)\bar{\mathbf{X}} N = 0. \tag{3.2}$$

The scale-invariant constraint above can be used to define a loss to penalize deviation from planarity:

$$L_{HP}(\Omega_{HP}) = \frac{1}{|\Omega_{HP}|}\|(\mathbf{I} - \frac{1}{|\Omega_{HP}|}\mathbf{1}\mathbf{1}^\top)\bar{\mathbf{X}}\gamma\|_2^2 \tag{3.3}$$

where N in Eq. (3.2) is replaced by normalized gravity γ due to the homogeneity of Eq. (3.2), and the squared norm is taken assuming the network predicts per-pixel depth $Z(x, y)$ up to additive zero-mean Gaussian noise. $\Omega_{HP} \subset \mathbb{R}^2$ is a subset of the image plane whose associated semantic classes have horizontal surfaces, such as "road", "sidewalk", "parking lot", etc. We call this loss "horizontal plane" loss, where the direction of gravity γ can be reliably and globally estimated.

Similarly, a "vertical plane" loss can be constructed to penalize deviation from a vertical plane whose normal N has *both unknown direction and norm* but lives in the null space of γ, *i.e.,* $N \in \mathcal{N}(\gamma)$. Thus, the vertical plane loss reads

$$L_{VP}(\Omega_{VP}) = \min_{\substack{N \in \mathcal{N}(\gamma) \\ \|N\|=1}} \frac{1}{|\Omega_{VP}|}\|(\mathbf{I} - \frac{1}{|\Omega_{VP}|}\mathbf{1}\mathbf{1}^\top)\bar{\mathbf{X}} N)\|_2^2 \tag{3.4}$$

where the constraint $\|N\| = 1$ avoids trivial solutions $N = 0$ again due to the homogeneity of the objective; Ω_{VP} is a subset of the image plane whose associated semantic classes have vertical surfaces, such as "building", "fence", "billboard", etc. The constrained minimization problem in the vertical plane loss L_{VP} is due to the unknown direction of the surface normals and introduces some difficulties in training. We discuss approximations in Sect. 3.3.2.

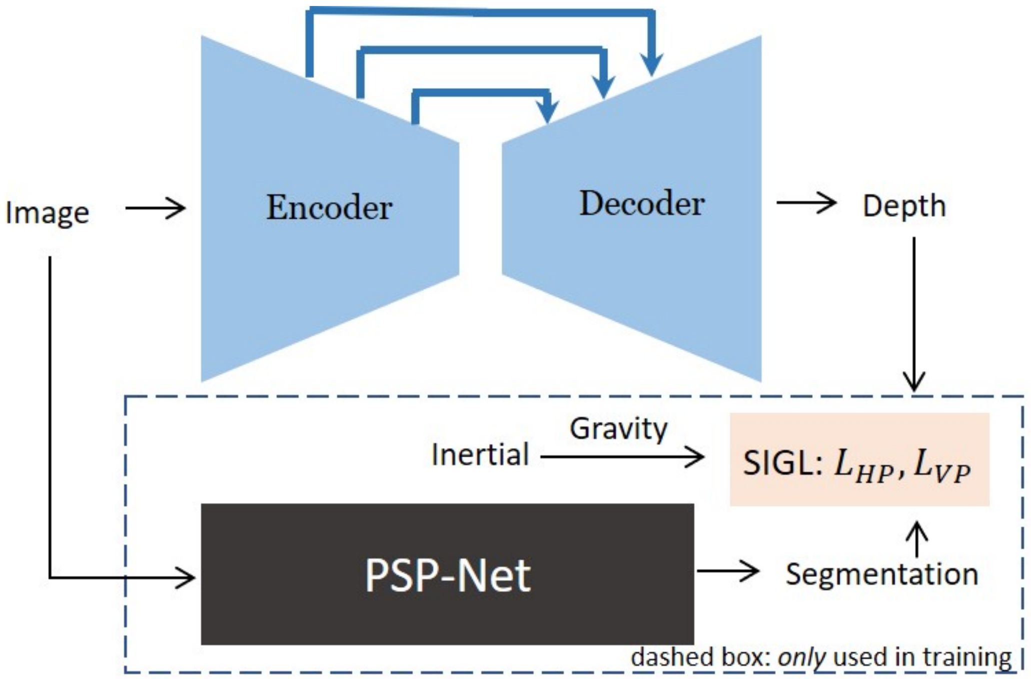

Figure 3.1: *Illustration of geo-supervised visual depth prediction.* Our visual depth predictor is an encoder-decoder convolutional neural network with skip connections. At inference time, the network takes an RGB image as the only input and outputs an inverse depth map. At training time, gravity extracted from inertial measurements biases the depth prediction *selectively*, which is informed by semantic segmentation produced by PSPNet. The other identical stream of the network and the photometric losses used for training are omitted in this figure.

3.3.2 Explanation of the objectives

Our idea is essentially to use priors about surface normals to regularize depth prediction. An intuitive way to achieve this is to compute the surface normals from the depth values first and then impose regularity, which will eventually bias the depth predictor via backpropagation. However, such a method involves normal estimation from depth, which can be problematic, especially with a simplistic but noisy normal estimator [QLL18].[1] On the other hand, one

[1] For instance, one can compute the point-wise sur-mated by connecting the underlying point to its near-face normal as the cross product of two vectors tangentest neighbors on the surface.
to the surface, where the tangent vectors are approxi-

could train a deep network to compute surface normals [WSR16], which is costly. Therefore, *we do not compute surface normals but directly regularize the depth values* via the scale-invariant constraint Eq. (3.2) which is a function of depth and the direction of gravity.

In what follows, we give an explanation of L_{HP} Eq. (3.3) from a statistical perspective. Let $M = |\Omega_{HP}|$ to avoid notation clutter and expand Eq. (3.3):

$$(\mathbf{I} - \frac{1}{M}\mathbf{1}\mathbf{1}^\top)\bar{\mathbf{X}}\gamma \tag{3.5}$$

$$= \begin{bmatrix} 1-\frac{1}{M} & \cdots & -\frac{1}{M} \\ \vdots & \ddots & \vdots \\ -\frac{1}{M} & \cdots & 1-\frac{1}{M} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1^\top\gamma \\ \mathbf{X}_2^\top\gamma \\ \cdots \\ \mathbf{X}_M^\top\gamma \end{bmatrix} = \begin{bmatrix} \vdots \\ \left(\mathbf{X}_i - \frac{1}{M}\sum_{j=1}^M \mathbf{X}_j\right)^\top\gamma \\ \vdots \end{bmatrix} \tag{3.6}$$

Let $\mu = \frac{1}{M}\sum_{j=1}^M \mathbf{X}_j$ be the sample mean of the 3D coordinates and the horizontal plane loss L_{HP} reads

$$L_{HP}(\Omega_{HP}) = \frac{1}{M}\sum_{i=1}^M \left((\mathbf{X}_i - \mu)^\top\gamma\right)^2 \tag{3.7}$$

which is the sample variance of the 3D coordinates projected to the direction of gravity γ (coinciding with the surface normal for horizontal planes). To minimize L_{HP} is to minimize the variance of the 3D coordinates along the surface normal.

Similarly, to minimize L_{VP} Eq. (3.4) is to minimize the variance of the 3D coordinates along some direction perpendicular to gravity. However, if the direction is unknown, one needs to jointly solve the direction while minimizing L_{VP}, which explains the constrained quadratic problem in L_{VP}. Though this can be solved via eigendecomposition, the gradients of the solver – needed in backpropagation – are non-trivial to compute. In fact, representing an optimization procedure as a layer of a neural network is an open research problem [AK17]. To alleviate both numerical and implementation difficulties, we uniformly sample unit vectors from the null space of gravity and compute the minimum of the objective over the samples as an approximation to the loss. Empirically, we found using eight directions sampled every 45 degrees from 0 to 360 generally performs well.

3.3.3 View synthesis as supervision and baselines

To showcase the ability to improve upon existing self-supervised monocular depth prediction networks, we add our losses to two publicly available models – Godard [GMB17] (`LR-Consistency`) and Yin [YS18b] (`GeoNet`) – as baselines and perform both quantitative and qualitative comparisons. We additionally apply our losses to Zhan [ZGW18] (`Stereo-Temporal`) and Wang [WBZ18b] (`DDVO`), the state-of-the-art methods in their respective training setting, stereo pairs/videos, and monocular videos. `LR-Consistency` is trained with rectified stereo image pairs, `GeoNet` and `DDVO` use monocular videos while `Stereo-Temporal` uses stereo videos. At test time, all training settings result in a system that takes a single image as input and predicts an inverse depth map as output. We show that by applying our losses to the baselines `LR-Consistency` and `GeoNet`, we achieve better performance than the state-of-the-art methods `Stereo-Temporal` and `DDVO`. Furthermore, we produce new state-of-the-art results by applying our losses to `Stereo-Temporal` and `DDVO`.

3.3.3.1 Training with stereo pairs

At training time, our first baseline model (`LR-Consistency`) takes a single left image as its input and predicts two disparity maps $D^L, D^R : \mathbb{R}^2 \supset \Omega \to \mathbb{R}_+$ for both left and right cameras. The network follows the fully convolutional encoder-decoder structure with skip connections. The total loss consists of three terms: Appearance loss, smoothness of disparity and left-right consistency, each of which is evaluated on both the left and the right streams across multiple scale levels. Here we address the view synthesis loss, which serves as the data term and is part of the appearance loss:

$$L^L_{\text{vs}} = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \|I^L(x,y) - I^R(x + D^L(x,y), y)\|_1. \tag{3.8}$$

The view synthesis loss is essentially the photometric difference of the left image $I^L(x,y)$ and the right image warped to the left view $I^R(x + D^L(x,y), y)$ according to the left disparity prediction $D^L(x,y)$. The right view synthesis loss is constructed in the same way. Though only one disparity map is needed at inference time, it has been shown that predicting both left and right disparity maps and including the left-right consistency loss Eq. (3.9) are in general beneficial [GMB17].

$$L_{\text{lr}}^{L} = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \|D^{L}(x,y) - D^{R}(x + D^{L}(x,y), y)\|_1 \tag{3.9}$$

3.3.3.2 Training with stereo videos

In our second baseline `Stereo-Temporal`, stereo videos are used to train a monocular depth predictor, where two frames of a stereo pair and another frame one time step ahead are involved in constructing a stereo-temporal version of the photometric loss: For the stereo pair, Eq. (3.8) is applied while for the temporal pair, Eq. (3.10) (detailed below) is applied.

3.3.3.3 Training with monocular videos

To train our third and fourth baseline models (`GeoNet` and `DDVO`), a single reference frame I_t is fed into the depth network and frames $I_{t'}, t' \in W_t$ in a temporal window centered at t are used to construct the view synthesis loss, also known as reprojection error:

$$L_{\text{vs}} = \frac{1}{|W_t||\Omega|} \sum_{t'\in W_t} \sum_{(x,y)\in\Omega} \|I_t(x,y) - I_{t'}\big(\pi(\hat{g}_{t't}\mathbf{X})\big)\|_1 \tag{3.10}$$

which is the difference between the reference frame I_t and neighboring frames $I_{t'}$ warped to it. $\mathbf{X}$ is the back-projected point defined in Eq. (3.1), π is a central (perspective) projection, and $\hat{g}_{t't}$ is the relative camera pose up to an unknown scale predicted by an auxiliary pose network which takes both I_t and $I_{t'}$ as its input. Note that the pose and depth networks are coupled via the view synthesis loss at training time; at test time, the depth network alone is needed to perform depth prediction with a single image as its input. Interestingly, in Sect. 3.5.6 we found that replacing the pose network with pose estimation from VIO produces better results compared to the multi-task learning diagram where pose and depth networks are trained simultaneously, which sheds light on the use of classic SLAM/Odometry systems in developing better learning algorithms.

A detailed discussion about other losses serving as regularization terms is beyond the scope of this paper and can be found in [GMB17, ZBS17, YS18b, WBZ18b].

3.4 Implementation Details

3.4.1 Semantic segmentation

At training time, we use PSPNet [ZSQ17] pre-trained on the CityScapes dataset [COR16] provided by the authors to obtain per-pixel labeling. For every pixel $(x, y) \in \mathbb{R}^2$, a probability distribution over 19 classes is predicted by PSPNet, of which the most likely class $c(x, y) \in C$ determines the orientation of the surface where the back-projected point $\mathbf{X}$ sits. We group the 19 classes into 7 categories[2] according to the CityScapes benchmark and test our losses on all of them. Empirically, we found that it is most beneficial to apply our losses to the "flat", "vehicle" and "construction" categories and therefore all the comparisons on KITTI against baseline methods are made with these categories regularized. The influence of other categories is studied in Sect. 3.5.4.

3.4.2 Gravity

For imagery captured by a static platform equipped with an inertial measurement unit (IMU), one can use the gravity $\gamma_b \in \mathbb{R}^3$ measured in the body frame (coinciding with the IMU frame) and simply apply the body-to-camera rotation $R_{cb} \in \mathrm{SO}(3)$ to obtain the gravity in the camera frame $\gamma = R_{cb}\gamma_b$ which is then used in Eq. (3.3) and (3.4). For moving platforms, one resorts to robust VIO, which is well studied [MR07, TCS15]. In Sect. 3.5.6, we demonstrated our approach on a visual-inertial odometry dataset, where both camera pose and gravity are estimated online by VIO.

For our experiments on the KITTI dataset, thanks to the GPS/IMU sensor package which provides linear acceleration of the sensor platform measured both in the body frame

[2] "flat": road, sidewalk; "human": rider, person;traffic light, traffic sign; "nature": vegetation, terrain; "vehicle": car, truck, bus, train, motorcycle, bicycle;"sky": sky.
"construction": building, wall, fences; "object": pole,

Metric	Definition
AbsRel	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\frac{\lvert Z(x,y)-Z^{\text{gt}}(x,y)\rvert}{Z^{\text{gt}}(x,y)}$
SqRel	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\frac{\lvert Z(x,y)-Z^{\text{gt}}(x,y)\rvert^2}{Z^{\text{gt}}(x,y)}$
RMSE	$\sqrt{\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert Z(x,y)-Z^{\text{gt}}(x,y)\rvert^2}$
RMSE log	$\sqrt{\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert\log Z(x,y)-\log Z^{\text{gt}}(x,y)\rvert^2}$
$\log_{10}$	$\frac{1}{\lvert\Omega\rvert}\sum_{(x,y)\in\Omega}\lvert\log Z(x,y)-\log Z^{\text{gt}}(x,y)\rvert$
Accuracy	% of $Z(x,y)$ s.t. $\delta \doteq \max\left(\frac{Z(x,y)}{Z^{\text{gt}}(x,y)}, \frac{Z^{\text{gt}}(x,y)}{Z(x,y)}\right) <$ threshold

Table 3.1: *Error and Accuracy Metrics.* $Z(x,y)$ is the predicted depth at $(x,y)\in\Omega$ and $Z^{\text{gt}}(z,y)$ is the corresponding ground truth. Three different thresholds $(1.25, 1.25^2$ and $1.25^3)$ are used in the accuracy metric as a convention in the literature.

$(\alpha_b \in \mathbb{R}^3)$ and the spatial frame $(\alpha_s \in \mathbb{R}^3)$, we are able to compute the spatial-to-body rotation $R_{bs} \in \mathrm{SO}(3)$ and then bring the gravity $\gamma_s = [0,0,9.8]^\top$ from the spatial frame to the camera frame $\gamma = R_{cb}R_{bs}\gamma_s$. In all settings, R_{cb} (rotational part of the body-to-camera transformation) is obtained via offline calibration procedures.

3.4.3 Training details

A GTX 1080 Ti GPU and Adam [KB14] optimizer are used in our experiments. Depending on different model variants and input image sizes, training time varies from 8 hours to 16 hours. For `LR-Consistency` and `GeoNet` which were initially implemented in TensorFlow, we implemented our losses also in TensorFlow and applied them to the existing code bases. Code of `Stereo-Temporal` is available online, but in Caffe, thus we migrated their model to TensorFlow and applied our losses. We also implemented our losses in PyTorch, which were then applied to `DDVO` of which the PyTorch version was made available by the author. Our code is available at `https://github.com/feixh/GeoSup`.

3.5 Experiments

To enable quantitative evaluation, we exploit the KITTI benchmark, and test our approach against the state-of-the-art as described in detail below (Sect. 3.5.2&3.5.3). We also carried out ablation studies (Sect. 3.5.4) and tested the generalizability of our approach (Sect. 3.5.5). In addition to KITTI, which features planar motion in driving scenarios, we have conducted experiments on VISMA dataset [FS18] – an indoor visual-inertial odometry dataset captured under non-trivial ego-motion (Sect. 3.5.6).

3.5.1 KITTI Eigen split

We compare our approach with recent state-of-the-art methods on the monocular depth prediction task using the KITTI Eigen split [EPF14] in two training domains: stereo pairs/videos and monocular videos (Sect. 3.3.3). The Eigen split test set contains 697 test images selected from 29 of 61 scenes provided by the raw KITTI dataset. Of the remaining 32 scenes containing 23,488 stereo pairs, 22,600 pairs are used for training, and the rest is used for validation per the training split proposed by [GBC16]. To generate ground truth depth maps for validation and evaluation, we take the Velodyne data points associated with each image and project them from the Velodyne frame to the left RGB camera frame. Each resulting ground truth depth map covers approximately 5% of the corresponding image and may be erroneous. To handle this, first, we use the cropping scheme proposed by [GBC16], which masks out the potentially erroneous extremities from the left, right and top areas of the ground truth depth map. Then we evaluate depth prediction only at pixels where ground truth depth is available. For visualization, we linearly interpolate each sparse depth map to cover the entire image (Fig. 3.2).

We additionally provide quantitative evaluations of variants of the models pre-trained on CityScapes and fine-tuned on KITTI. CityScapes dataset contains 22,973 training stereo pairs captured in various cities across Germany with a similar modality as KITTI. We cropped each input image to keep only the top 80% of the image, removing the reflective hood.

The error and accuracy metrics, which are initially proposed by [EPF14] and adopted by others, are used (Table 3.1). Also as a convention in the literature, performances evaluated with depth prediction capped at 50 and 80 meters are reported as suggested by [GMB17]. The choice of 80 meters is two-fold: 1) maximum depth present in the KITTI dataset is on the order of 80 meters and 2) non-thresholded measures can be sensitive to the significant errors in depth caused by prediction errors at small disparity values. For the same reason, depth prediction is capped at 70 meters in the Make3D experiment. Prediction capped at 50 meters is also evaluated since depth at closer range is more applicable to real-world scenarios.

3.5.2 Training with stereo pairs

The first baseline we adopt is Godard [GMB17] (with `VGG` [SZ14] as feature extractor), to which SIGL is imposed at training time along with the view synthesis loss Eq. (3.8) and other generic regularizers used in [GMB17]. The model is trained from scratch with stereo pairs following the Eigen split and compared to both supervised [EPF14, LSL16] and self-supervised methods [GMB17, ZGW18]. In addition, we apply our losses to variants of the baseline (with `ResNet` [HZR16] as feature extractor; w/ & w/o post-processing) and evaluate different training schemes (w/ & w/o pre-training on CityScapes). Quantitative comparisons can be found in Table 3.2, where the results with SIGL added as an additional regularizer follow the results of the baseline models and variants. In the column marked "Data", `K` refers to Eigen split benchmark on the KITTI dataset, and `CS` refers to the CityScapes dataset. Methods marked with `CS+K` are pre-trained on CityScapes and then fine-tuned on KITTI Eigen split. `pp` denotes post-processing. Cap Xm means depth predictions are capped at X meters. Results of Zhan [ZGW18] `Stereo-Temporal` are taken from their paper. The rest of the results are taken from [GMB17] unless otherwise stated.

