2023-01-03

Scale Consistent RGB Predicted Depth SLAM with Applications to Augmented Reality

Nawal, Mehbas Fairuz

UNIVERSITY OF CALGARY

Scale Consistent RGB Predicted Depth SLAM with Applications to Augmented Reality

by

Mehbas Fairuz Nawal

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

CALGARY, ALBERTA

JANUARY, 2023

Abstract

Simultaneous Localization and Mapping (SLAM) based on visual sensors is a key technology for autonomous navigation and Augmented/Virtual Reality. The fundamental problem involves mapping of an unknown environment from sparse/dense depths, and concurrently localizing the sensor in the environment based on the map built from a sequence of images. Nevertheless, estimation of depth from a single camera is an ill-posed problem, which is why monocular SLAM systems often suffer from heavy scale drifts in long sequences, especially in the absence of loops for correction. With the rapid advancement in deep learning, neural networks have demonstrated noteworthy performance in single image depth estimation (SIDE). However, most unsupervised SIDE methods predict inconsistent depths over a video sequence, thereby limiting their application in SLAM systems. This thesis proposes an additional loss term that takes the spatial geometry of the scene into account during training to further constrain the network to produce scale consistent predictions over a sequence of images in an unsupervised manner. As the neural network learns from unlabeled monocular images, the predictions still suffer from per-frame uncertainty. Thus, the proposed SLAM approach (RGB-PD SLAM) using the CNN-based depth maps incorporates a novel optimization step using sparse bundle adjustment (SBA) to deal with the noises in the predictions. The potency of the proposed SLAM approach is validated through a simple AR application.

Acknowledgements

My first and foremost gratitude goes to Almighty Allah (SWT) for granting me strength and guidance for the completion of this thesis and for the countless other blessings He bestows upon me every day.

I would like to thank my supervisor and mentor, Dr. Henry Leung, for this opportunity and his invaluable advice and support throughout my studies. Special thanks to my committee members, Dr. Kangsoo Kim and Dr. Jay Carriere for taking the time to read and provide feedback for this thesis. I would also like to thank the NSERC-We-TRAC training program for funding a part of this research.

Besides, I am thankful to every individual in my research group, especially Gayan, Abbas, Dylan and Emre, for their critiques, suggestions, and words of encouragement. I am immensely grateful for my family-like friends in Calgary, particularly Adrita, Bushra, Daisy auntie, Karima, Mohsina, Morshed uncle, Nabil, Nibir, Nusrat, Raisa, Rakib, Ruksana auntie, Sabab, Samia, Shabab, Sreya, and Yousuf for their unconditional love and support.

I want to thank my parents, brother, grandfather, and in-laws, without whose prayers, love, support, and sacrifices I would not be here today. Lastly, I would like to express my heartfelt appreciation to my beloved husband, Dr. Nahiyan Faisal Azad, for supporting my decisions and withstanding every storm with me through thick and thin. Thank you.

List of Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

AR	Augmented Reality
BA	Bundle Adjustment
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DLT	Direct Linear Transformation
DOF	Degrees of Freedom
DVO	Direct Visual Odometry
GAN	Generative Adversarial Network
GPS	Global Positioning System
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit

LiDAR	Light Detection and Ranging
MRF	Markov Random Field
ORB	Oriented FAST and Rotated BRIEF
RANSAC	Random Sample and Consensus
RGB-PD	RGB-Predicted Depth
RNN	Recurrent Neural Network
ROI	Region of Interest
SBA	Sparse Bundle Adjustment
SE(3)	Special Euclidean Group in 3-Dimension
SFM	Structure from Motion
SIDE	Single Image Depth Estimation
SLAM	Simultaneous Localization and Mapping
SO(3)	Special Orthogonal Group in 3-Dimension

SSIM	Structural Similarity
VO	Visual Odometry
VR	Virtual Reality
vSLAM	Visual SLAM

1. Introduction and Background Study

Simultaneous Localization and Mapping (SLAM) [1]-[3] in robotics is the 3D reconstruction of an unknown scene, while estimating the position and orientation (pose) of a moving sensor within the scene. The interdependency between the sensor's location and the map around it, hence, makes it a chicken-egg problem. This technology has been extensively studied over a few decades, primarily to achieve the task of robot mapping, such as in unmanned aerial vehicles (UAV) [4] and consumer robots [5]. Later, SLAM-based applications became popular in computer-aided designs (CAD modeling), Augmented and Virtual Reality (AR/VR) applications, and intelligent transportation systems.