We want to remind the reader that the first baseline model atop which we built ours is Godard [GMB17] `VGG` which initially performed worse than the `Stereo-Temporal` model of Zhan [ZGW18] by a large margin, but by applying our losses to the baseline at training time we managed to boost its performance and make it perform even better than the

Method	Data	Error metric				Accuracy ($\delta <$)		
		AbsRel	SqRel	RMSE	RMSElog	1.25	1.25^2	1.25^3
Depth: cap 80m								
TrainSetMean*	`K`	0.361	4.826	8.102	0.377	0.638	0.804	0.894
Eigen [EPF14] `Coarse`*	`K`	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen [EPF14] `Fine`*	`K`	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu [LSL16]*	`K`	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Godard [GMB17] `VGG`	`K`	0.148	1.344	5.927	0.247	0.803	0.922	0.964
+SIGL	`K`	**0.139**	**1.211**	**5.702**	**0.239**	**0.816**	**0.928**	**0.966**
Zhan [ZGW18] `Stereo-Temporal`	`K`	0.144	1.391	5.869	0.241	0.803	0.928	0.969
+SIGL	`K`	**0.137**	**1.061**	**5.692**	**0.239**	**0.805**	0.928	0.969
Godard [GMB17] `VGG pp`	`CS+K`	0.124	1.076	5.311	0.219	0.847	0.942	0.973
+SIGL	`CS+K`	**0.114**	**0.885**	**4.877**	**0.203**	**0.858**	**0.950**	**0.978**
Godard [GMB17] `ResNet pp`	`CS+K`	0.114	0.898	4.935	0.206	0.861	0.949	0.976
+SIGL	`CS+K`	**0.112**	**0.836**	**4.892**	**0.204**	**0.862**	**0.950**	**0.977**
Depth: cap 50m								
Garg [GBC16]	`K`	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard [GMB17] `VGG`	`K`	0.140	0.976	4.471	0.232	0.818	0.931	0.969
+SIGL	`K`	**0.132**	**0.891**	**4.312**	**0.225**	**0.831**	**0.936**	**0.970**
Zhan [ZGW18] `Stereo-Temporal`	`K`	0.135	0.905	4.366	0.225	0.818	0.937	0.973
+SIGL	`K`	**0.131**	**0.829**	**4.217**	**0.224**	**0.824**	0.937	0.973
Godard [GMB17] `VGG pp`	`CS+K`	0.112	0.680	3.810	0.198	0.866	0.953	0.979
+SIGL	`CS+K`	**0.108**	**0.658**	**3.728**	**0.192**	**0.870**	**0.955**	**0.981**
Godard [GMB17] `ResNet pp`	`CS+K`	0.108	0.657	3.729	0.194	0.873	0.954	0.979
+SIGL	`CS+K`	**0.106**	**0.615**	**3.697**	**0.192**	**0.874**	**0.956**	**0.980**

* With ground truth depth supervision.

+SIGL: training with SIGL enabled

Table 3.2: *Training with stereo pairs on KITTI.* `K` denotes the KITTI dataset, `CS+K` denotes pretraining on Cityscape and finetuning on KITTI. Depth values are capped at 50 and 80 meters. Our method consistently improves baseline algorithms.

`Stereo-Temporal` model at test time. Note that the `Stereo-Temporal` model also exploits temporal information in addition to stereo pairs for training while our first baseline built atop Godard does not.

As a second baseline, we apply our losses additionally to the `Stereo-Temporal` model of Zhan to further push the state-of-the-art. Table 3.2 shows that our losses improve the `Stereo-Temporal` model across all error metrics with the accuracy metrics $\delta < 1.25^2$ and $\delta < 1.25^3$ being comparable. Another variant of Zhan's model pre-trains on NYU-V2 [SHK12] in a fully supervised fashion and is therefore not pertinent to this comparison. Fig. 3.2 shows a head-to-head qualitative comparison of ours and the baseline models.

3.5.3 Training with monocular videos

To demonstrate the effectiveness of our loss in the second training setting (monocular videos), we impose SIGL to our third (Yin [YS18b]) and fourth (Wang [WBZ18b]) baseline. Using the KITTI Eigen split, we follow the training and validation 3-frame sequence selection proposed by [ZBS17] where the first and third frames are treated as the source views and the central (second) frame is treated as the reference as in Eq. (3.10). Of the 44,540 total sequences, 40,109 are used for training and 4,431 for validation. We evaluate our system on the aforementioned 697 test images [EPF14]. The same training and evaluation scheme are also applied to other top-performing methods [ZBS17, MWA18] in addition to the selected baselines.

Table 3.3 shows detailed comparisons against state-of-the-art self-supervised methods trained using monocular video sequences. We compare against best-performing model variants of Wang [WBZ18b] (`PoseCNN` & `PoseCNN+DDVO`) and Yin [YS18b] (`ResNet`) with and without pre-training on CityScapes. By adding our losses to existing models, we observe systematic performance improvement across all metrics. Though initially performing worse than Wang [WBZ18b] `PoseCNN+DDVO`, Yin [YS18b] `ResNet` with the proposed losses even outperforms the original `PoseCNN+DDVO`. Moreover, we achieve new state-of-the-art by adding our losses to `PoseCNN+DDVO` trained on both CityScapes and KITTI. Fig. 3.2 illustrates

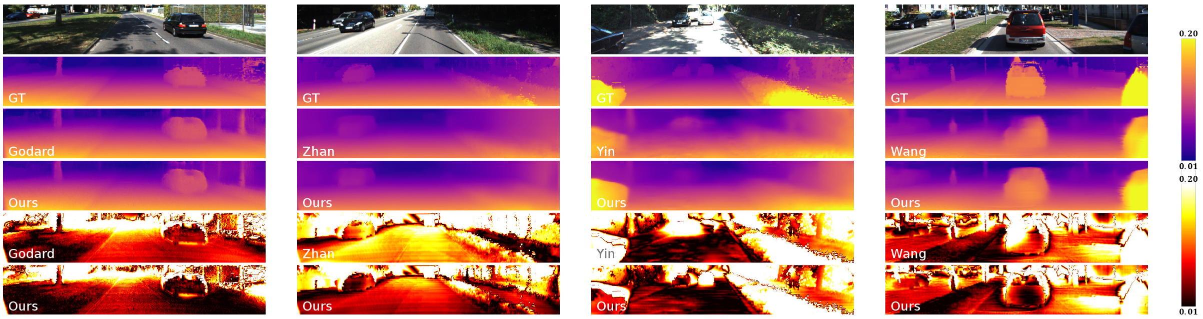

Figure 3.2: *Qualitative results on KITTI Eigen split.* (best viewed at 5× with color) Top to bottom, each column shows an input RGB image, the corresponding ground truth inverse depth map, the predictions of baseline models trained without and with our priors, AbsRel error maps of baseline models trained without and with our priors. All the models are trained on KITTI Eigen split. For the purpose of visualization, ground truth is interpolated and all the images are cropped according to [GBC16]. For the error map, darker means smaller error. Typical image regions where we do better (darker in the error map) include cars, roads and walls.

representative image regions where we do better.

3.5.4 Ablation study

To study the contribution of each semantic category to the performance improvement, we performed an ablation study: We apply our losses to different semantic categories, one at a time, train the network until convergence, and show how the quality of depth prediction varies (Table 3.4). In Table 3.4, Godard *et al.* [GMB17] is the baseline model where only the most generic regularizers, *e.g.,* smoothness and consistency, are used. The second column indicates the semantic category of which the depth prediction is regularized using our losses in addition to the generic regularizers. For the meaning of the semantic categories, see Sect. 3.4.1.

It turns out that the "flat" category contributes most to the performance gain over the baseline model, which is expected because most of the KITTI images contain a large portion of roads and sidewalks. We also observed that regularization of the "construction" and

Method	Data	Error metric				Accuracy ($\delta <$)		
		AbsRel	SqRel	RMSE	RMSElog	1.25	1.25^2	1.25^3
Depth: cap 80m								
Zhou [ZBS17]	K	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Mahjourian [MWA18]	K	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Yin [YS18b] ResNet	K	0.155	1.296	5.857	0.233	0.793	0.931	0.973
+SIGL	K	**0.142**	**1.124**	**5.611**	**0.223**	**0.813**	**0.938**	**0.975**
Wang [WBZ18b] PoseCNN	K	0.155	1.193	**5.613**	0.229	0.797	0.935	0.975
+SIGL	K	**0.147**	**1.076**	5.640	**0.227**	**0.801**	0.935	0.975
Wang [WBZ18b] PoseCNN+DDVO	K	0.151	1.257	5.583	0.228	**0.810**	0.936	0.974
+SIGL	K	**0.146**	**1.068**	**5.538**	**0.224**	0.809	**0.938**	**0.975**
Zhou [ZBS17]	CS+K	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian [MWA18]	CS+K	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Yin [YS18b] ResNet	CS+K	0.153	1.328	5.737	0.232	0.802	0.934	0.972
+SIGL	CS+K	**0.147**	**1.076**	**5.468**	**0.222**	**0.806**	**0.938**	**0.976**
Wang [WBZ18b] PoseCNN+DDVO	CS+K	0.148	1.187	5.496	0.226	0.812	0.938	0.975
+SIGL	CS+K	**0.142**	**1.094**	**5.409**	**0.219**	**0.821**	**0.941**	**0.976**
Depth: cap 50m								
Zhou [ZBS17]	K	0.201	1.391	5.181	0.264	0.696	0.900	0.966
Mahjourian [MWA18]	K	0.155	0.927	4.549	0.231	0.781	0.931	0.975
Yin [YS18b] ResNet	K	0.147	0.936	4.348	0.218	0.810	0.941	0.977
+SIGL	K	**0.135**	**0.834**	**4.193**	**0.208**	**0.831**	**0.948**	**0.979**
Wang [WBZ18b] PoseCNN†	K	0.149	0.920	4.303	0.216	0.813	0.943	0.979
+SIGL	K	**0.140**	**0.816**	**4.234**	**0.212**	**0.818**	**0.945**	**0.980**
Wang [WBZ18b] PoseCNN+DDVO†	K	0.144	0.935	4.234	0.214	**0.827**	0.945	0.977
+SIGL	K	**0.139**	**0.808**	**4.180**	**0.209**	0.826	**0.948**	**0.980**
Zhou [ZBS17]	CS+K	0.190	1.436	4.975	0.258	0.735	0.915	0.968
Mahjourian [MWA18]	CS+K	0.151	0.949	4.383	0.227	0.802	0.935	0.974
Yin [YS18b] ResNet*	CS+K	/	/	/	/	/	/	/
+SIGL	CS+K	**0.141**	**0.837**	**4.160**	**0.209**	**0.823**	**0.947**	**0.980**
Wang [WBZ18b] PoseCNN+DDVO†	CS+K	0.142	0.901	4.202	0.213	0.827	0.946	0.978
+SIGL	CS+K	**0.135**	**0.832**	**4.119**	**0.206**	**0.836**	**0.949**	**0.980**

* Not available.

† Evaluated with prediction released by the author.

+SIGL: training with SIGL enabled

Table 3.3: *Training with monocular videos on KITTI.* K denotes the KITTI dataset, CS+K denotes pretraining on Cityscape and finetuning on KITTI. Depth values are capped at 50 and 80 meters. Our method consistently improves baseline algorithms.

Method	Category	Error metric				Accuracy ($\delta <$)		
		AbsRel	SqRel	RMSE	RMSElog	1.25	1.25^2	1.25^3
Godard [GMB17]	`/`	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Ours	`Human`	0.152	1.394	5.945	0.251	0.801	0.921	0.963
Ours	`Sky`	0.148	1.368	5.864	0.245	0.807	0.923	0.964
Ours	`Object`	0.146	1.335	5.986	0.249	0.800	0.920	0.963
Ours	`Nature`	0.146	1.292	5.826	0.247	0.804	0.923	0.964
Ours	`Vehicle`	0.143	1.304	5.797	0.241	0.814	0.927	0.966
Ours	`Construction`	0.142	1.252	5.729	0.240	0.810	0.928	0.967
Ours	`Flat`	0.141	1.270	5.779	0.239	0.814	0.927	0.966
Ours	`V+C+F`	0.139	1.211	5.702	0.239	0.816	0.928	0.966

Table 3.4: *Ablation study on KITTI.* Category denotes the semantic category that SIGL is applied. Deformable objects such as humans tend to yield poor performance (this is expected as deformable objects do not conform to our planar shape model). Object categories that can be easily modeled by planes tend to yield better performance. We see that the best combination of semantic classes to use are vehicle, construction and flat objects.

"vehicle" category provides reasonable improvement while the "nature" category (trees and hedges) helps a little. Applying our priors to the "human", "sky" and "object" categories does not consistently improve over the baseline, for the following reasons: "sky" does not have well-defined surface normals; "human" has deformable surfaces of which normals can point arbitrarily; "object" category consists of thin structures which project to few pixels rendering it hard to apply segmentation and our losses. The best is achieved when we apply our losses to "vehicle", "construction" and "flat" categories, denoted by `V+C+F` in Table 3.4.

3.5.5 Generalize to other datasets: Make3D

To showcase the generalizability of our approach, we follow the convention of [GMB17, ZBS17, YS18b, WBZ18b]: Our model trained *only* on KITTI Eigen split is directly tested on Make3D [SSN09]. Make3D contains 534 images with 2272 × 1707 resolution, of which 134 are used for testing.[3] Low resolution ground truth depths are given as 305 × 55 range

[3] Ideally we want to test on the whole Make3Dmethods to which we compare train on it. For a fair dataset since we do not train on Make3D, but othercomparison, we only use the 134 images for testing.

maps and must be resized and interpolated for evaluation. We follow [GMB17] and [ZBS17] in applying a central cropping to generate a 852 × 1707 crop centered on the image. We use the standard $C1$ evaluation metrics for Make3D and measure our performance on depths less than 70 meters. Table 3.5 shows a quantitative comparison to the competitors, both supervised and self-supervised, with two different training settings. Note that the results of [KLK12, LSL16, LRB16] are directly taken from [GMB17]. Since the exact cropping scheme used in [GMB17] is not available, we re-implemented it closely following the description in [GMB17]. We trained our model on KITTI Eigen split and compared against models of [GMB17, ZBS17, YS18b, WBZ18b] also trained on Eigen split (as provided by the authors) for a fair comparison.

A careful inspection of the baseline models (Godard [GMB17] in stereo and Yin [YS18b] in monocular supervision) versus ours reveals that the application of our losses does not hurt the generalizability of the baselines. Fig. 3.3 shows some qualitative results on Make3D. Though our model registers some failure cases in texture-less regions, a rough scene layout is present in the prediction. Regarding that the model is only trained on KITTI, of which the data modality is very different from that of Make3D, the prediction is sensible. But after all, a single image only affords to hypothesize depth, so we expect that any method using such predictions would have mechanisms to handle model deficiencies.

3.5.6 Evaluation on indoor datasets

To the best of our knowledge, none of the top-performing methods in self-supervised depth prediction have shown experimental results beyond planar motion, *i.e.,* driving scenarios such as KITTI and CityScapes, probably due to two reasons: Lack of rectified stereo pairs for training ([GMB17, ZGW18]) and difficulty to learn complex ego-motion along with depth prediction from video sequences ([ZBS17, YS18b, WBZ18b]).

However, with two modifications to the `GeoNet` model of Yin [YS18b] – a multi-task learning approach where ego-motion and depth prediction are jointly learned, we managed to train our model and outperform `GeoNet` on publicly available VISMA [FS18] dataset which

Method	Supervision	AbsRel	SqRel	RMSE	$\log_{10}$
TrainSetMean	Depth	0.893	15.517	11.542	0.223
Karsch [KLK12]	Depth	0.417	4.894	8.172	0.144
Liu [LSL16]	Depth	0.462	6.625	9.972	0.161
Laina [LRB16]	Depth	**0.198**	**1.665**	**5.461**	**0.082**
Godard [GMB17] `VGG`	Stereo	0.468	9.236	12.525	0.165
Ours	Stereo	**0.458**	**8.681**	**12.335**	**0.164**
Zhou [ZBS17]	Mono	0.407	5.367	11.011	0.167
Yin [YS18b] `ResNet`	Mono	0.376	4.645	10.350	0.152
Wang [WBZ18b] `PoseCNN+DDVO`	Mono	0.387	4.720	**8.09**	0.204
Ours	Mono	**0.356**	**4.517**	10.047	**0.144**

Table 3.5: *Generalizability test on Make3D.* Our models are only trained on KITTI Eigen split and tested on novel imagery from Make3D. We can see clear benefits in having a pose prior (bias from gravity) with even a simple shape model (e.g. planes).

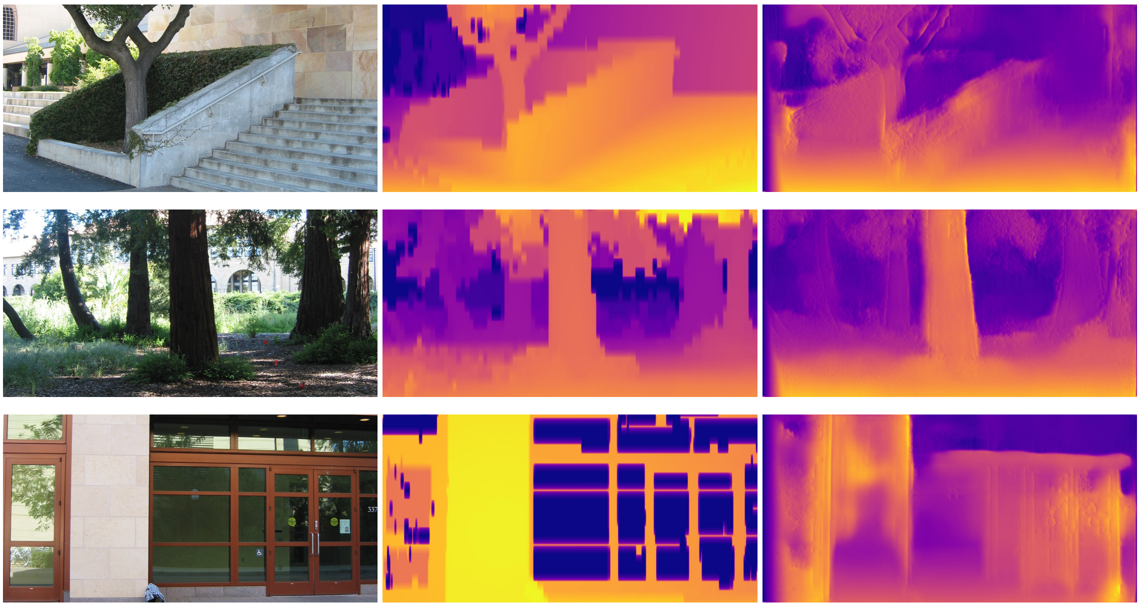

Figure 3.3: *Qualitative results on Make3D.* Left to right, each row shows an input RGB image, the corresponding ground truth disparity map and our prediction. Our model is *only* trained on KITTI and directly applied to Make3D.

Method	Error metric				Accuracy ($\delta <$)		
	AbsRel	SqRel	RMSE	RMSElog	1.25	1.25^2	1.25^3
`GeoNet`	0.204	0.157	0.518	0.250	0.702	0.914	0.975
`OursVIO`	0.154	0.111	0.446	0.211	0.796	0.940	0.983
`OursVIO++`	**0.149**	**0.105**	**0.421**	**0.202**	**0.820**	**0.947**	0.983

Table 3.6: *Quantitative results on VISMA validation.* We trained three models on the VISMA dataset. We see that the baseline model GeoNet has difficulty recovering the 3d scene. The drawback of applying generic (non-semantically informed) priors such as local smoothness is clearly visible. Upon applying our method, we see a significant performance boost. Exploiting priors in indoor scenes is particularly important as there are many textureless surfaces (e.g. walls).

features monocular videos of indoor scenes captured by a hand-held visual-inertial sensor platform under challenging motion. As a first modification, we replace the pose network in `GeoNet` with pose estimation from a VIO system [TCS15], which makes the network easier to train (we call this model `OursVIO`). Second, to further improve the quality of predicted depth maps, we impose our gravity-induced regularization terms to `OursVIO`, where gravity is also estimated online by VIO. Our second model is named `OursVIO++`.

VISMA dataset contains time-stamped monocular videos (30 Hz) from a PointGrey camera and inertial measurements (100 Hz) from an Xsens unit, which are used in both VIO and network training. RGB-D reconstructions (dense point clouds) of the same scenes from a Kinect are also available, along with the spatial alignment $g_{\text{VIO}\leftarrow\text{RGBD}} \in \text{SE}(3)$ from RGB-D to VIO provided by the author. To get ground truth depth for cross-modality validation, we apply $g_{\text{VIO}\leftarrow\text{RGBD}}$ to the dense point clouds which are then projected to the PointGrey video frames. PSPNet trained on ADE20K [ZZP17] produces segmentation masks for training.[4] Of the $10K$ frames in VISMA, we remove static ones and construct 3-frame sequences

[4] Among the 91 categories in ADE20K which PSPNet is trained on, we select “floor”, “ceiling”, “wall”, “window”, “door”, “building”, “chair”, “cabinet”, “desk”, “table” to apply our losses.

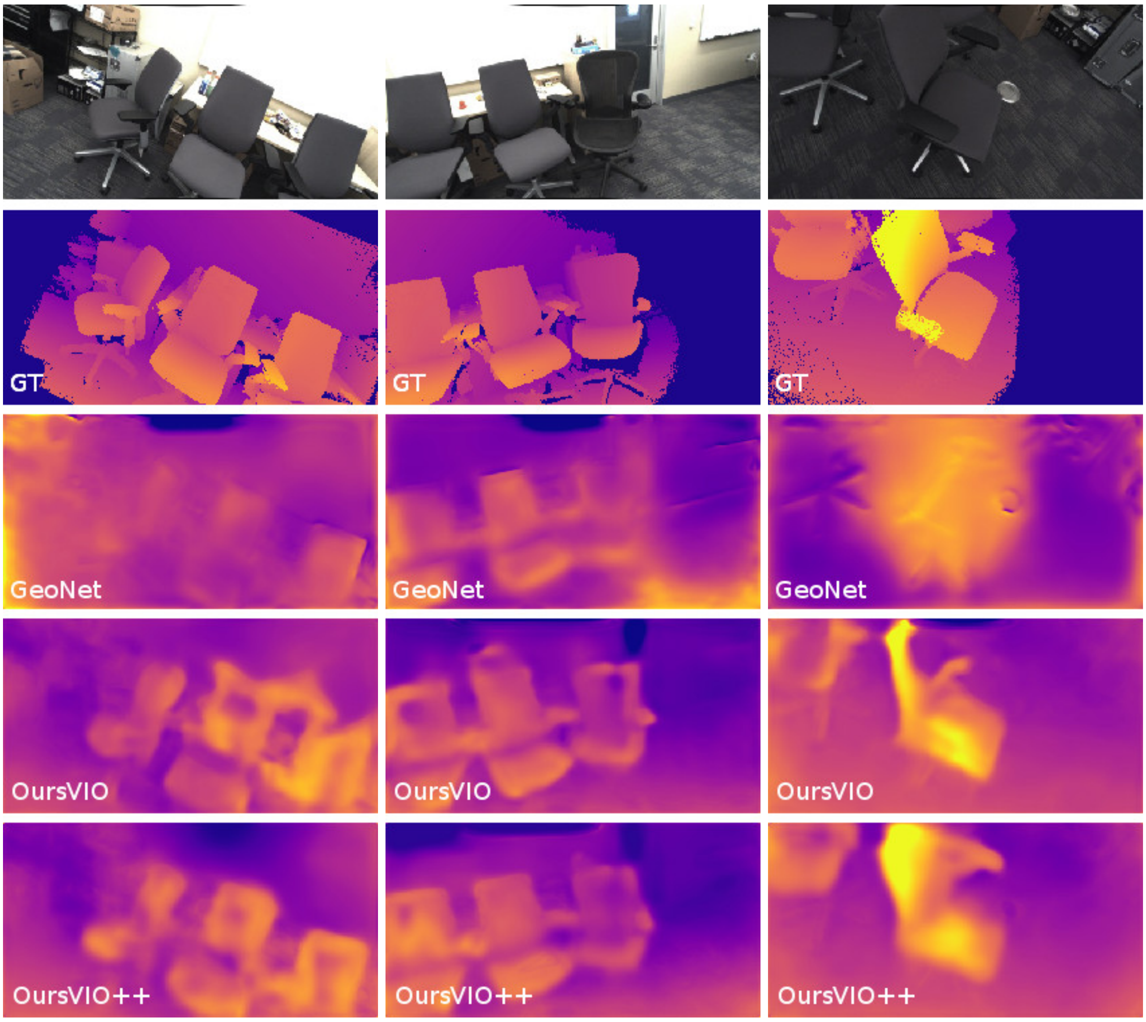

Figure 3.4: *Qualitative comparison on VISMA validation.* Top to bottom, each column shows an input RGB image, the corresponding ground truth inverse depth map, results of `GeoNet` (baseline), `OursVIO`, and `OursVIO++`. Both `OursVIO` and `OursVIO++` show largely improved results over the baseline, especially for images captured at extreme viewpoint (large in-plane rotation and top-down view). `OursVIO++` (with gravity-induced priors) further improves over `OursVIO` (without priors) at planar regions, *e.g.*, the chair backs, where holes have been filled.

(triplet) which are five frames apart in the original video to ensure sufficient parallax, resulting $8,511$ triplets in total. We randomly sample 100 triplets for validation and use the rest for training. Fig. 3.4 and Table 3.6 show comparisons of `GeoNet`, `OursVIO` and `OursVIO++`, all trained from scratch on VISMA until validation error stops decreasing. Both `OursVIO` and `OursVIO++` improve over the baseline model by a large margin. Moreover, `OursVIO++` trained with our gravity-induced losses has the capability to further refine results of `OursVIO` trained without our losses.

3.6 Discussion

Gravity informs the shape of objects populating the scene, which is a powerful prior to visual scene analysis. We have presented a simple illustration of this power by adding a prior to standard monocular depth prediction methods that biases the normals of surfaces of known classes to align to gravity or its complement. Far more can be done: While in this work we use known biases in the shape of certain object classes, such as the fact that roads tend to be perpendicular to gravity, in the future we could learn such biases directly.

CHAPTER 4

Dense Depth Posterior from Single Image and Sparse Range

4.1 Introduction

There are many dense depth maps that are compatible with a given image and a sparse point cloud. Any point-estimate, therefore, depends critically on the prior assumptions made. Ideally, one would compute the entire posterior distribution of depth maps, rather than a point-estimate. The posterior affords to reason about confidence, integrating evidence over time, and in general, is a (Bayesian) sufficient representation that accounts for all the information in the data.

Motivating application. In autonomous navigation, a sparse point cloud from lidar may be insufficient to make planning decisions: Is the surface of the road in Fig. 4.1 (middle, better viewed when enlarged) littered with pot-holes, or is it a smooth surface? Points that are nearby in image topology, projecting onto adjacent pixels, may be arbitrarily far in the scene. For instance, pixels that straddle an occluding boundary correspond to large depth gaps in the scene. While the lidar may not measure every pixel, if we know it projects onto a tree, trees tend to stand out from the ground, which informs the topology of the scene. On the other hand, pixels that straddle illumination boundaries, like shadows cast by trees, seldom correspond to large depth discontinuities.

Depth completion is the process of assigning a depth value to each pixel. While there are several deep learning-based methods to do so, we wish to have the entire posterior estimate over depths. Sparse range measurements serve to ground the posterior estimate in a metric space. This could then be used by a decision and control engine downstream.

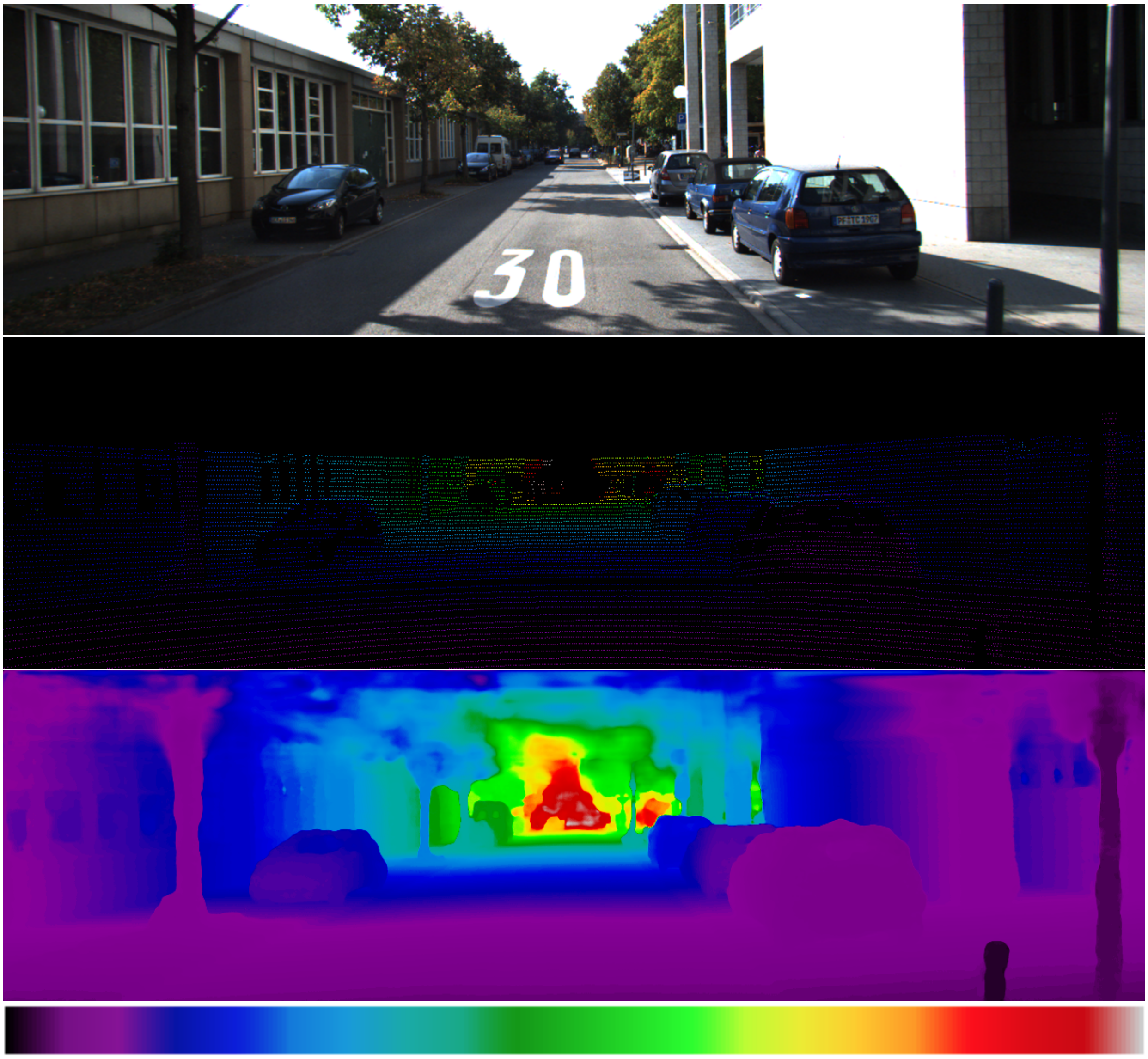

Figure 4.1: *Example of our depth completion method.* An image (top) is insufficient to determine the geometry of the scene; a point cloud alone (middle) is similarly ambiguous. Lidar returns are shown as colored points, but black regions are uninformative: Are the black regions holes in the road surface, or due to radiometric absorption? Combining a single image, the lidar point cloud, and previously seen scenes allows inferring a dense depth map (bottom) with high confidence. Color bar from left to right: zero to infinity.

Side information. If the dense depth map is obtained by processing the given image and sparse point cloud alone, the quality of the resulting decision or control action could be no

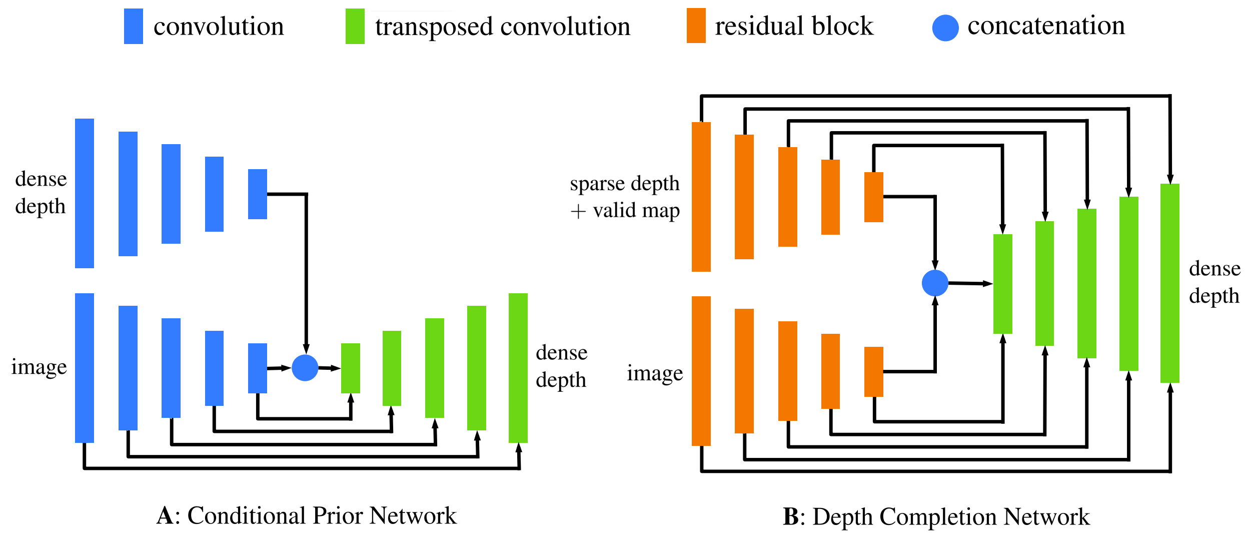

Figure 4.2: *Our network architecture.* (A): the architecture of the Conditional Prior Network (CPN) to learn the conditional of the dense depth given a single image. (B): Our proposed Depth Completion Network (DCN) for learning the mapping from a sparse depth map and an image to a dense depth map. Connections within each encoder/decoder block are omitted for simplicity.

better than if the raw data was fed downstream (Data Processing Inequality). However, if depth completion can exploit a prior or aggregate experience from previously seen images and corresponding dense depth maps, then it is possible for the resulting dense depth map to improve the quality of the decision or action, assuming that the training set is representative. To analyze a depth completion algorithm, it is important to understand what prior assumptions, hypotheses or side information is being exploited.

Goal. We seek methods to *estimate the geometry and topology of the scene given an image, a sparse depth map, and a body of training data consisting of images and the associated dense depth maps.* Our assumption is that the distribution of seen images and corresponding depth maps is representative of the present data (image and sparse point cloud) once restricted to a sparse domain.

Our method yields the full posterior over depth maps, which is much more powerful than any point estimate. For instance, it allows reasoning about confidence intervals. We elect the simplest point estimate possible, which is the maximum, to evaluate the accuracy of the posterior. It should be noted, however, that when there are multiple hypotheses with similar posterior, the point estimate could jump from one mode to another, and yet the posterior being an accurate representation of the unknown variable. More sophisticated point estimators, for instance, taking into account memory, or spatial distribution, non-maximum suppression, etc. could be considered, but here we limit ourselves to the simplest one.

Key idea. While an image alone is insufficient to determine a depth map, certain depth maps are more probable than others given the image and a previously seen dataset. The key to our approach is a conditional prior model $P(d|I, \mathcal{D})$ that scores the compatibility of each dense depth map d with the given image I based on the previously observed dataset $\mathcal{D}$. This is computed using a Conditional Prior Network (CPN) [YS18a] in conjunction with a model of the likelihood of the observed sparse point cloud z under the hypothesized depth map d, to yield the posterior probability and, from it, a maximum a-posteriori (MAP) estimate of the depth map for benchmark evaluation:

$$\hat{d} = \arg\max_{d} P(d|I, z) \propto P(z|d) P_{\mathcal{D}}(d|I). \tag{4.1}$$

Let $D \subset \mathbb{R}^2$ be the image domain, sampled on a regular lattice of dimension $N \times M$, $I : D \to \mathbb{R}^3$ is a color image, with the range quantized to a finite set of colors, $d : D \to \mathbb{R}+$ is the dense depth map defined on the lattice D, which we represent with an abuse of notation as a vector of dimension MN: $d \in \mathbb{R}_+^{NM}$. $\Omega \subset D$ is a sparse subset of the image domain, with cardinality $K = |\Omega|$, where the function d takes values $d(\Omega) = z \in \mathbb{R}_+^K$. Finally, $\mathcal{D} = \{d_j, I_j\}_{j=1}^n$ is a dataset of images I_j and their corresponding dense depth maps $d_j \in \mathbb{R}_+^{NM}$. Since we do not treat $\mathcal{D}$ as a random variable but a given set of data, we write it as a subscript. In some cases, we may have additional data available during training, for instance stereo imagery, in which case we include it in the dataset, and discuss in detail how to exploit it in Sect. 4.3.3.

Results. We train a deep neural network model to produce an estimate of the posterior distribution of dense depth maps given an image and a sparse point cloud (sparse range map), that leverages a Conditional Prior Network to restrict the hypothesis space, weighted by a classical likelihood term. We use a simple maximum a-posteriori (MAP) estimate to evaluate our approach on benchmark datasets, including the KITTI-unsupervised, where the dense depth map is predicted given an image and a point cloud with 5% pixel coverage, and the KITTI-supervised, where a point cloud with 30% coverage is given for training. We achieve top performance in both. We also validate on additional data in the Supplementary Materials [YWS19].

4.2 Related Work

Semi-Dense Depth Completion. Structured light sensors typically provide dense depth measurements with about 20% missing values; At this density, the problem is akin to inpainting [CS12, LRL14, SC13] that use morphological operations [KHW18, PGA16]. The regime we are interested in involves far sparser point clouds ($>$ 90% missing values).

Supervised Depth Completion. Given a single RGB image and its associated sparse depth measurements along with dense ground truth, learning-based methods [EFK18, HFY18, RPY18, USS17, ZF18] minimize the corresponding loss between prediction and ground truth depth. [USS17] trains a deep network to regress depth using a sparse convolutional layer that discounts the invalid depth measurements in the input while [HFY18] proposes a sparsity-invariant upsampling layer, sparsity-invariant summation, and joint sparsity-invariant concatenation and convolution. [EFK18] treats the binary validity map as a confidence map and adapts normalized convolution for confidence propagation through layers. [DVP18] implements an approximation of morphological operators using the contra-harmonic mean (CHM) filter [MAS13] and incorporates it as a layer in a U-Net architecture for depth completion. [CWL18] proposes a deep recurrent auto-encoder to mimic the optimization procedure of compressive sensing for depth completion, where the dictionary is embedded in the neural network. [ZF18] predicts surface normals and occlusion boundaries from the RGB image, which gives a coarse representation of the scene structure. The predicted surface normals and occlusion boundaries are incorporated as constraints in a global optimization framework guided by sparse depth.

Unsupervised Depth Completion. In this problem setting, dense ground truth depth is not available as supervision, so a strong prior is key. [MCK19] proposes minimizing the photometric consistency loss among a sequence of images with a second-order smoothness prior based on a similar formulation in single image depth prediction [MWA18, WBZ18a, ZBS17]. Instead of having a separate pose network or using direct visual odometry methods, [MCK19] uses Perspective-n-Point (PnP) [LMF09] and Random Sample Consensus (RANSAC) [FB81] to obtain pose. We exploit recently introduced method to learn the conditional prior [YS18a] to take into account scene semantics rather than using a local smoothness assumption.

Stereo as Supervision. Recent works in view synthesis [FNP16, XGF16] and unsupervised single image depth prediction, [FWS19, GBC16, GMB17, WHS19] propose using view synthesis to hallucinate a novel view image by reconstruction loss. In the case of stereo pairs, [GBC16, GMB17, WHS19] propose training networks to predict the disparities of an input image by reconstructing the unseen right view of a stereo pair given the left. In addition to the photometric reconstruction loss, local smoothness is assumed; [GMB17] additionally proposed edge-aware smoothness and left-right consistency. Although during inference, we assume only one image is given, at training time we may have stereo imagery available, which we exploit as in Sect. 4.3.3. In this work, we incorporate only the stereo photometric reconstruction term. Despite our network predicting depths and the network [GBC16, GMB17, WHS19] predicting disparities, we are able to incorporate this training scheme seamlessly into our approach.

Exploiting Semantics and Contextual Cues. While methods [EFK18, HFY18, MCK19, RPY18, USS17, ZF18] learn a representation for the depth completion task through ground truth supervision, they do not have any explicit modeling of the semantics of the scene. Recently, [SSP16] explored this direction by predicting object boundary and semantic labels through a deep network and using them to construct locally planar elements that serve as input to a global energy minimization for depth completion. [CWY18] proposes to complete the depth by anisotropic diffusion with a recurrent convolution network, where the affinity matrix is computed locally from an image. [JCW18] also trains a U-Net for joint depth completion and semantic segmentation in the form of multitask learning in an effort to incorporate semantics in the learning process.

To address contextual cues and scene semantics, [YS18a] introduces a Conditional Prior Network (CPN) in the context of optical flow, which serves as a learning scheme for inferring the distribution of optical flow vectors given a single image. We leverage this technique and formulate depth completion as a maximum a-posteriori problem by factorizing it into a likelihood term and a conditional prior term, making it possible to explicitly model the semantics induced regularity of a single image. Even though our method could be applied to sparse-to-dense interpolation for optical flow, where the sparse matches can be obtained using [YS17, YLS15], here we focus our test on depth completion task.

4.3 Method

In order to exploit a previously observed dataset $\mathcal{D}$, we use a Conditional Prior Network (CPN) [YS18a] in our framework. Conditional Prior Networks infer the probability of an optical flow given a single image. During training, ground truth optical flow is encoded (upper branch in Fig. 4.2-A), concatenated with the encoder of an image (lower branch), and then decoded into a reconstruction of the input optical flow.

In our implementation, the upper branch encodes dense depth, concatenated with the encoding of the image, to produce a dense reconstruction of depth at the decoder, together with a normalized likelihood that can serve as a posterior score. We consider a CPN as a function that, given an image (lower branch input) maps any sample putative depth map (upper branch input) to a positive real number, which represents the conditional probability/prior of the input dense depth map given the image.

We denote the ensemble of parameters in the CPN as w^{CPN}; with abuse of notation, we denote the decoded depth with $d' = w^{CPN}(d, I)$. When trained with a bottleneck imposed on the encoder (upper branch), the reconstruction error is proportional to the conditional distribution:

$$Q(d, I; w^{CPN}) = e^{-\|w^{CPN}(d,I)-d\|^{\eta}} \propto P_{\mathcal{D}}(d|I) \tag{4.2}$$

where η indicates the specific norm used for calculating Q. In Sect. 4.4.2 and Sect. 4.5, we show the training details of CPN, and also quantitatively show the effect of different choices of the norm η. In the following, we assume w^{CPN} is trained, and Q will be used as the conditional prior. For the proof that Q computed by CPN represents the conditional prior as in Eq. (4.2), please refer to [YS18a].

In order to obtain a posterior estimate of depth, the CPN needs to be coupled with a likelihood term.

4.3.1 Supervised Single Image Depth Completion

Supervised learning of dense depth assumes the availability of ground truth dense depth maps. In the KITTI depth completion benchmark [USS17], these are generated by accumulating the neighboring sparse lidar measurements. Even though it is called ground truth, the density is only $\sim 30\%$ of the image domain, whereas the density of the unsupervised benchmark is $\sim 5\%$. The training loss in the supervised modality is just the prediction error:

$$L(w) = \sum_{j=1}^{N} \|\phi(z_j, I_j; w) - d_j\|^{\gamma} \tag{4.3}$$

where ϕ is the map from sparse depth z and image I to dense depth, realized by a deep neural network with parameters w, and $\gamma = 1$ fixed in the supervised training.

Our network structure for ϕ is detailed in Fig. 4.2-B, which has a symmetric two-branch structure, each encoding different types of input: one sparse depth, the other an image; skip connections are enabled for two branches. Note that our network structure is unique among all the top performing ones on the KITTI depth completion benchmark: We do not use specifically-designed layers for sparse inputs, such as sparsity invariant layers [HFY18, USS17]. Instead of early fusion of sparse depth and image, our depth defers fusion to decoding, which entails fewer learnable parameters, detailed in [YWS19]. A related idea is proposed in [JCW18]; instead of a more sophisticated NASNet block [ZVS18], we use the more common ResNet block [HZR16]. Although simpler than competing methods, our network achieves state-of-the-art performance (Sect. 4.5).

4.3.2 Unsupervised Single Image Depth Completion

Supervised learning requires ground truth dense depth, which is hard to come by. Even the "ground truth" provided in the KITTI benchmark is only 30% dense and interpolated from even sparser maps. When only sparse independent measurements of depth are available, for instance from lidar, with less than 10% coverage (e.g. 5% for KITTI), we call depth completion *unsupervised* as the only input are sensory data, from images and a range measurement device, with no annotation or pre-processing of the data.

The key to our approach is the use of a CPN to score the compatibility of each dense depth map d with the given image I based on the previously observed data $\mathcal{D}$. In some cases, we may have additional sensory data available *during training*, for instance, a second image taken with a camera with a known relative pose, such as stereo. In this case, we include the reading from the second camera in the training set $\mathcal{D}$, as described in Sect. 4.3.3. When only a single image is given, the CPN Eq. (4.2) is combined with a model of the likelihood of the observed sparse point cloud z under the hypothesized depth map d:

$$P(z|d) \propto \exp(-\|z - d(\Omega)\|^{\gamma}) \tag{4.4}$$

which is simply a Gaussian around the hypothesized depth, restricted to the sparse subset Ω, when $\gamma = 2$. The overall loss is:

$$\begin{aligned} L^u(w) &= -\sum_{j=1}^{N} logP(d_j|I_j, z_j, \mathcal{D}) \\ &= \sum_{j=1}^{N} \|z_j - d_j(\Omega)\|^{\gamma} + \alpha \sum_{j=1}^{N} \|w^{CPN}(d_j, I_j) - d_j\|^{\eta} \\ &= \sum_{j=1}^{N} \|z_j - \phi(z_j, I_j; w)(\Omega)\|^{\gamma} + \alpha \sum_{j=1}^{N} \|w^{CPN}(\phi(z_j, I_j; w), I_j) - \phi(z_j, I_j; w)\|^{\eta} \end{aligned} \tag{4.5}$$

Note that γ, η control the actual norm used during training, as well as the modeling of the likelihood and conditional distribution. We experiment with these parameters in Sect. 4.5.1, and show our quantitative analysis there.

4.3.3 Disparity Supervision

Some datasets come with stereo imagery. We want to be able to exploit it, but without having to require its availability at inference time. We exploit the strong relation between depth and disparity. In addition to the sparse depth z and the image I, we are given a second image I' as part of a stereo pair, which is rectified (standard pre-processing), to first-order we assume that there exists a displacement $s = s(x), x \in D$ such that

$$I(x) \approx I'(x + s) \tag{4.6}$$

which is the intensity constancy constraint. We model, again simplistically, disparity s as $s = FB/d$, where F is the focal length and B is the baseline (distance between the optical centers) of the cameras. Hence, we can synthesize disparity s from the predicted dense depth d, thus to constrain the recovery of 3-d scene geometry. More specifically, we model the likelihood of seeing I' given I, d as:

$$P(I'|I, d) \propto \exp(-\frac{\sum_x \|I(x) - I'(x + s(d(x)))\|}{\delta^2}) \tag{4.7}$$

However, the validity of the intensity constancy assumption is affected by complex phenomena such as translucency, transparency, inter-reflection, etc. In order to mitigate the error in the assumption, we could also employ a perceptual metric of structural similarity (SSIM) [WBS04]. SSIM scores corresponding 3×3 patches $p(x), p'(x) \in \mathbb{R}_+^{3\times3}$ centered at x in I and I', respectively, to measure their local structural similarity. Higher scores denote more similarity; hence we can subtract the scores from 1 to form a robust version of Eq. (4.7). We use $P_{raw}(I'|I, d)$ and $P_{ssim}(I'|I, d)$ to represent the probability of I' given I, d measured in raw photometric value and SSIM score respectively. When the stereo pair is available, we can form the conditional prior as follows by applying conditional independence:

$$\begin{aligned} P(d|I, I', \mathcal{D}) &\propto P(I'|I, d, \mathcal{D})P(d|I, \mathcal{D}) \\ &= P(I'|I, d)P_{\mathcal{D}}(d|I) \end{aligned} \tag{4.8}$$

Similar to the training loss Eq. (4.5) for the unsupervised single image depth completion

Method	iRMSE	iMAE	RMSE	MAE	Rank
Dimitrievski [DVP18]	3.84	1.57	1045.45	310.49	13.0
Cheng [CWY18]	2.93	1.15	1019.64	279.46	7.5
Huang [HFY18]	2.73	1.13	841.78	253.47	6.0
Ma [MCK19]	2.80	1.21	**814.73**	249.95	5.5
Eldesokey [EFK18]	2.60	1.03	829.98	233.26	4.75
Jaritz [JCW18]	2.17	0.95	917.64	234.81	3.0
Ours	**2.12**	**0.86**	836.00	**205.40**	**1.5**

Table 4.1: *Quantitative results on the supervised KITTI depth completion benchmark.* Our method achieves state of the art performance in three metrics, iRMSE, iMAE, and MAE. [MCK19] performs better than us by 2.6% on the RMSE metric; however, we outperform [MCK19] on all other metrics by 24.3%, 28.9% and 17.8% on the iRMSE, iMAE and MAE, respectively. The last column is the average rank over ranks on all the four metrics.

setting, we can derive the loss for the stereo setting as follows:

$$\begin{aligned} L^s(w) &= -\sum_{j=1}^{N} logP(d_j|I_j, I'_j, z_j, \mathcal{D}) \\ &= L^u(w) + \beta \sum_{j,x} \|I_j(x) - I'_j(x + s(d_j(x)))\| \end{aligned} \tag{4.9}$$

where $d_j = \phi(z_j, I_j; w)$ and L^u is the loss defined in Eq. (4.5). Note that, the above summation term is the instantiation for $P_{raw}(I'|I, d)$, which can also be replaced by the SSIM counterpart. Rather than choosing one or the other, we compose the two with tunable parameters β_c and β_s, our final loss for stereo setting depth completion is:

$$L^s(w) = L^u(w) + \beta_c\psi_c + \beta_s\psi_s \tag{4.10}$$

with ψ_c represents the raw intensity summation term in Eq. (4.9), and ψ_s for the SSIM counterpart. Next, we elaborate our implementation details and evaluate the performance of our proposed method in different depth completion settings.

4.4 Implementation Details

4.4.1 Network architecture

We modify the public implementation of CPN [YS18a] by replacing the input of the encoding branch with a dense depth map. Fusion of the two branches is simply a concatenation of the encodings. The encoders have only convolutional layers, while the decoder is made of transposed convolutional layers for upsampling.

Our proposed network, unlike the base CPN, as seen in Fig. 4.2-A, contains skip connections between the layers of the depth encoder and the corresponding decoder layers, which makes the network symmetric. We also use ResNet blocks [HZR16] in the encoders instead of pure convolutions. A stride of 2 is used for downsampling in the encoder and the number of channels in the feature map after each encoding layer is $[64*k, 128*k, 256*k, 512*k, 512*k]$. In all our experiments, we use $k = 0.25$ for the depth branch, and $k = 0.75$ for the image branch, taking into consideration that an RGB image has three channels while depth map only has one channel. Our network has fewer parameters than those based on early fusion (e.g. [MCK19] used ≈27.8M parameters in total; where as we only use ≈18.8M). We provide an example comparing our network architecture and that of [MCK19] in the Supplementary Materials [YWS19].

4.4.2 Training Procedure

We begin by detailing the training procedure for CPN. Once learned, we apply CPN as part of our training loss and do not need it during inference. In order to learn the conditional prior of the dense depth maps given an image, we require a dataset with images and corresponding dense depth maps. We are unaware of any real-world dataset for outdoor scenes that meets our criterion. Therefore, we train the CPN using the Virtual KITTI dataset [GWC16]. It contains 50 high-resolution monocular videos with a total of $21,260$ frames, together with ground truth dense depth maps, generated from five different virtual worlds under different lighting and weather conditions. The original Virtual KITTI image has a large resolution

	Validation Set		Test Set			
Loss	RMSE	MAE	iRMSE	iMAE	RMSE	MAE
Ma [MCK19]	1384.85	358.92	4.07	1.57	1299.85	350.32
L^u	1325.79	355.86	3.69	1.37	1285.14	353.16
$L^s(\psi_c)$	1320.26	353.24	3.63	1.34	1274.65	349.88
$L^s(\psi_c, \psi_s)$	**1310.03**	**347.17**	**3.58**	**1.32**	**1263.19**	**343.46**

Table 4.2: *Quantitative results on the unsupervised KITTI depth completion benchmark.* Our baseline approach using CPN as a regularizer outperforms [MCK19] on the iRMSE, iMAE and RMSE metrics on the test set, whereas [MCK19] marginally performs better than us on MAE by 0.8%. We note that [MCK19] achieves this performance using photometric supervision. When including our photometric term (Eq. (4.10)), we outperform [MCK19] on every metric and achieve state-of-the-art performance.

of 1242 × 375, which is too large to feed into a normal commercial GPU. So we crop it to 768 × 320 and use a batch size of 4 for training. The initial learning rate is set to $1e^{-4}$, and is halved every 50,000 steps 300,000 steps in total.

We implement our approach using TensorFlow [ABC16]. We use Adam [KB14] to optimize our network with the same batch size and learning rate schedule as the training of CPN. We apply histogram equalization and also randomly crop the image to 768 × 320. We additionally apply random flipping both vertical and horizontal to prevent overfitting. In the case of unsupervised training, we also perform a random shift within a 3 × 3 neighborhood to the sparse depth input and the corresponding validity map. We use $\alpha = 0.045$, $\beta = 1.20$ for Eq. (4.9), and the same α is applied with $\beta_c = 0.15$, $\beta_s = 0.425$ for Eq. (4.10). We choose $\gamma = 1$ and $\eta = 2$, but as one may notice in Eq. (4.2), the actual conditional prior also depends on the choice of the norm η. To show the reasoning behind our choice, we will present as an empirical study in Fig. 4.3 to show the effects of the different pairing of norms with a varying α by evaluating each model on the RMSE metric.

In the next section, we report representative experiments in both the supervised and

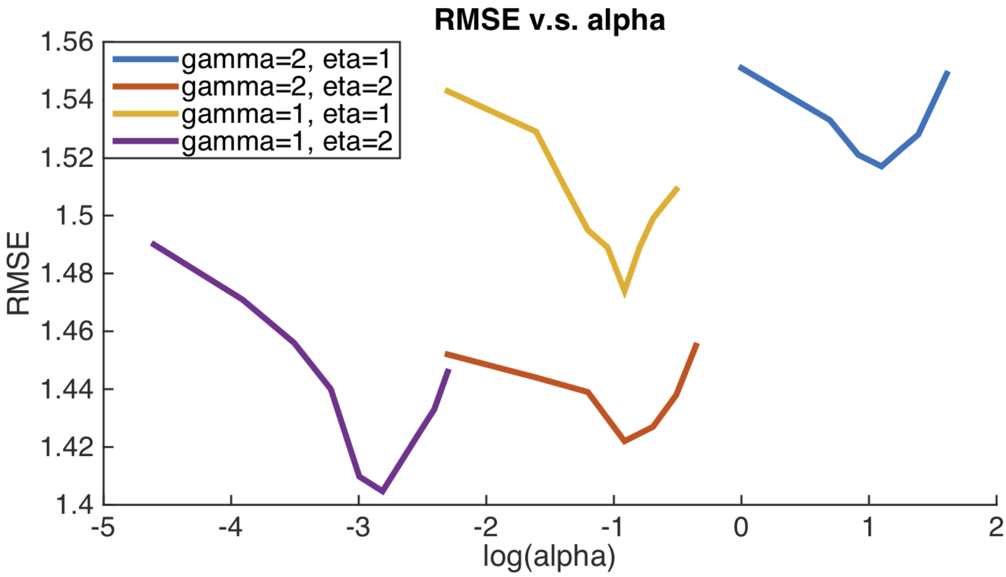

Figure 4.3: *A study on the choice of* γ. This plot shows the empirical study on the choice of norms γ, η in the likelihood term and the conditional prior term respectively. Each curve is generated by varying α in Eq. (4.5) with fixed γ, η. And the performance is measured in RMSE.

unsupervised benchmarks.

4.5 Experiments

We evaluate our approach on the KITTI depth completion benchmark [USS17]. The dataset provides $\sim 80k$ raw image frames and corresponding sparse depth maps. The sparse depth maps are the raw output from the Velodyne lidar sensor, each with a density of about 5%. The ground truth depth map is created by accumulating the neighboring 11 raw lidar scans, with roughly 30% pixels annotated. We use the officially selected 1,000 samples for validation and we apply our method to 1,000 testing samples, with which we submit to the official KITTI website for evaluation. We additionally perform an ablation study on the effects of the sparsity of the input depth measurements on the NYUv2 indoor dataset [SHK12] in the Supplementary Materials [YWS19].

4.5.1 Norm Selection

As seen in Eq. (4.5), γ, η control the actual norms (penalty functions) applied to the likelihood term and conditional prior term respectively, which in turn determine how we model the distributions. General options are from the binary set $\{1, 2\}$. i.e. $\{\mathcal{L}_1, \mathcal{L}_2\}$, however, there is currently no agreement on which one is better suited for the depth completion task. [MCK19] shows $\gamma = 2$ gives significant improvement for their network, while both [USS17, JCW18] claim to have better performance when $\gamma = 1$ is applied. In our approximation of the posterior in Eq. (4.5), the choice of the norms gets more complex as the modeling (norm) of the conditional prior will also depend on the likelihood model. Currently, there is no clear guidance on how to make the best choice, as it may also depend on the network structure. Here we try to explore the characteristic of different norms, at least for our network structure, by conducting an empirical study on a simple version (channel number of features reduced) of our depth completion network using different combinations of γ and η. As shown in Fig. 4.3, the performance on the KITTI depth completion validation set varies in a wide range with different γ, η. Clearly for our depth completion network, $\mathcal{L}_1$ is always better than $\mathcal{L}_2$ on the likelihood term. And the lowest RMSE is achieved when a $\mathcal{L}_2$ is also applied on the conditional prior term. Thus the best coupling is $\gamma = 1, \eta = 2$ for Eq. (4.5).

4.5.2 Supervised Depth Completion

We evaluate the proposed Depth Completion Network described in Sect. 4.3.1 on the KITTI depth completion benchmark. We show a quantitative comparison between our approach and the top performers on the benchmark in Tab. 4.1. Our approach achieves the state-of-the-art in three metrics by outperforming [EFK18, JCW18], who each held the state-of-the-art in different metrics on the benchmark. We improve over [JCW18] in iRMSE and iMAE by 2.3% and 9.5%, respectively, and [EFK18] in MAE by 11.9%. [MCK19] performs better on the RMSE metric by 2.6%; however, we outperform [MCK19] by 24.3%, 28.9% and 17.8% on the iRMSE, iMAE and MAE metrics, respectively. Note in the online table of KITTI

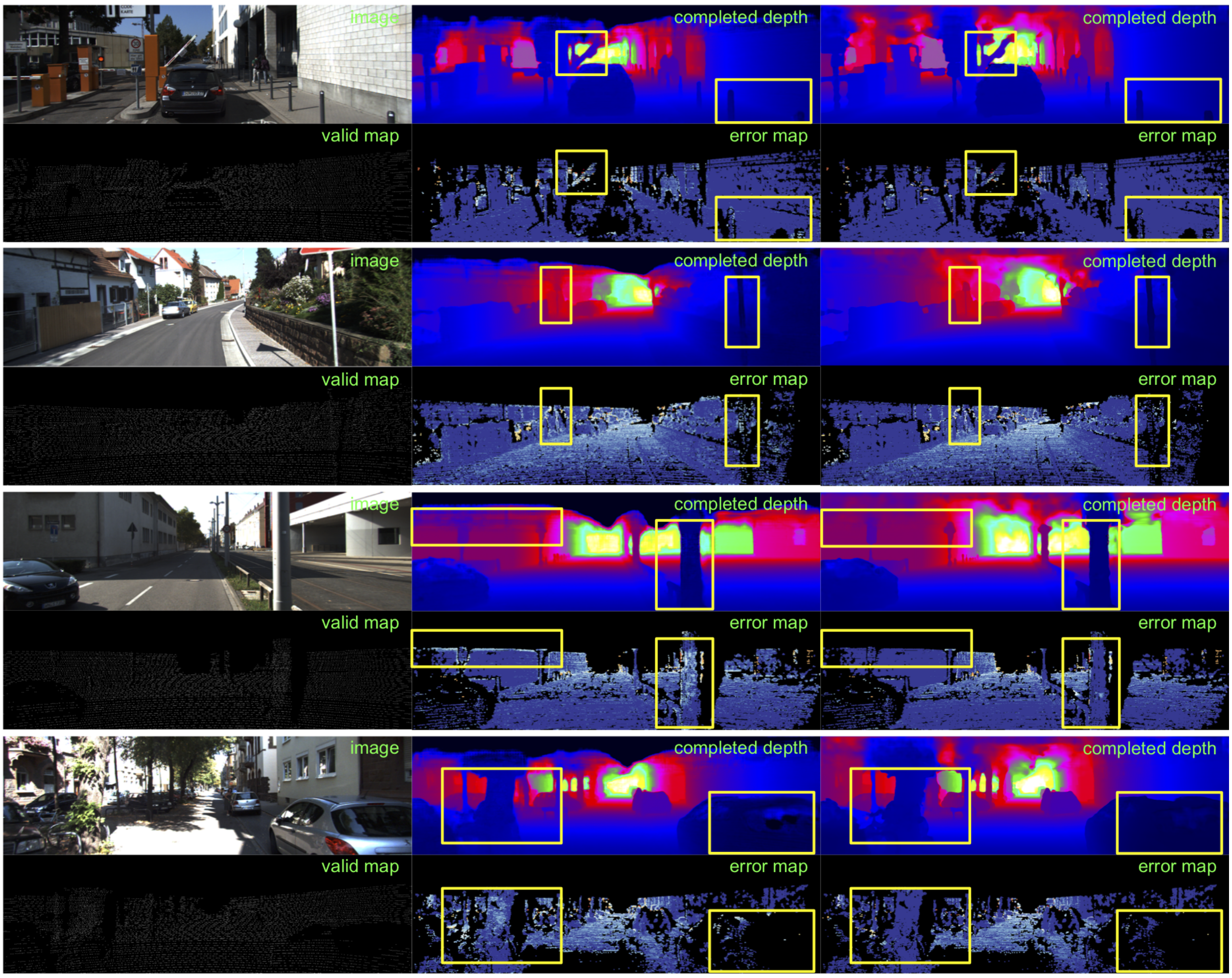

Figure 4.4: *Qualitative comparison to Ma et al. [MCK19] on KITTI depth completion test set in the supervised setting.* Image and validity map of the sparse measurements (1st column), dense depth results and corresponding error map of [MCK19] (2nd column) and our results and error map (3rd column). Warmer color in the error map denotes higher error. The yellow rectangles highlight the regions for detailed comparison. Note that our network consistently performs better on fine and far structures and our completed dense depth maps have less visual artifacts.

depth completion benchmark[1], all methods are solely ranked by the RMSE metric, which may not fully reflect the performance of each method. Thus we propose to rank all methods

[1] `http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_completion`

by averaging over the rank numbers on each metric, and the overall ranking is shown in the last column of Tab. 4.1. Not surprisingly, our depth completion network gets the smallest rank number due to its generally good performance on all metrics.

Fig. 4.4 shows a qualitative comparison of our method to the top performing method on the test set of the KITTI benchmark. We see that our method produces depths that are more consistent with the scene with fewer artifacts (e.g. grid-like structures [MCK19], holes in objects [EFK18]). Also, our network performs consistently better on fine and far structures, which may be traffic signs and poles on the roadside that provide critical information for safe driving as shown in the second row in Fig. 4.4. More in the Supplementary [YWS19].

4.5.3 Unsupervised Depth Completion

We show that our network can also be applied to unsupervised setting using only the training loss Eq. (4.5) to achieve the state-of-the-art results as well. We note that the simplest way for the network to minimize the data term is to directly copy the sparse input to the output, which will make the learning inefficient. To facilitate the training, we change the stride of the first layer from 1 to 2 and replace the final layer of the decoder with a nearest neighbor upsampling.

We show a quantitative comparison (Tab. 4.2) between our method and that of [MCK19] along with an ablation study on our loss function. We note that the results of [MCK19] are achieved using their full model, which includes their multi-view photometric term. Our approach using just Eq. (4.5) is able to outperform [MCK19] in every metric with the exception of MAE where [MCK19] marginally beats us by 0.8%. By applying our reconstruction loss Eq. (4.9), we outperform [MCK19] in every metric. Moreover, our full model Eq. (4.10) further improves over all other variants and is state-of-the-art in unsupervised depth completion. We present a qualitative comparison between our approach and that of [MCK19] in Fig. 4.5. Visually, we observe the results of [MCK19] still contain the artifacts as seen before. The artifacts, i.e. circles, as detailed in Fig. 4.5, are signs that their network is probably overfitted to the input sparse depth, due to the lack of semantic regularity. Our

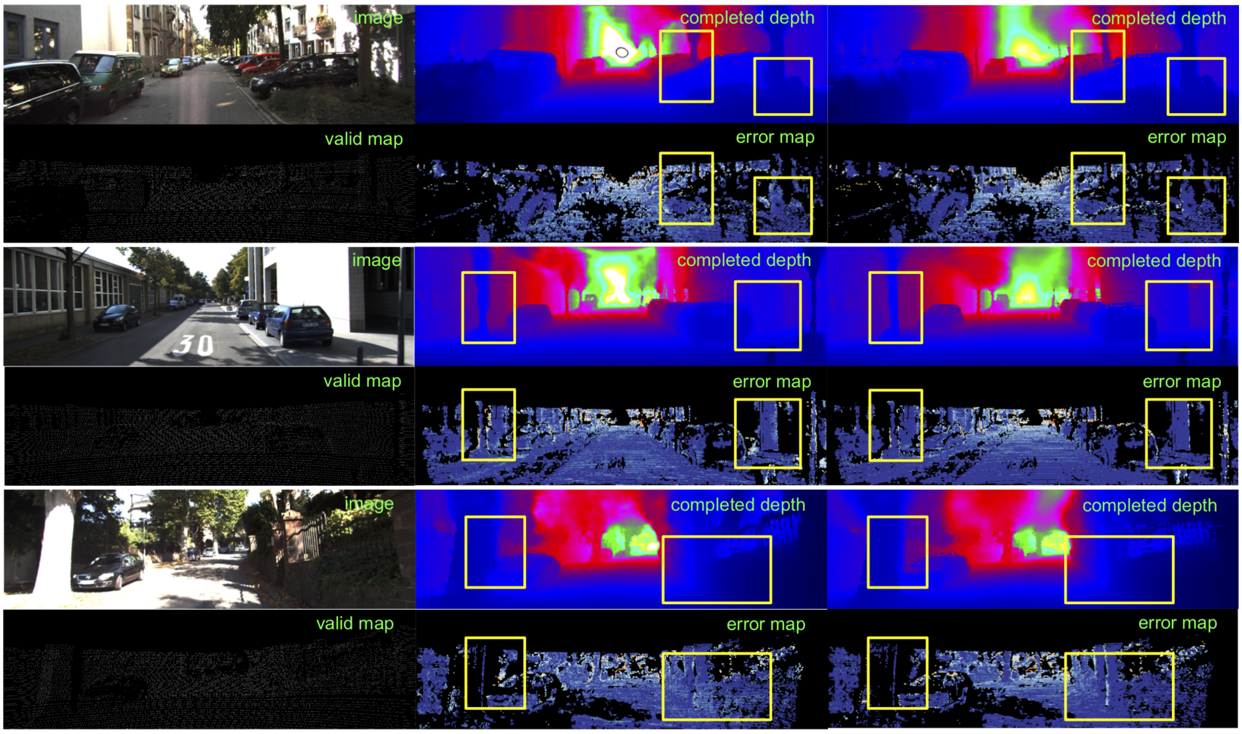

Figure 4.5: *Qualitative comparison to Ma et al. [MCK19] on the KITTI depth completion test set in the unsupervised setting.* Image and validity map of the sparse measurements (1st column), dense depth results and corresponding error map of [MCK19] (2nd column) and ours (3rd column). Warmer color in the error map denotes higher error. Yellow rectangles highlight the regions for detailed comparison. Note again that our network consistently performs better on fine and far structures and our completed dense depth maps have less visual artifacts (this includes the circle in the center of their prediction, row 1, column 2).

approach, however, does not suffer from these artifacts; instead, our predictions are globally correct and consistent with the scene geometry.

4.6 Discussion

In this work, we have described a system to infer a posterior probability over the depth of points in the scene corresponding to each pixel, given an image and a sparse aligned point cloud. Our method leverages a Conditional Prior Network, that allows the association of a probability to each depth value based on a single image, and combines it with a likelihood term for sparse depth measurements. Moreover, we exploit the availability of stereo imagery in constructing a photometric reconstruction term that further constrains the predicted depth to adhere to the scene geometry.

We have tested the approach both in a supervised and unsupervised setting. It should be noted that the difference between "supervised" and "unsupervised" in the KITTI benchmark is more quantitative than qualitative: the former has about 30% coverage in depth measurements, the latter about 5%. We show in Tab. 4.1 and 4.2 that our method achieves state-of-the-art performance in both supervised and unsupervised depth completion on the KITTI benchmark. Although we outperform other methods on score metrics that measures the deviation from the ground truth, we want to emphasize that our method does not simply produce a point estimate of depth, but provides a confidence measure, that can be used for more downstream processing, for instance for planning, control and decision making.

We have explored the effect of various hyperparameters, and are in the process of expanding the testing to real-world environments, where there could be additional errors and uncertainty due to possible time-varying misalignment between the range sensor and the camera, or between the two cameras when stereo is available, faulty intrinsic camera calibration, and other nuisance variability inevitably present on the field that is carefully weeded out in evaluation benchmarks such as KITTI. This experimentation is a matter of years, and well beyond the scope of this paper. Here we have shown that a suitably modified Conditional Prior Network can successfully transfer knowledge from prior data, including synthetic ones, to provide context to input range values for inferring missing data. This is important for downstream processing as the context can, for instance, help differentiate whether gaps in the point cloud are free space or photometrically homogeneous obstacles, as discussed in our motivating example in Fig. 4.1.

CHAPTER 5

Depth Completion from Inertial Odometry and Vision

5.1 Introduction

A sequence of images is a rich source of information about both the three-dimensional (3D) shape of the environment and the motion of the sensor within. Motion can be inferred at most up to a scale and a global Euclidean reference frame, provided sufficient parallax and a number of visually discriminative Lambertian regions that are stationary in the environment, and are visible from the camera. The position of such regions in the scene defines the Euclidean reference frame, with respect to which motion is estimated. Scale as well as two directions of orientation can be further identified by fusion with inertial measurements (accelerometers and gyroscopes) and, if available, a magnetometer can fix the last (Gauge) degree of freedom.

Because the regions defining the reference frame have to be visually distinctive ("features"), they are typically *sparse*. In theory, three points are sufficient to define a Euclidean Gauge if visible at all times. In practice, because of occlusions, any Structure From Motion (SFM) or simultaneous localization and mapping (SLAM) system maintains an estimate of the location of a sparse set of features, or "sparse point cloud," typically in the hundreds to thousands. These are sufficient to support a point-estimate of motion, but a rather poor representation of shape as they do not reveal the topology of the scene: The empty space between points could be empty, or occupied by a solid with a smooth surface radiance (appearance). Attempts to *densify* the sparse point cloud, by interpolation or regularization with generic priors such as smoothness, piecewise planarity and the like, typically fail since SFM yields far too sparse a reconstruction to inform topology. This is where the image

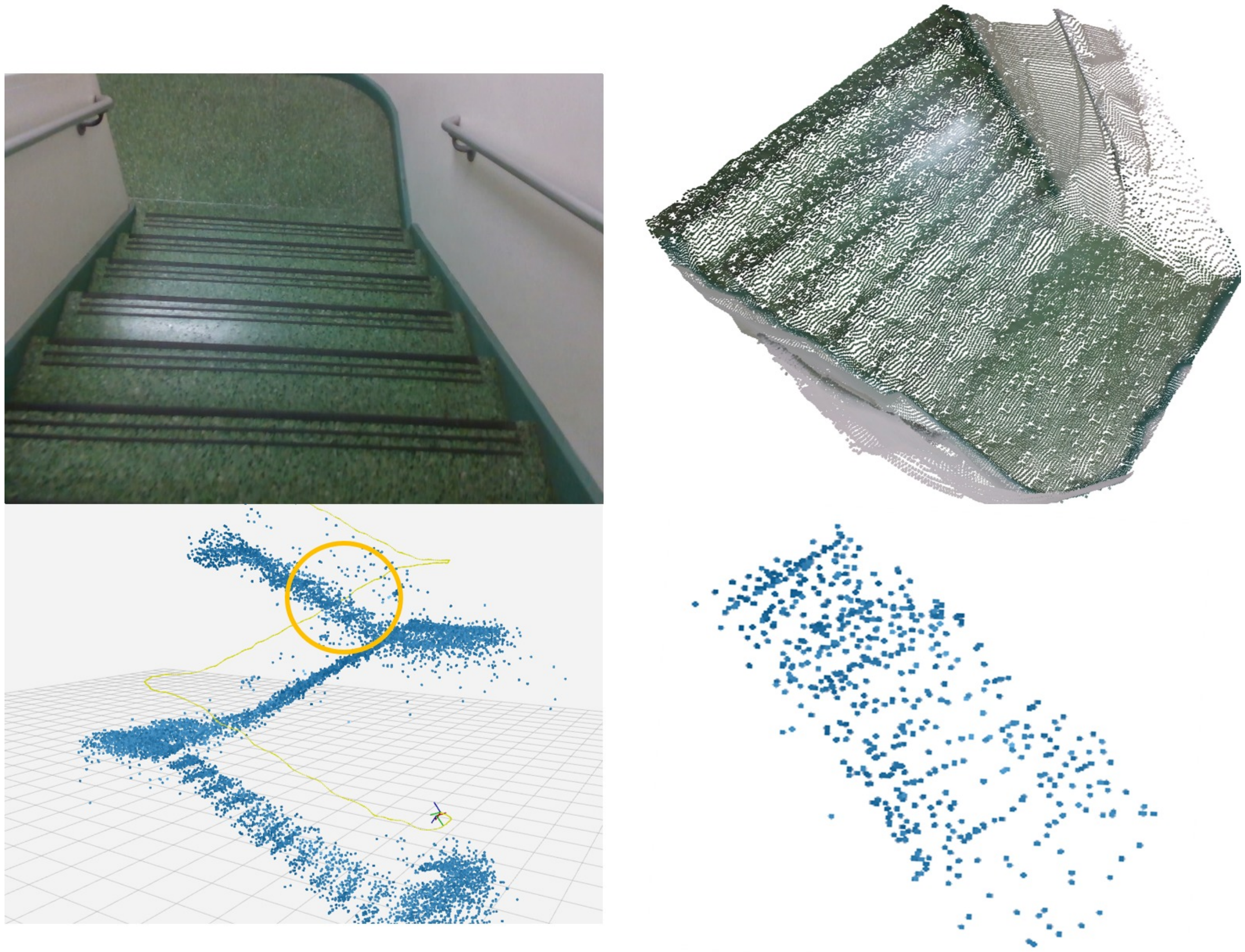

Figure 5.1: *Depth completion with Visual-Inertial Odometry (VIO) on the proposed VOID dataset* (best viewed in color at 5×). Bottom left: sparse reconstruction (blue) and camera trajectory (yellow) from VIO. The highlighted region is densified and zoomed in on the top right. Top left shows an image of the same region which is taken as input, and fused with the sparse depth image by our method. On the bottom right is the same view showing only the sparse points, insufficient to determine scene geometry and topology.

comes back in.

Inferring shape is ill-posed, even if the point cloud was generated with a lidar or structured light sensor. Filling the gaps relies on assumptions about the environment. Rather than designing ad-hoc priors, we wish to use the image to inform and restrict the set of possible scenes that are compatible with the given sparse points.

Summary of contributions

We use a predictive cross-modal criterion to score dense depth from images and sparse depth. This kind of approach is sometimes referred to as "self-supervised." Specifically, our method (i) exploits a set of constraints from temporal consistency (a.k.a. photometric consistency across temporally adjacent frames) to pose (forward-backward) consistency in a combination that has not been previously explored. To enable our pose consistency term, we introduce (ii) a set of logarithmic and exponential mapping layers for our network to represent motion using exponential coordinates, which we found empirically superior to other parameterizations.

The challenge in using sparse depth as a supervisory (feedback) signal is precisely that it is sparse. Information at the points does not propagate to fill the domain where depth is defined. Some computational mechanism to "diffuse the information" from the sparse points to their neighbors is needed. Our approach proposes (iii) a simple method akin to using a piecewise planar *"scaffolding"* of the scene, sufficient to transfer the supervisory signal from sparse points to their neighbors. This yields a two-stage approach, where the sparse points are first processed to design the scaffolding ("meshing and interpolation") and then "refined" using the images as well as priors from the constraints just described.

One additional contribution of our approach is (iv) to launch the first visual-inertial + depth dataset. The role of inertials is to enable reconstruction in *metric* scale, which is critical for robotic applications. Although scale can be obtained via other sensors, e.g., stereo, lidar, and RGB-D, we note they are not as widely available as monocular camera + inertial (almost every modern phone has it) and consume more power. Since inertial sensors are now ubiquitous and typically co-located with cameras in many mobile devices from phones to cars, we hope this dataset will foster additional exploration into combining the complementary strengths of visual and inertial sensors.

To evaluate our method, since no other visual-inertial + depth benchmark is available, and to facilitate comparison with similar methods, we adopt the KITTI benchmark, where a Velodyne (lidar) sensor provides sparse points with scale, unlike monocular SFM, but like visual-inertial odometry (VIO). Although the biases in lidar are different from VIO, this can be considered a baseline. Note that we only use the monocular stream of KITTI (not stereo) for fair comparison.

The result is a (v) two-stage approach of scaffolding and refining with a network that contains much fewer parameters than competing methods, yet achieves state-of-the-art performance in the "unsupervised" KITTI benchmark (a misnomer). The supervision in the KITTI benchmark is really fusion from separate sensory channels, combined with ad-hoc interpolation and extrapolation. It is unclear whether the benefit from having such data is outweighed by the biases it induces on the estimate, and in any case such supervision does not scale; hence, we forgo (pseudo) ground truth annotations altogether.

5.2 Related Work

Supervised Depth Completion minimizes the discrepancy between ground truth depth and depth predicted from an RGB image and sparse depth measurements. Methods focus on network topology [MCK19, USS17, YWS19], optimization [CWL18, DVP18, ZF18], and modeling [EFK18, HFY18]. To handle sparse depth, [MCK19] employed early fusion, where the image and sparse depth are convolved separately and the results concatenated as the input to a ResNet encoder. [JCW18] proposed late fusion via a U-net containing two NASNet encoders for image and sparse depth and jointly learned depth and semantic segmentation, whereas [YWS19] used ResNet encoders for late fusion. [EFK18] proposed a normalized convolutional layer to propagate sparse depth and used a binary validity map as a confidence measure. [HFY18] proposed an upsampling layer and joint concatenation and convolution to deal with sparse inputs. All these methods require per-pixel ground-truth annotation. What is called "ground truth" in the benchmarks is actually the result of data processing and aggregation of many consecutive frames. We skip such supervision and just infer dense depth by learning the cross-modal fusion from the virtually infinite volume of un-annotated data.

Unsupervised Depth Completion include [MCK19, SNC19, YWS19] who predict depth by minimizing the discrepancy between prediction and sparse depth input as well as the photometric error between the input image and its reconstruction from other viewpoints available only during training. [MCK19] used Perspective-n-Point (PnP) [LMF09] and Random Sample Consensus (RANSAC) [FB81] to align monocular image sequences for their photometric term with a second-order smoothness prior. Yet, [MCK19] does not generalize well to indoor scenes that contains many textureless regions (e.g. walls), where PnP with RANSAC may fail. [SNC19] used a local smoothness term, but instead minimized the photometric error between rectified stereo-pairs where pose is known. [YWS19] also leveraged stereo pairs and a more sophisticated photometric loss [WBS04]. [YWS19] replaced the generic smoothness term with a conditional prior to measure compatibility between the prediction and a learned depth model obtained by training a separate network on ground-truth depth. This method can be considered semi-unsupervised, and requires ground truth for training the prior. Using a network trained on a specific domain (e.g. outdoors) as a prior for an unsupervised method will not generalize when given extra data on a different domain (e.g. indoors). In contrast, our method is *fully unsupervised* and do not use any auxiliary ground-truth supervision. Moreover, our method outperforms [MCK19, YWS19] on the KITTI depth completion benchmark [USS17] while using fewer parameters.

Rotation Parameterization. To construct the photometric consistency loss during training, an auxiliary pose network is needed if no camera poses are available. While the translational part of the relative pose can be modeled as $T \in \mathbb{R}^3$, the rotational part belongs to the special orthogonal group $R \in SO(3) \doteq \{R \in \mathbb{R}^{3\times3} | R^\top R = I, \det(R) = +1\}$ [MSK12], which is represented by a 3×3 matrix. [KGC15] uses quaternions, which require an *additional* norm constraint; this is a soft constraint imposed in the loss function, and thus is not guaranteed. [FWS19, YS18b, ZBS17] use Euler angles which requires the composition of several matrices that may result in the rotation matrix to no longer be orthogonal. We use the exponential map on $SO(3)$ to map the output of the pose network to a rotation matrix. Though theoretically similar, we empirically found that the exponential map is more beneficial than the Euler angles in Sec. 5.7.

Our contributions are a simple, yet effective two-stage approach resulting in a large reduction in network parameters while achieving state-of-the-art performance on the unsupervised

KITTI depth completion benchmark; using exponential parameterization of rotation for our pose network; a pose consistency term that enforces forward and backward motion to be the inverse of each other, and finally a new depth completion benchmark for visual-inertial odometry systems with indoor and outdoor scenes and challenging motion.

5.3 Method Formulation

We reconstruct a 3D scene given an RGB image $I_t : \mathbb{R}^2 \supset \Omega \mapsto \mathbb{R}^3_+$ and the associated set of sparse depth measurements $z_s : \Omega \supset \Omega_s \mapsto \mathbb{R}_+$.

We begin by assuming that world surfaces are graphs of smooth functions (charts) locally supported on a piecewise planar domain (scaffolding). We construct the scaffolding from the sparse point cloud ("Scaffolding" in Fig. 5.2) to obtain z_i, then learn a completion model refining z_i by leveraging the monocular sequences (I_{t-1}, I_t, I_{t+1}), of frames before and after the given time t, and the sparse depth z_s. We compose a surrogate loss $\mathcal{L}$ (Eqn. 5.2) for driving the training process, using an encoder-decoder architecture $f_\theta(\cdot)$ parameterized by weights θ, where the input is an image with its scaffolding (I_t, z_i), and the output is the dense depth $\hat{z} = f_\theta(I_t, z_i)$.

5.3.1 A Two-Stage Approach

Depth completion is a challenging problem due to the sparsity level of the depth input, z_s. As the density of sparse depth measurements covers $\approx 5\%$ of the image plane for the outdoor self-driving scenario (Sec. 5.5.1) and less than $\approx 1\%$ for the indoor setting (Sec. 5.7.3), generally only a single measurement will be present within a local neighborhood and in most instances none. This renders *conventional convolutions ineffective* as each sparse depth measurement can be seen as a Dirac delta and convolving a kernel over the entire sparse depth input will give mostly zero activations. Hence, [EFK18], [HFY18], and [USS17] proposed specialized operations to propagate the information from the sparse depth input through the network. We, instead, propose a two-stage approach that circumvents this problem by first approximating a coarse scene geometry with scaffolding and training a network to refine the approximation.

5.3.2 Scaffolding

Given sparse depth measurements z_s, our goal is to create a coarse approximation of the scene; yet, the topology of the scene is not informed by z_s. Hence, we must rely on a prior or an assumption – that surfaces are locally smooth and piecewise planar. We begin by applying the lifting transform [Bro79] to z_s, mapping z_s from 2-d to 3-d space. We then compute its convex hull [BDD96], of which the lower envelope is taken as the Delaunay triangulation of the points in z_s – resulting in a triangular mesh in Barycentric coordinates.

To form the tessellation of the triangular mesh, we approximate each surface using linear interpolation within the Barycentric coordinates and the resulting scaffolding is projected back onto the image plane to produce z_i. For a given triangle, simple interpolation is sufficient for recovering the plane as a linear combination of the co-planar points. For sets of points not co-planar, interpolation will give an approximation, with which we refine using a network.

5.3.3 Refinement

Given an RGB image and its corresponding piece-wise planar scaffolding (I_t, z_i), we train a network to recover the 3-d scene by refining z_i based on information from I_t. Our network learns to refine *without* ground-truth supervision by minimizing Eqn. 5.2.

Network Architecture. We propose two encoder-decoder architectures with skip connections following the late fusion paradigm [JCW18, YWS19]. Each encoder has an image branch and a depth branch – the image branch contains 75% of the total features in the encoder and the depth branch 25%. The latent representation of the branches are concatenated and fed to the decoder. We propose a VGG11 encoder ($\approx$ 5.7M parameters) containing 8 convolution layers for each branch as our best performing model, and a VGG8 encoder ($\approx$ 2.4M parameters) containing only 5 convolution layers for each branch as our light-weight model. Both VGG11 and VGG8 encoders use a generic decoder with $\approx$ 4M parameters – giving us a total of $\approx$ 9.7M and $\approx$ 6.4M parameters, respectively. This is in contrast to other unsupervised methods [MCK19] (who follows early fusion and concatenates features from the two branches after the first convolution) and [YWS19] (late fusion) – both of whom use ResNet34 encoders with $\approx$ 23.8M and $\approx$ 14.8M parameters, respectively. Both [MCK19, YWS19] employ the same decoder with $\approx$ 4M parameters – totaling to $\approx$ 27.8M and $\approx$ 18.8M parameters, respectively. We show in Sec. 5.5.1 that despite having 76.1% and 61.5% fewer encoder parameters than [MCK19] and [YWS19], our VGG11 model outperforms both [MCK19] and [YWS19]. Moreover, we note that the performance of our VGG8 model is still comparable to that of VGG11 and still surpasses [MCK19] and [YWS19] while having a 89.9% and 83.9% reduction in the encoder parameters. More details on our network architectures can be found in Sec. V of Supp. Mat.

Logarithmic and Exponential Map Layers. To construct our objective (Eqn. 5.2), we leverage a pose network [KGC15] to regress the relative camera poses $g = (R, T) \in SE(3) \doteq \{(R, T) | R \in SO(3), T \in \mathbb{R}^3\}$. We present a novel logarithmic map layer: $\log : SO(3) \mapsto so(3)$, where $so(3)$ is the tangent space of $SO(3)$, and an exponential map layer: $\exp : \; so(3) \mapsto SO(3)$ – for mapping R between $SO(3)$ and $so(3)$. We use the logarithmic map to construct the pose consistency loss (Eqn. 5.6), and the exponential to map the output of the pose network $\omega \doteq [\omega_1, \omega_2, \omega_3]^\top \in \mathbb{R}^3$ as coordinates in $so(3)$ to a rotation matrix:

$$R(\omega) = \exp(\hat{\omega}) \doteq I + \hat{\omega} \sin \|\omega\|_2 + \hat{\omega}^2 (1 - \cos \|\omega\|_2) \tag{5.1}$$

where the hat operator $\hat{\cdot}$ maps $\omega \in \mathbb{R}^3$ to a skew-symmetric matrix [MSK12]. We train of our pose network using a surrogate loss (Eqn. 5.3) without explicit supervision. An ablation study on the use of exponential coordinates and pose consistency term on KITTI odometry is available in Supp. Mat. Sec. III.

Our approach contains two stages: (i) we generate a coarse piecewise planar approximation of the scene from the sparse depth inputs z_s via scaffolding and (ii) we feed the resulting depth map along with the associated RGB image to our network for refinement (Fig. 5.2). This approach *alleviates* the network from the need of learning from sparse inputs, for which [MCK19] and [YWS19] compensated with parameters. We show the effectiveness of this approach by achieving the state-of-the-art on the unsupervised KITTI depth completion

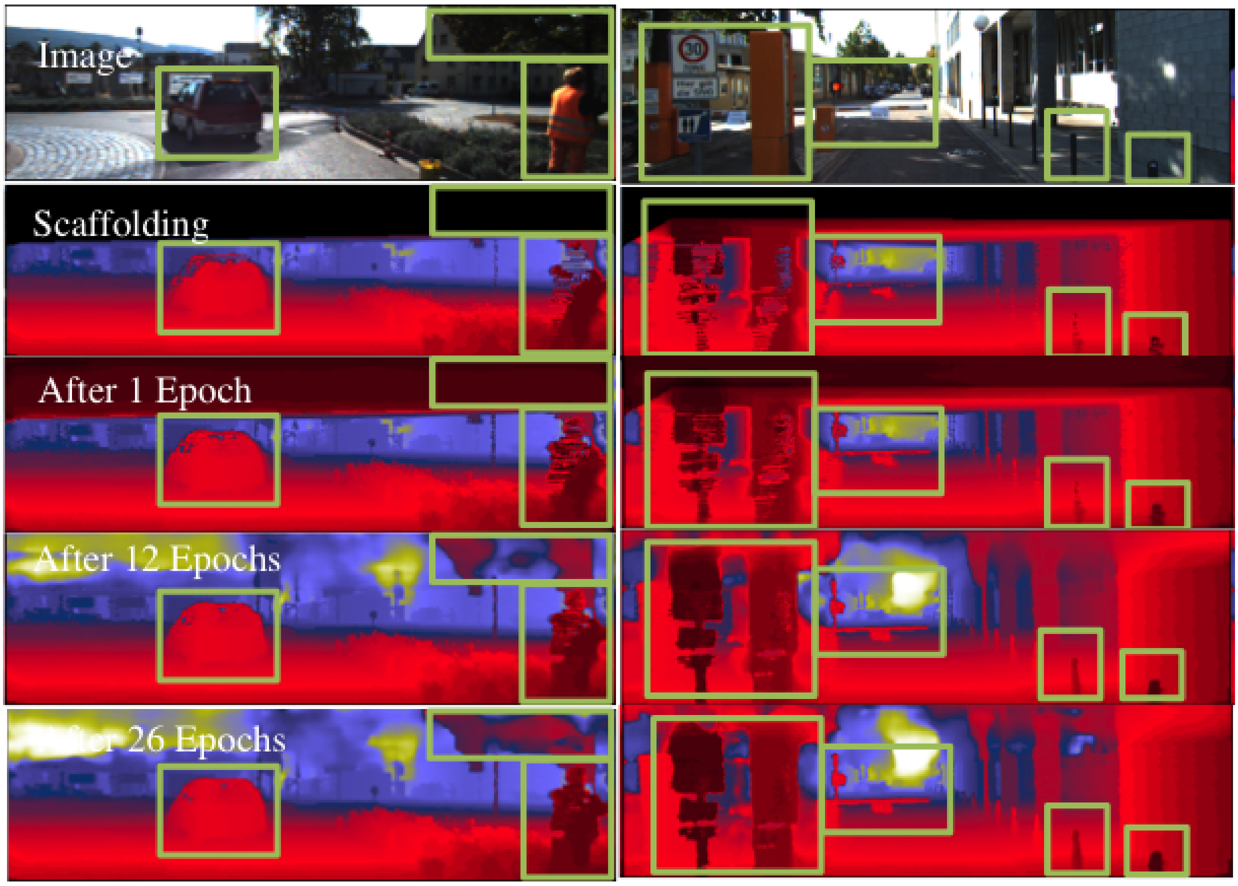

Figure 5.2: *Learning to refine* (best viewed at 5× with color). Our network learns to refine the input scaffolding. Green rectangles highlight the regions for comparison throughout the course of training. The network first learns to copy the input and later learns to fuse information from RGB image to refine the approximated depth from scaffolding (see column 1 pedestrian and column 2 street signs).

benchmark with half as many parameters as the prior-art.

5.4 Loss Function

Our loss function is a linear combination of four terms that constrain (i) the photometric consistency between the observed image and its reconstructions from the monocular sequence, (ii) the predicted depth to be similar to that of the associated available sparse depth, (iii) the composition of the predicted forward and backward relative poses to be the identity, and (iv) the prediction to adhere to local smoothness.

$$\mathcal{L} = w_{ph}L_{ph} + w_{sz}L_{sz} + w_{pc}L_{pc} + w_{sm}L_{sm} \tag{5.2}$$

where L_{ph} denotes photometric consistency, L_{sz} sparse depth consistency, L_{pc} pose consistency, and L_{sm} local smoothness. Each loss term L is described in the next subsections and the associated weight w in Sec. 5.6.

5.4.1 Photometric Consistency

We enforce temporal consistency by minimizing the discrepancy between each observed image I_t and its reconstruction $\hat{I}_\tau$ from temporally adjacent images I_τ, where $\tau \in T \doteq \{t-1, t+1\}$:

$$\hat{I}_\tau(x) = I_\tau\big(\pi g_{\tau t}\mathbf{K}^{-1}\bar{x}z(x)\big) \tag{5.3}$$

where $\bar{x} = [x^T \ 1]^T$ are the homogeneous coordinates of $x \in \Omega$, $g_{\tau t} \in SE(3)$ is the relative pose of the camera from time t to τ, $\mathbf{K}$ denotes the camera intrinsics, and π refers to the perspective projection.

Our photometric consistency term is a combination of the average per pixel reprojection residual with an $L1$ penalty and `SSIM` [WBS04], a perceptual metric that is invariant to local illumination changes:

$$L_{ph} = \frac{1}{|\Omega|}\sum_{\tau\in T}\sum_{x\in\Omega} w_{co}|I_t(x) - \hat{I}_\tau(x)| + w_{st}\big(1 - \mathtt{SSIM}(I_t(x), \hat{I}_\tau(x))\big) \tag{5.4}$$

We use 3 × 3 image patches centered at location x for `SSIM`. w_{co} and w_{st} can be found in Sec. 5.6.

5.4.2 Sparse Depth Consistency

Our sparse depth consistency term provides our predictions with *metric* scale by encouraging the predictions $\hat{z}$ to be similar to that of the *metric* sparse depth z_s available from lidar in KITTI dataset (Sec. 5.5.1) and sparse reconstruction in our visual-inertial dataset (Sec. 5.5.2). Our sparse depth consistency loss is the $L1$-norm of the difference between the predicted depth $\hat{z}$ and the sparse depth z_s averaged over Ω_s (the support of the sparse depth):

$$L_{sz} = \frac{1}{|\Omega_s|} \sum_{x \in \Omega_s} |\hat{z}(x) - z_s(x)| \tag{5.5}$$

5.4.3 Pose Consistency

A pose network takes an ordered pair of images (I_t, I_τ) and outputs the relative pose $g_{\tau t} \in SE(3)$ (forward pose). When a temporally swapped pair (I_τ, I_t) is fed to the network, the network is expected to output $g_{t\tau}$ (backward pose) – the inverse of $g_{\tau t}$, *i.e.*, $g_{\tau t} \cdot g_{t\tau} = e \in SE(3)$. The forward-backward pose consistency thus penalizes the deviation of the composed pose from the identity:

$$L_{pc} = \| \log(g_{\tau t} \cdot g_{t\tau}) \|_2^2 \tag{5.6}$$

where $\log : SE(3) \mapsto se(3)$ is the logarithmic map.

5.4.4 Local Smoothness

We impose a smoothness loss on the predicted depth $\hat{z}$ by applying an $L1$ penalty to the gradients in both the x and y directions of the predicted depth $\hat{z}$:

$$L_{sm} = \frac{1}{|\Omega|} \sum_{x \in \Omega} \lambda_X(x) |\partial_X \hat{z}(x)| + \lambda_Y(x) |\partial_Y \hat{z}(x)| \tag{5.7}$$

where $\lambda_X = e^{-|\partial_X I_t(x)|}$ and $\lambda_Y = e^{-|\partial_Y I_t(x)|}$ are the edge-awareness weights to allow for discontinuities in regions corresponding to object boundaries.

5.4.5 The Role of Inertials

Although inertials are not directly present in the loss, their role in *metric* depth completion is crucial. Without inertials, a SLAM system cannot produce sparse point clouds in metric scale, which are then used as both the input to the scaffolding stage (Sec. 5.3.2) and a supervisory signal (Eqn. 5.5).

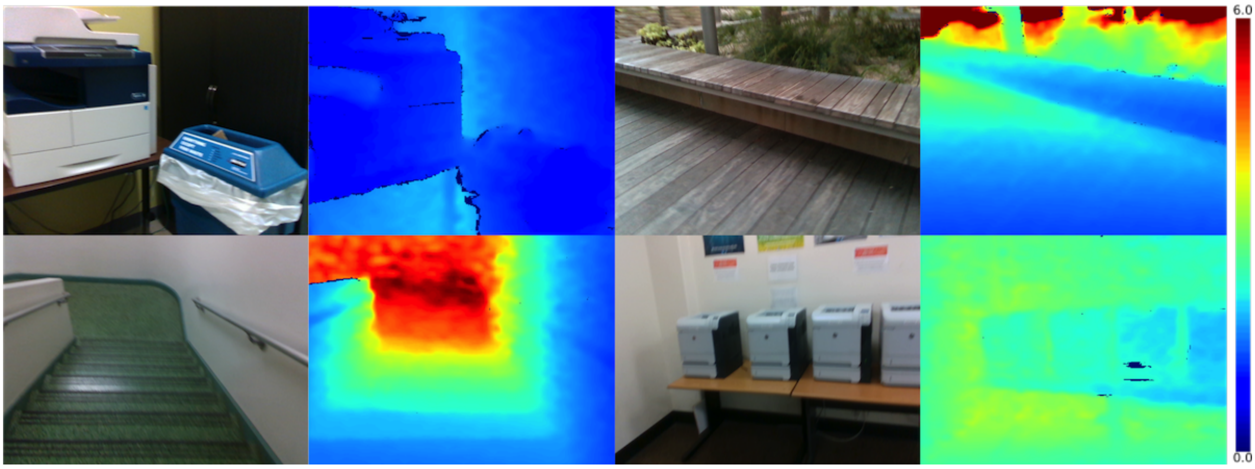

Figure 5.3: *Sample RGB + D images* in the VOID dataset (best viewed in color at 5×). Color bar shows the depth range.

5.5 Datasets

5.5.1 KITTI Benchmark

We evaluate our approach on the KITTI depth completion benchmark [USS17]. The dataset provides $\approx 80,000$ raw image frames and associated sparse depth maps. The sparse depth maps are the raw output from the Velodyne lidar sensor, each with a density of $\approx 5\%$. The ground-truth depth map is created by accumulating the neighbouring 11 raw lidar scans, with dense depth corresponding to the bottom 30% of the images. We use the officially selected 1,000 samples for validation and we apply our method to 1,000 testing samples, with which we submit to the official KITTI website for evaluation. The results are reported in Table 5.2.

5.5.2 VOID Benchmark

While KITTI provides a standard benchmark for evaluating depth completion in the driving scenario, there exists no standard depth completion benchmark for the indoor scenario. [MCK19, YWS19] used NYUv2 [SHK12] – an RGB-D dataset – to develop and evaluate their models on indoor scenes. Yet, each perform a different evaluation protocol with different sparse depth samples – varying densities of depth values were randomly sampled from the depth frame, preventing direct comparisons between methods. Though this is reasonable as a proof of concept, it is not realistic in the sense that no sensor measures depth at random locations.

The VOID dataset. We propose a new publicly available dataset for a real world use case of depth completion by bootstrapping sparse reconstruction in *metric* space from a SLAM system. While it is well known that metric scale is not observable in the purely image-based SLAM and SFM setting, it has been resolved by the recent advances in VIO [JS11, MR07], where metric pose and structure estimation can be realized in a gravity-aligned and scaled reference frame using a inertial measurement unit (IMU). To this end, we leverage an off-the-shelf VIO system [1], atop which we construct our dataset and develop our depth completion model. While there are some visual-inertial datasets (e.g. TUM-VI [SGD18] and PennCOSYVIO [PSD17]), they lack per-frame dense depth measurements for cross-modal validation, and are also relatively small – rendering them unsuitable for training deep learning models.

Our dataset is dubbed "Visual Odometry with Inertial and Depth" or "VOID" for short and is comprised of RGB video streams and inertial measurements for *metric* reconstruction along with per-frame dense depth for cross-modal validation.

Data acquisition. Our data was collected using the latest Intel RealSense D435i camera [2], which was configured to produce synchronized accelerometer and gyroscope measurements at 400 Hz, along with synchronized VGA-size (640×480) RGB and depth streams at 30 Hz. The depth frames are acquired using active stereo and is aligned to the RGB frame using the sensor factory calibration (see Fig. 5.3). All the measurements are time-stamped.

The SLAM system we use is based on [JS11] – an EKF-based VIO model. While the VIO recursively estimates a joint posterior of the state of the sensor platform (e.g. pose, velocity, sensor biases, and camera-to-IMU alignment) and a small set of reliable feature points, the 3D structure it estimates is extremely sparse – typically $20 \sim 30$ feature points

[1] `https://github.com/ucla-vision/xivo`

[2] `https://realsense.intel.com/depth-camera/`

Metric	units	Definition
MAE	*mm*	$\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{z}(x)-z_{gt}(x)\rvert$
RMSE	*mm*	$\big(\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert\hat{z}(x)-z_{gt}(x)\rvert^2\big)^{1/2}$
iMAE	*1/km*	$\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert 1/\hat{z}(x)-1/z_{gt}(x)\rvert$
iRMSE	*1/km*	$\big(\frac{1}{\lvert\Omega\rvert}\sum_{x\in\Omega}\lvert 1/\hat{z}(x)-1/z_{gt}(x)\rvert^2\big)^{1/2}$

Table 5.1: *Error metrics.* Error metrics for evaluating KITTI and VOID depth completion benchmarks, where z_{gt} is the ground truth.

(in-state features). To facilitate 3D reconstruction, we track a moderate amount of out-of-state features in addition to the in-state ones, and estimate the depth of the feature points using auxiliary depth sub-filters [MSK12].

The benchmark. We evaluate our method on the VOID depth completion benchmark, which contains 56 sequences in total, both indoor and outdoor with challenging motion. Typical scenes include classrooms, offices, stairwells, laboratories, and gardens. Of the 56 sequences, 48 sequences ($\sim 40K$ frames) are designated for training and 8 sequences for testing, from which we sampled 800 frames to construct the testing set. Our benchmark provides sparse depth maps at three density levels. We configured our SLAM system to track and estimate depth of 1500, 500 and 150 feature points, corresponding to $0.5\%, 0.15\%$ and 0.05% density of VGA size, which are then used in the depth completion task.

5.6 Implementation Details

Our approach was implemented using TensorFlow [ABC16]. With a Nvidia GTX 1080Ti, training takes $\approx$ 90 hours for our VGG11 model and $\approx$ 70 hours for our VGG8 model on KITTI depth completion benchmark (Sec. 5.5.1) for 30 epochs; whereas training takes $\approx$ 10 hours and $\approx$ 7 hours on the VOID benchmark (Sec. 5.5.2) for 10 epochs. Inference takes $\approx$ 22 ms per image. We used Adam [KB14] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize our network end-to-end with a base learning rates of 1.2×10^{-4} for KITTI and 1×10^{-4} for

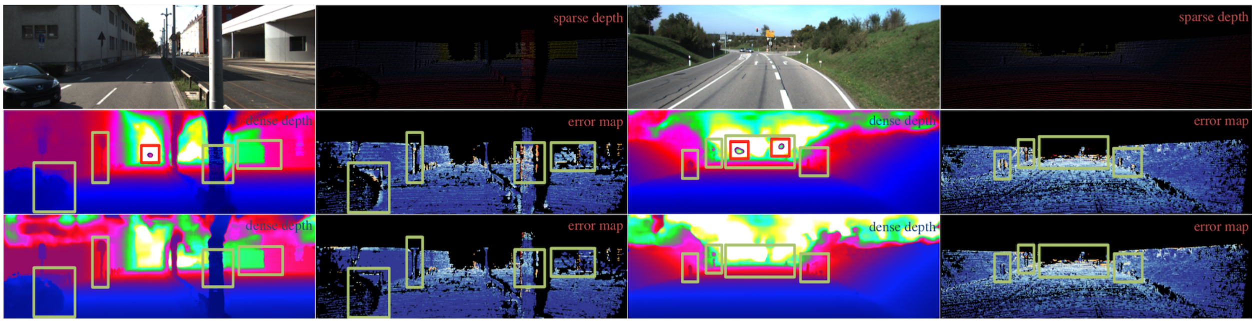

Figure 5.4: *Qualitative evaluation on KITTI benchmark.* Top to bottom: input image and sparse depth, results of [MCK19], our results. Results are taken from KITTI online test server. Warmer colors in the error map denote higher error. Green rectangles highlight regions for detail comparison. We perform better in general, particularly on thin structures and far regions. [MCK19] exhibit artifacts resembling scanlines and "circles" for far away regions (highlighted in red).

VOID . We decrease the learning rate by half after 18 epochs for KITTI and 6 epochs for VOID , and again after 24 epochs and 8 epochs, respectively. We train our network with a batch size of 8 using a 768 × 320 resolution for KITTI and 640 × 480 for VOID . We are able to achieve our results on the KITTI benchmark using the following set of weights for each term in our loss function: $w_{ph} = 1.00$, $w_{co} = 0.20$, $w_{st} = 0.40$, $w_{sz} = 0.20$, $w_{pc} = 0.10$ and $w_{sm} = 0.01$. For the VOID benchmark, we increased w_{sz} to 1.00 and w_{sm} to 0.10. We do not use any data augmentation.

5.7 Experiments and Results

5.7.1 KITTI Depth Completion Benchmark

We compare our approach with recent unsupervised depth completion methods on the official KITTI depth completion benchmark in Table 5.2 using error metrics in Table 5.1 and show quantitative results in Fig. 5.4. The results of the methods listed are taken directly from their papers. We note that [YWS19] only reported their result in their paper and do have have an entry in KITTI depth completion benchmark for their unsupervised model. Hence, we compare qualitatively with the prior-art [MCK19]. Our VGG11 model outperforms the state-of-the-art [YWS19] on every metric by as much as 12.8% on MAE, 7.4% on RMSE, 9.1% on iMAE while using 48.4% fewer parameters. Our light-weight VGG8 model also outperforms [YWS19] on MAE by 11.3%, RMSE by 7.8% and iMAE by 3% while having 66% fewer parameters; [YWS19] beat our light-weight model by 2.2% on iRMSE. We note that [YWS19] trains a separate network using ground-truth depth and uses it as supervision to train their model for depth completion. Moreover, [YWS19] exploits rectified stereo-imagery where the pose of the cameras is known; whereas, we learn our pose by jointly training the pose network with our depth predictor. In comparison to [MCK19] (who also uses monocular videos), our VGG11 model outperforms them by 14.5% on MAE, 10% on RMSE, 23.6% on iMAE, and 12.5% on iRMSE while using 65.1% fewer parameters. Our VGG8 model outperforms [MCK19] by 13.1% on MAE, 10.4% on RMSE, 18.5% on iMAE, and 10.1% on iRMSE while using 80% fewer parameters. We also note that the qualitative results of [MCK19] contains artifacts such as apparent scanlines of the Velodyne and "circles" in far regions. As an introspective exercise, we plot the mean error of our model at varying distances on the KITTI validation set (Fig. 5.5) and overlay it with the ground truth depth distribution to show that our model performs very well in distances that matter in real-life scenarios. Our performance begins to degrade at distances larger than 80 meters; this is due to the lack of sparse measurements and insufficient parallax – problems that plague methods relying on multi-view supervision.

5.7.2 KITTI Depth Completion Ablation Study

We analyze the effect brought by each of our contributions through a quantitative evaluation on the KITTI depth completion validation set (Table 5.3). Our two baseline models, scaffolding and vanilla model trained without scaffolding, perform poorly in comparison to the models that are trained with scaffolding – showcasing the effectiveness of our refinement approach. Although the loss functions are identical, exponential parameterization consistently improves over Euler angles across all metrics. While [FWS19, WBZ18a, YS18b] train their pose network using the photometric error as a surrogate loss with no additional constraint,

Method	MAE	RMSE	iMAE	iRMSE
Schneider et al. [SSP16]	605.47	2312.57	2.05	7.38
Ma et al. [MCK19]	350.32	1299.85	1.57	4.07
Yang et al. [YWS19]	343.46	1263.19	1.32	3.58
Ours VGG8	304.57	**1164.58**	1.28	3.66
Ours VGG11	**299.41**	1169.97	**1.20**	**3.56**

Table 5.2: *KITTI depth completion benchmark.* We compare our model to unsupervised methods on the KITTI depth completion benchmark [USS17]. Our VGG11 model outperforms state-of-the-art [YWS19] across all metrics while using 48.4% less parameters. Our light-weight (VGG8) model achieves similar performance and in fact marginally outperforms our VGG11 model despite having 34% fewer parameters than our VGG11 model. Moreover, our VGG8 model outperforms [MCK19] and across all metrics and [YWS19] on MAE, RMSE, and iMAE despite having 80% and 66% fewer parameters, respectively.

we show that it is beneficial to impose our pose consistency term (Sec. 5.6). By constraining the forward and backward poses to be inverse of each other, we obtain a more accurate pose resulting in better depth prediction. Our experiments verify this claim as we see an improvement in MAE by 2.3%, RMSE by 1.3%, iMAE by 5.5%, and iRMSE by 3.9% in Table 5.3. We note that the improvement does not seem significant on KITTI as the motion is mostly planar; however, when predicting non-trivial 6 DoF motion (Sec. 5.7.4), we see a significant boost when employing this term. Our model trained with the full loss function produces the best results (bolded in Table 5.2) and is the state-of-the-art for unsupervised KITTI depth completion benchmark. We further propose a light-weight (VGG8) model that only contains ≈ 6.4M parameters. Although our VGG8 model has 3.3M fewer (34% reduction) parameters than our VGG11 model, we note that the performance does not degrade by much – our VGG8 model only trails the VGG11 model by 1.2% in MAE, 6.6% in iMAE, 3.5% in iRMSE, and marginally beats our VGG11 model on RMSE by 0.7% on the KITTI validation set.

Model	Encoder	Rot.	MAE	RMSE	iMAE	iRMSE
Scaffolding	-	-	443.57	1990.68	1.72	6.43
$L_{ph} + L_{sz} + L_{sm}$ (vanilla)	VGG11	Euler	347.14	1330.88	1.46	4.22
$L_{ph} + L_{sz} + L_{sm}$	VGG11	Euler	327.84	1262.46	1.31	3.87
$L_{ph} + L_{sz} + L_{sm}$	VGG11	Exp.	312.10	1255.21	1.28	3.86
$L_{ph} + L_{sz} + L_{pc} + L_{sm}$	VGG11	Exp.	**305.06**	1239.06	**1.21**	**3.71**
$L_{ph} + L_{sz} + L_{pc} + L_{sm}$	VGG8	Exp.	308.81	**1230.85**	1.29	3.84

Table 5.3: *KITTI depth completion ablation study.* We compare variants of our model on the KITTI depth completion validation set. Each model is denoted by its loss function. The results of Scaffolding Only is produced using linear interpolation over a triangular mesh; we assign average depth to regions with missing interpolated depth. It is clear that scaffolding alone (row 1) and our baseline model trained *without* scaffolding (row 2) do poorly compared to our models that combine both (rows 3-6). Our full model using VGG11 produces the best overall results and achieves state-of-the-art on the test set Table 5.2. We note that our light-weight VGG8 model achieve similar performance and even marginally beating our VGG11 model on the RMSE metric despite having 34% fewer parameters.

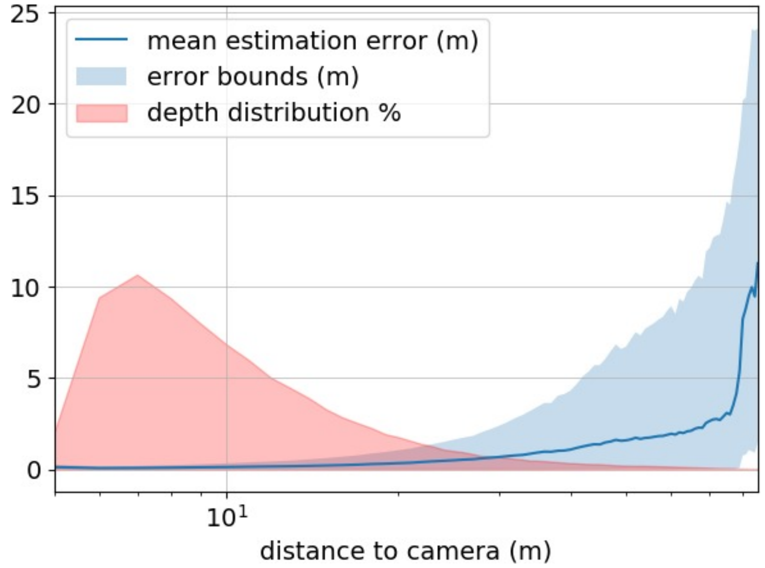

Figure 5.5: *Error characteristics of our model on KITTI.* The abscissa shows the distance of sparse data points measured by Velodyne, of which the percentage of all the data points is shown in red; the blue curve shows the mean absolute error of the estimated depth at the given distance, of which the 5-*th* and 95-*th* percentile enclose the light blue region.

5.7.3 VOID Depth Completion Benchmark

We evaluate our method on the VOID depth completion benchmark for all three density levels (Table 5.5) using error metrics in Table 5.1. As the photometric loss (Sec. 5.4) is largely dependent on obtaining the correct pose, we additionally propose a hybrid model, where the relative camera poses from our visual-inertial SLAM system are used to construct the photometric loss to show a upper bound on performance. In contrast to the KITTI depth completion benchmark, which provides ≈ 5% sparse depth over the image domain concentrated on the bottom third of the image, the VOID benchmark only provides ≈ 0.5%, ≈ 0.15% and ≈ 0.05% densities in sparse depth (10, 33, and 100 times less than KITTI). Yet, our method is still able to produce reasonable results for indoor scenes with a MAE of ≈ 8.5 centimeters on 0.5% density and ≈ 17.9 centimeters when given only 0.05%. As most scenes contain textureless regions, sparse depth supervision becomes important as photometric reconstruction is unreliable. Hence, we see a degrade in performance as the density decreases. Yet, we degrade gracefully: as the density decreases by 100X, our error only doubles. Also,

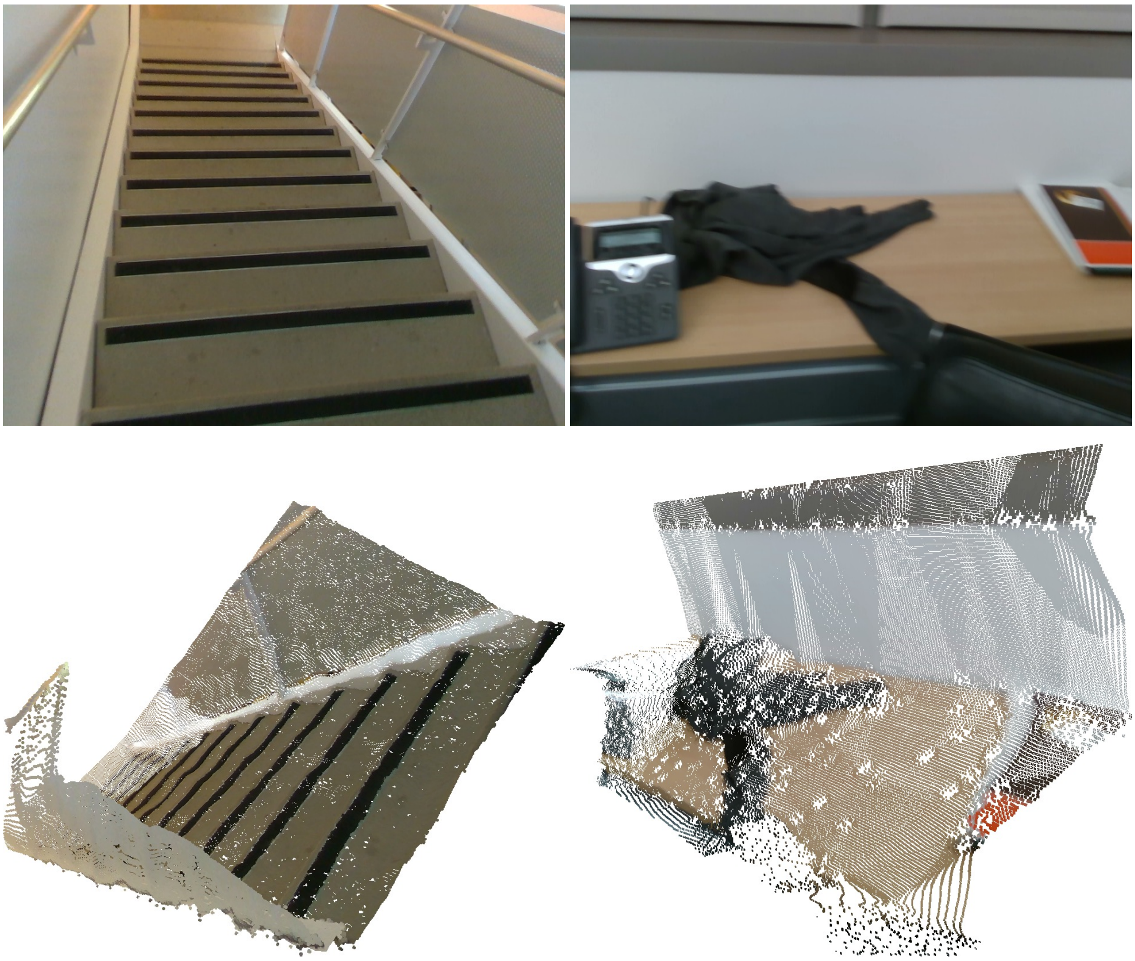

Figure 5.6: *Qualitative evaluation on VOID benchmark.* Top: Input RGB images. Bottom: Densified depth images back-projected to 3D, colored, and viewed from a different vantage point.

we observe systematic performance improvement in all the evaluation metrics (Table 5.5) when replacing the pose network with SLAM pose. This can be largely attributed to the necessity for the correct pose to minimize photometric error during training. Our pose network may not be able to consistently predict the correct pose due to the challenging motion of the dataset. Fig. 5.6 shows two sample RGB images with the densified depth images back-projected to 3D, colored, and viewed from a different vantage point.

Method	MAE	RMSE	iMAE	iRMSE
PoseNet + Eul.	108.97	212.16	64.54	142.64
PoseNet + Exp.	103.31	179.05	63.88	131.06
PoseNet + Exp. + L_{pc}	85.05	169.79	48.92	104.02
SLAM Pose	**73.14**	**146.40**	**42.55**	**93.16**

Table 5.4: *VOID depth completion benchmark and ablation study.* We compare the variants of our pose network. SLAM Pose replaces the output of pose network with SLAM estimated pose to gauge an upper bound in performance. When using our pose consistency term with exponential parameterization, our method approaches the performance of our method when using SLAM pose.

5.7.4 VOID Depth Completion Ablation Study

To better understand the effect of rotation parameterization and our pose consistency loss (Eqn. 5.6) on the depth completion task, we compare variants of our model and again replace the pose network with SLAM pose to show an upper-bound on performance. Although exponential outperforms Euler parameterization, we note that their results are in fact 29.2 and 32.9% worse than using SLAM pose on MAE, 18.2 and 30.1% worse on RMSE, 33% and 34% worse on iMAE, and 29% and 34.7% worse on iRMSE, respectively. However, we observe a performance boost when applying our pose consistency term and our model improves over exponential without pose consistency by 17.7% on MAE, 5.2% on RMSE, 23.4% on iMAE, and 20.6% on iRMSE. Moreover, it only trails the one trained with SLAM pose by 14% on MAE, 13.8% on RMSE, 13% on iMAE, and 10.4% on iRMSE. This trend still holds when density decreases (Table 5.5). This suggests that despite the additional constraint, the pose network still have some difficulties predicting the pose due to the challenging motion. This finding, along with results from Table 5.5, sheds light to the usage of classic SLAM systems in the era of deep learning, which also urges us to develop and test pose networks on the VOID dataset which features non-trivial 6 DoF motion – much more challenging than the mostly-planar motion found in the KITTI dataset.

Density	Pose From	MAE	RMSE	iMAE	iRMSE
$\sim 0.5\%$	PoseNet	85.05	169.79	48.92	104.02
	SLAM	73.14	146.40	42.55	93.16
$\sim 0.15\%$	PoseNet	124.11	217.43	66.95	121.23
	SLAM	118.01	195.32	59.29	101.72
$\sim 0.05\%$	PoseNet	179.66	281.09	95.27	151.66
	SLAM	174.04	253.14	87.39	126.30

Table 5.5: *Depth completion on VOID with varying sparse depth density.* The VOID dataset contains VGA size images (480 × 640) of both indoor and outdoor scenes with challenging motion. For "Pose From", SLAM refers to relative poses estimated by a SLAM system, and PoseNet refers to relative poses predicted by a pose network.

5.8 Discussion

In this work, we introduced a two-stage approach that achieves state-of-the-art performance on the KITTI depth completion benchmark. By learning to refine the scaffolding built from sparse points, we can bypass the sparse input problem that previous works have tried to solve by using sparsity-invariant operations. We additionally explored rotation parameterization and proposed a pose consistency constraint that enforced forward-backward motion consistency, both improving our results on the depth completion task for both KITTI and our newly proposed VOID dataset benchmarks. We also show that they improve the predicted pose on KITTI odometry dataset in Supp. Mat. Sec. III. However, we note that the performance of our model using a pose network still trails the model trained with SLAM pose on the VOID dataset. This can be attributed to the challenging motion on VOID as opposed to the planar motion on KITTI.

While deep networks have attracted a lot of attention as a general framework to solve an array of problems, we must note that pose may be difficult to learn on datasets with non-trivial 6 DoF motion – which the SLAM community has studied for decades. We hope that

VOID will serve as a platform to develop models that can handle challenging motion and further foster fusion of multi-sensory data. Furthermore, we show that deep learning can be applied to predict the dense reconstruction from extremely sparse point clouds (e.g. features tracked by SLAM). We also show that we can improve the performance of our model by directly using pose from a SLAM system instead of pose network. These findings motivate a possible marriage between SLAM and deep learning that can benefit one another.

CHAPTER 6

Discussion

In this thesis, we detailed four approaches for developing visioning systems for an agent to learn to infer dense depth from images and, if available, sparse depth. In hopes of achieving full autonomy, all of the proposed approaches exploit epipolar geometry and principles of structure from motion (SFM) to enable continuous learning without any form of explicit supervision.

In Chapter 2, we proposed an adaptive weighting scheme that examines the data fidelity residuals in determination of the degree of regularization to impose on a model. This adaptive weighting scheme varies in both the spatial and temporal domain through an annealing process that allows the model to first maximize data fidelity and later apply regularity. The method incurs no extra parameters. To improve the learning process, we further proposed a two-branch decoder that first minimizes the data fidelity term alone in one branch then provide the second branch with the coarse predictions and features to produce the final predictions. In the process of developing this architecture, we effectively removed $\approx$ 10 million parameters from the model, making it faster and smaller while improving performance. We showed the benefits of using adaptive regularization on the KITTI [GLU12] and Make3D [SSN09] benchmarks.

Although we do see consistent improvement in our model after applying our adaptive regularization scheme, the adaptation only takes into account of a single image, the stereo-counterpart. While this is due to the nature of the training, minimizing a surrogate loss that involves reconstructing a stereo counter-part, we believe that this can be extended to multiple frames. In the case of multiple views, the reconstruction residual from the additional frames may be combined (e.g. the mean residual over multiple frames) to better inform the adaptive weighting scheme. In addition, when computing the global residual, it is done over a single image, making the degree of regularity different from image to image. Although one may argue that this allows more flexibility in the adaptation scheme, it also result in more noise in the adaptation. We believe that computing a true global mean may produce better performance.

In Chapter 3, we presented a method for exploiting gravity as a supervisory signal. As we assume that we are given an IMU as part of the system, we can obtain the direction of gravity without the need for dead-reckoning. We note that gravity influences the pose of some objects, but not all; therefore, we required a semantic segmentation module to provide a per pixel labeling on the scene. We then applied our shape model (horizontal and vertical planes) based on the object class and whether they should leverage the horizontal or the vertical plane prior. In our experiments on KITTI [GLU12], Make3D [SSN09], and VISMA2 (derived from VOID [WFS19]), we saw consistent improvements by leveraging our pose and shape priors and were able to improve worse performing methods to state-of-the-art performance. While we required a separate pose and semantic segmentation network during training, both of these networks are not needed during test time. Hence, our method incur no additional parameters. We showed that our framework is generic (all we need is an IMU and a semantic segmentation network) and that we can apply it to an array of models.

While our approach did improve other models, we would like to note that it is still a proof of concept. There are many avenues that we left unexplored. We only make use of gravity at training time. While this allowed our models to take in any stream, most device have both camera and IMU. Hence, we could in fact exploit the benefits of an IMU, for instance, to obtain the canonical frame at test time. Moreover, we used a very simple shape model consisting of horizontal and vertical frames. We did not address objects that require both horizontal and vertical plane priors. This opens opportunities to explore more sophisticated shape models (e.g. b-splines). Currently, we pre-defined the classes for applying horizontal and vertical plane priors. Perhaps a more general approach would be to learn the object classes that should map to a given shape prior.

In Chapter 4, we proposed to train a conditional prior network for the purpose of regularizing our predictions. As the prior network is trained using ground-truth annotations from a synthetic dataset, it contains semantics as well as contextual information, enabling the prior network to successfully guide regularization. We additionally proposed a novel network architecture, following the late fusion paradigm. Our supervised model, to this day, still holds the best MAE and iMAE on the online KITTI [USS17] depth completion benchmark. This is largely attributed to our network structure, two deep ResNet encoders that process image and sparse depth separately. The successes of our model in this work are also its drawbacks. While having a trained prior provides better regularization, it lacks applicability outside of the domain of the prior. A prior network trained in the outdoor domain would not be able to generalize to the indoor domain, even if we have extra data. While one may argue that we can simply train an additional prior network for a different domain, this is not scalable and would prevent us from continuously learning in any given novel environment – defeating the strength of unsupervised learning. In regards to speed and memory, our architecture contains two ResNet encoders, doubling the number of layers on the encoder side, and causing training and testing to be slow. It also takes up large amounts of memory rendering training to be difficult on a standard desktop setup. We restricted ourselves to smaller batch size, which in turn produces noisier gradients. This additionally limited the type of devices our method can be used on. Motivated by these drawbacks, we developed a small and computationally cheap approach that our outperforms this work in the unsupervised setting.

Conscious of the drawbacks in Chapter 4, we proposed a two-stage approach of approximate and refine for the purpose of depth completion in Chapter 5. We begin by computing the scaffolding of the scene based on the sparse depth measurements by Delaunay triangulation. The tessellations of the scaffoldings were produced by linear interpolation within the Barycentric coordinates. In the case where three points lie on the same plane, linear interpolation produces the optimal solution. However, as points are sparse, such is rarely the case. Hence, we proposed a light-weight network to refine the coarse approximation. The network, containing fewer than half of the parameters of standard depth completion networks, takes an image and its associated scaffolding as input and predicts dense depth. The use of the scaffolding is not only to approximate the scene, but also to propagate depth to missing values. This alleviates the burden of needing to process the sparse input and thus allowing a small network to outperform much bigger ones [MCK19, SNC19, YWS19, WFS19].

Although the system performs quite well when compared to current depth completion methods, its performance is still limited by the optimization process as it utilizes a uniform weighting scheme for regularization. It also does not account for occlusions and disocclusions. Hence, we return to our conundrum: to iteratively find correspondences without knowing if the regions are actually co-visible and imposing regularity with the hope that that we have applied it to the right place, at the right time, with the right amount. This begs for an adaptive approach to learning depth completion.

In regards to utilizing classical methods in the ages of deep learning: as discussed in Chapter 5, our two-stage approach fuses together classical and deep learning techniques and shows that it is possible for a happy marriage between the two. This is a specific approach for the depth completion problem, but the framework of approximating and refining is general. For instance, such can be extended to pose estimation. We can estimate the pose from a sequence of images and calibration, both of subjected to noise and nuisance factors, and refine the result using a deep network – in order words, to model the factors that are not considered by classical methods. Moreover, pose does not need to be learned – this is well-known, given a set of distinctive features, SFM and SLAM systems are able to localize themselves in an environment. In this age of deep learning, we are increasing the size of our networks, and are constantly hungry for more data. Yet, we know how to solve many of the tasks at hand, without the need to learn from additional data. Perhaps incorporating known processes and regularities will allow us to re-approach problems from a different perspective – one that is less data hungry and does not need to re-learn the known regularities of our visual world.

REFERENCES

[ABC16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. "TensorFlow: A System for Large-Scale Machine Learning." In *OSDI*, volume 16, pp. 265–283, 2016.

[AK17] Brandon Amos and J Zico Kolter. "Optnet: Differentiable optimization as a layer in neural networks." In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 136–145. JMLR. org, 2017.

[ATP18] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. "Generative Adversarial Networks for unsupervised monocular depth prediction." In *15th European Conference on Computer Vision (ECCV) Workshops*, 2018.

[BDD96] C Bradford Barber, David P Dobkin, David P Dobkin, and Hannu Huhdanpaa. "The quickhull algorithm for convex hulls." *ACM Transactions on Mathematical Software (TOMS)*, **22**(4):469–483, 1996.

[Bro79] Kevin Q Brown. "Voronoi diagrams from convex hulls." *Information processing letters*, **9**(5):223–228, 1979.

[CFY16] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. "Single-image depth perception in the wild." In *Advances in Neural Information Processing Systems*, pp. 730–738, 2016.

[COR16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. "The cityscapes dataset for semantic urban scene understanding." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[CS12] Massimo Camplani and Luis Salgado. "Efficient spatio-temporal hole filling strategy for kinect depth maps." In *Three-dimensional image processing (3DIP) and applications Ii*, volume 8290, p. 82900E. International Society for Optics and Photonics, 2012.

[CWL18] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. "Deep Convolutional Compressed Sensing for LiDAR Depth Completion." *arXiv preprint arXiv:1803.08949*, 2018.

[CWY18] Xinjing Cheng, Peng Wang, and Ruigang Yang. "Depth Estimation via Affinity Learned with Convolutional Spatial Propagation Network." In *European Conference on Computer Vision*, pp. 108–125. Springer, Cham, 2018.

[DVP18] Martin Dimitrievski, Peter Veelaert, and Wilfried Philips. "Learning Morphological Operators for Depth Completion." In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 450–461. Springer, 2018.

[EF15] David Eigen and Rob Fergus. “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture.” In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650–2658, 2015.

[EFK18] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. “Propagating Confidences through CNNs for Sparse Data Regression.” *arXiv preprint arXiv:1805.11913*, 2018.

[EKC18] Jakob Engel, Vladlen Koltun, and Daniel Cremers. “Direct sparse odometry.” *IEEE transactions on pattern analysis and machine intelligence*, **40**(3):611–625, 2018.

[EPF14] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network.” In *Advances in neural information processing systems*, pp. 2366–2374, 2014.

[FB81] Martin A Fischler and Robert 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.

[FGW18] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. “Deep Ordinal Regression Network for Monocular Depth Estimation.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2002–2011, 2018.

[FNP16] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. “Deepstereo: Learning to predict new views from the world’s imagery.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5515–5524, 2016.

[FS18] Xiaohan Fei and Stefano Soatto. “Visual-inertial object detection and mapping.” In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 301–317, 2018.

[FWS19] Xiaohan Fei, Alex Wong, and Stefano Soatto. “Geo-supervised visual depth prediction.” *IEEE Robotics and Automation Letters*, **4**(2):1661–1668, 2019.

[GBC16] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. “Unsupervised cnn for single view depth estimation: Geometry to the rescue.” In *European Conference on Computer Vision*, pp. 740–756. Springer, 2016.

[GK92] Nikolas P Galatsanos and Aggelos K Katsaggelos. “Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation.” *IEEE Transactions on image processing*, **1**(3):322–336, 1992.

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for autonomous driving? the kitti vision benchmark suite.” In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3354–3361. IEEE, 2012.

[GMB17] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. “Unsupervised monocular depth estimation with left-right consistency.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279, 2017.

[GPM14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial nets.” In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[GWC16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. “Virtual worlds as proxy for multi-object tracking analysis.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4340–4349, 2016.

[HEH07] Derek Hoiem, Alexei A Efros, and Martial Hebert. “Recovering surface layout from an image.” *International Journal of Computer Vision*, **75**(1):151–172, 2007.

[HFY18] Zixuan Huang, Junming Fan, Shuai Yi, Xiaogang Wang, and Hongsheng Li. “HMS-Net: Hierarchical Multi-scale Sparsity-invariant Network for Sparse Depth Completion.” *arXiv preprint arXiv:1808.08685*, 2018.

[HHY19] Tong He, Haibin Huang, Li Yi, Yuqian Zhou, Chihao Wu, Jue Wang, and Stefano Soatto. “GeoNet: Deep Geodesic Networks for Point Cloud Analysis.” *arXiv preprint arXiv:1901.00680*, 2019.

[HKB17] Byung-Woo Hong, Ja-Keoung Koo, Martin Burger, and Stefano Soatto. “Adaptive regularization of some inverse problems in image analysis.” *arXiv preprint arXiv:1705.03350*, 2017.

[HKD17] Byung-Woo Hong, Ja-Keoung Koo, Hendrik Dirks, and Martin Burger. “Adaptive Regularization in Convex Composite Optimization for Variational Imaging Problems.” In *German Conference on Pattern Recognition*, pp. 268–280. Springer, 2017.

[HKJ13] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. “Pm-huber: Patchmatch with huber regularization for stereo matching.” In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 2360–2367. IEEE, 2013.

[HS19] Tong He and Stefano Soatto. “Mono3D++: Monocular 3D Vehicle Detection with Two-Scale 3D Hypotheses and Task Priors.” *arXiv preprint arXiv:1901.03446*, 2019.

[HZC13] Christian Hane, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. “Joint 3D scene reconstruction and class segmentation.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 97–104, 2013.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[JCW18] Maximilian Jaritz, Raoul de Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. "Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation." In *International Conference on 3D Vision (3DV)*, 2018.

[JGB17] Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art." *arXiv preprint arXiv:1704.05519*, 2017.

[JGK17] Omid Hosseini Jafari, Oliver Groth, Alexander Kirillov, Michael Ying Yang, and Carsten Rother. "Analyzing modular cnn architectures for joint depth prediction and semantic segmentation." In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4620–4627. IEEE, 2017.

[JS11] Eagle S Jones and Stefano Soatto. "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach." *The International Journal of Robotics Research*, **30**(4):407–430, 2011.

[JSZ15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. "Spatial transformer networks." In *Advances in neural information processing systems*, pp. 2017–2025, 2015.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*, 2014.

[KGC15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. "Posenet: A convolutional network for real-time 6-dof camera relocalization." In *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946, 2015.

[KHW18] Jason Ku, Ali Harakeh, and Steven L Waslander. "In defense of classical image processing: Fast depth completion on the cpu." In *2018 15th Conference on Computer and Robot Vision (CRV)*, pp. 16–22. IEEE, 2018.

[KK11] Philipp Krähenbühl and Vladlen Koltun. "Efficient inference in fully connected crfs with gaussian edge potentials." In *Advances in neural information processing systems*, pp. 109–117, 2011.

[KLD14] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M Rehg. "Joint semantic segmentation and 3d reconstruction from monocular video." In *European Conference on Computer Vision*, pp. 703–718. Springer, 2014.

[KLK12] Kevin Karsch, Ce Liu, and Sing Bing Kang. "Depth extraction from video using non-parametric sampling." In *European Conference on Computer Vision*, pp. 775–788. Springer, 2012.

[KPS16] Seungryong Kim, Kihong Park, Kwanghoon Sohn, and Stephen Lin. "Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields." In *European Conference on Computer Vision*, pp. 143–159. Springer, 2016.

[KSH14] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. "Automatic scene inference for 3d object compositing." *ACM Transactions on Graphics (TOG)*, **33**(3):32, 2014.

[KWI13] Janusz Konrad, Meng Wang, Prakash Ishwar, Chen Wu, and Debargha Mukherjee. "Learning-based, automatic 2D-to-3D image and video conversion." *IEEE Transactions on Image Processing*, **22**(9):3485–3496, 2013.

[LLS15] Ian Lenz, Honglak Lee, and Ashutosh Saxena. "Deep learning for detecting robotic grasps." *The International Journal of Robotics Research*, **34**(4-5):705–724, 2015.

[LMF09] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. "Epnp: An accurate o (n) solution to the pnp problem." *International journal of computer vision*, **81**(2):155, 2009.

[LRB16] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. "Deeper depth prediction with fully convolutional residual networks." In *3D Vision (3DV), 2016 Fourth International Conference on*, pp. 239–248. IEEE, 2016.

[LRL14] Si Lu, Xiaofeng Ren, and Feng Liu. "Depth enhancement via low-rank matrix completion." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3390–3397, 2014.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[LSL15] Fayao Liu, Chunhua Shen, and Guosheng Lin. "Deep convolutional neural fields for depth estimation from a single image." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5162–5170, 2015.

[LSL16] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. "Learning depth from single monocular images using deep convolutional neural fields." *IEEE transactions on pattern analysis and machine intelligence*, **38**(10):2024–2039, 2016.

[LSP14] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. "Pulling things out of perspective." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 89–96, 2014.

[LYC18] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. "Planenet: Piece-wise planar reconstruction from a single rgb image." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2579–2588, 2018.

[MAS13] Jonathan Masci, Jesús Angulo, and Jürgen Schmidhuber. "A learning framework for morphological operators using counter–harmonic mean." In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pp. 329–340. Springer, 2013.

[MCK19] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. "Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera." In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[MR07] Anastasios I Mourikis and Stergios I Roumeliotis. "A multi-state constraint Kalman filter for vision-aided inertial navigation." In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3565–3572. IEEE, 2007.

[MSK12] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012.

[MWA18] Reza Mahjourian, Martin Wicke, and Anelia Angelova. "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5667–5675, 2018.

[NMG01] Nhat Nguyen, Peyman Milanfar, and Gene Golub. "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement." *IEEE Transactions on image processing*, **10**(9):1299–1308, 2001.

[PGA16] Cristiano Premebida, Luis Garrote, Alireza Asvadi, A Pedro Ribeiro, and Urbano Nunes. "High-resolution LIDAR-based depth mapping using bilateral filter." *arXiv preprint arXiv:1606.05614*, 2016.

[PHC15] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. "Spatio-temporal video autoencoder with differentiable memory." *arXiv preprint arXiv:1511.06309*, 2015.

[PSD17] Bernd Pfrommer, Nitin Sanket, Kostas Daniilidis, and Jonas Cleveland. "Penncosyvio: A challenging visual inertial odometry benchmark." In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3847–3854. IEEE, 2017.

[QLL18] Xiaojun Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. "GeoNet: Geometric Neural Network for Joint Depth and Surface Normal Estimation." In *CVPR*, 2018.

[RKT09] Chris Russell, Pushmeet Kohli, Philip HS Torr, et al. "Associative hierarchical crfs for object class image segmentation." In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 739–746. IEEE, 2009.

[RPY18] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. "SBNet: Sparse Blocks Network for Fast Inference." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8711–8720, 2018.

[RT16] Anirban Roy and Sinisa Todorovic. "Monocular depth estimation using neural regression forest." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5506–5514, 2016.

[SC13] Ju Shen and Sen-Ching S Cheung. "Layer depth denoising and completion for structured-light rgb-d cameras." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1187–1194, 2013.

[SCN06] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. "Learning depth from single monocular images." In *Advances in neural information processing systems*, pp. 1161–1168, 2006.

[SGD18] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry." In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1680–1687. IEEE, 2018.

[SHK12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. "Indoor segmentation and support inference from rgbd images." In *European Conference on Computer Vision*, pp. 746–760. Springer, 2012.

[SJC08] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. "Semantic texton forests for image categorization and segmentation." In *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8. IEEE, 2008.

[SNC19] Shreyas S Shivakumar, Ty Nguyen, Steven W. Chen, and Camillo J Taylor. "DFuseNet: Deep Fusion of RGB and Sparse Depth Information for Image Guided Dense Depth Completion." *arXiv preprint arXiv:1902.00761*, 2019.

[SSC11] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. "Real-time visual odometry from dense RGB-D images." In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 719–722. IEEE, 2011.

[SSK13] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. "Real-time human pose recognition in parts from single depth images." *Communications of the ACM*, **56**(1):116–124, 2013.

[SSN09] Ashutosh Saxena, Min Sun, and Andrew Y Ng. "Make3d: Learning 3d scene structure from a single still image." *IEEE transactions on pattern analysis and machine intelligence*, **31**(5):824–840, 2009.

[SSP10] Danail Stoyanov, Marco Visentini Scarzanella, Philip Pratt, and Guang-Zhong Yang. "Real-time stereo reconstruction in robotically assisted minimally invasive surgery." In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 275–282. Springer, 2010.

[SSP16] Nick Schneider, Lukas Schneider, Peter Pinggera, Uwe Franke, Marc Pollefeys, and Christoph Stiller. "Semantically guided depth upsampling." In *German Conference on Pattern Recognition*, pp. 37–48. Springer, 2016.

[SZ14] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*, 2014.

[TCS15] Konstantine Tsotsos, Alessandro Chiuso, and Stefano Soatto. "Robust inference for visual-inertial sensor fusion." In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

[USS17] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. "Sparsity invariant cnns." In *2017 International Conference on 3D Vision (3DV)*, pp. 11–20. IEEE, 2017.

[UZU17] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. "Demon: Depth and motion network for learning monocular stereo." In *IEEE Conference on computer vision and pattern recognition (CVPR)*, volume 5, p. 6, 2017.

[WBS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. "Image quality assessment: from error visibility to structural similarity." *IEEE transactions on image processing*, **13**(4):600–612, 2004.

[WBZ18a] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. "Learning Depth from Monocular Videos using Direct Methods." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2022–2030, 2018.

[WBZ18b] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. "Learning Depth from Monocular Videos using Direct Methods." In *CVPR*, 2018.

[WCP09] Andreas Wedel, Daniel Cremers, Thomas Pock, and Horst Bischof. "Structure- and motion-adaptive regularization for high accuracy optic flow." In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1663–1668. IEEE, 2009.

[WFS19] Alex Wong, Xiaohan Fei, and Stefano Soatto. "VOICED: Depth Completion from Inertial Odometry and Vision." *arXiv preprint arXiv:1905.08616*, 2019.

[WHS19] Alex Wong, Byung-Woo Hong, and Stefano Soatto. "Bilateral Cyclic Constraint and Adaptive Regularization for Unsupervised Monocular Depth Prediction." *arXiv preprint arXiv:1903.07309*, 2019.

[WSR16] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. "SURGE: Surface Regularized Geometry Estimation from a Single Image." In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pp. 172–180. Curran Associates, Inc., 2016.

[WTP09] Manuel Werlberger, Werner Trobin, Thomas Pock, Andreas Wedel, Daniel Cremers, and Horst Bischof. "Anisotropic Huber-L1 Optical Flow." In *BMVC*, volume 1, p. 3, 2009.

[XGF16] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks." In *European Conference on Computer Vision*, pp. 842–857. Springer, 2016.

[XRO17] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation." In *Proceedings of CVPR*, volume 1, 2017.

[XWT18] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. "Structured attention guided convolutional neural fields for monocular depth estimation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3917–3925, 2018.

[YLS15] Yanchao Yang, Zhaojin Lu, and Ganesh Sundaramoorthi. "Coarse-to-fine region selection and matching." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5051–5059, 2015.

[YS17] Yanchao Yang and Stefano Soatto. "S2F: Slow-to-fast interpolator flow." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2087–2096, 2017.

[YS18a] Yanchao Yang and Stefano Soatto. "Conditional prior networks for optical flow." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 271–287, 2018.

[YS18b] Zhichao Yin and Jianping Shi. "GeoNet: Unsupervised Learning of Deep Depth, Optical Flow and Camera Pose." In *CVPR*, 2018.

[YWS18] Nan Yang, Rui Wang, Jörg Stückler, and Daniel Cremers. "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry." In *European Conference on Computer Vision*, pp. 835–852. Springer, 2018.

[YWS19] Yanchao Yang, Alex Wong, and Stefano Soatto. "Dense Depth Posterior (DDP) from Single Image and Sparse Range." *arXiv preprint arXiv:1901.10034*, 2019.

[ZBS17] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. "Unsupervised learning of depth and ego-motion from video." In *CVPR*, number 6, p. 7, 2017.

[ZF18] Yinda Zhang and Thomas Funkhouser. "Deep Depth Completion of a Single RGB-D Image." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 175–185, 2018.

[ZGW18] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. “Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 340–349, 2018.

[ZKA16] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. “Learning dense correspondence via 3d-guided cycle consistency.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 117–126, 2016.

[ZLH18] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. “DF-Net: Unsupervised Joint Learning of Depth and Flow using Cross-Task Consistency.” *arXiv preprint arXiv:1809.01649*, 2018.

[ZSQ17] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. “Pyramid scene parsing network.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

[ZTS16] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. “View synthesis by appearance flow.” In *European conference on computer vision*, pp. 286–301. Springer, 2016.

[ZVS18] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. “Learning transferable architectures for scalable image recognition.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.

[ZZP17] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. “Scene parsing through ade20k dataset.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.