In early SLAM algorithms, different types of sensing techniques have been used for pose and depth estimation. In the early 1970s, the U.S. Department of Defense developed Global Position System (GPS) for timing and space-based navigation [6], but GPS can every so often be inaccurate due to its signal strength issue, especially in confined spaces. Inertial Measurement Units (IMU) [7] on mobile devices have also been used for tracking the devices' position and orientation, but IMUs are subject to drift over time as they calculate the pose by performing mathematical integration with respect to time. Laser range sensors such as LiDAR [8] measure time of flight to infer accurate but sparse measurements of the scene in real-time and can be used in both indoor and outdoor settings. However, they are expensive and suffer from performance degradation in sub-optimal weather conditions such as in rain and fog. Besides, they consume power inefficiently, thus limiting its usage in power-limited devices and small spaces like the human colon for medical imaging. Ultrasonic methods [9] are generally an inexpensive source of spatial sensing over the terrain, but the efficiency extensively depends on the acoustic reflectivity of the material and surface. Nonetheless, vision-based methods rapidly emerged and were

demonstrated to be more versatile in motion tracking despite their high computational complexity and need for more sophisticated algorithms to extract and process data.

A camera with a display is an appealing configuration as vision sensors are low-cost and ubiquitous, with long range and high resolution. Since the input is visual information only, the technology is referred to as *visual SLAM (vSLAM)*. Visual sensors are mainly three types: monocular, stereo, and RGB-D cameras. Such sensors that provide scale or depth measurements, like stereo or RGB-D cameras, have a limited scale range in which they perform well. Monocular cameras, however, are immune to this issue as they are inherently scale-invariant. Simultaneously, since the absolute scale from 2D images cannot be inferred in monocular SLAM, scale estimates suffer from drift accumulation with time. And because of the unknown scale factor associated with it, monocular SLAM suffers from scale inconsistency. Considering the above factors and with rapid development in deep learning, neural networks have shown significant promise towards single image depth estimation (SIDE) and gained prominence in various Computer Vision tasks including SLAM.

Supervised learning approaches [10]-[11] achieve superior performance but depend on high quality, pixel aligned, ground truth depth data at training time, which is challenging to acquire at scale. To address this problem, self-supervised learning has emerged as an alternative for SIDE that either uses calibrated stereo pairs [12]-[13] or unlabelled image sequences from videos [14]-[15] for training. These methods rely on geometric constraints between multiple views instead of ground truth to learn depth information. Considering that stereo-based learning specifically requires calibrated binocular images with a known transformation between them for training, video-based learning makes a good alternative as monocular images are more readily available. However, the latter is more challenging due to the unknown camera motion and still suffers from

scale ambiguity and inconsistency because the network predicts a "relative depth" for the world as opposed to the metric depth in the previous setting. Therefore, this thesis proposes an improved unsupervised learning framework with an additional geometric constraint to achieve more accurate and consistent depth predictions from a sequence of monocular images. Moreover, this research shows how more scale-consistent depth predictions lead to more accurate camera pose estimations, which is essential in AR.

1.1 Multiview Geometry

Understanding the scene context using cameras is a rudimentary problem in Computer Vision. An essential part of scene context from the perspective of AR is the recovery of scene geometry because of its direct relevance in associating virtual and real worlds. vSLAM techniques rely on multi-view geometric constraints to infer the camera pose and the scene geometry from a sequence of images.

1.1.1 Epipolar Geometry

Epipolar geometry [16, p. 239] is the projective geometry between two views. When a 3D scene is viewed by a camera or two different cameras (e.g., stereo configuration) from two distinct positions, the 3D points in the world reference frame and their respective 2D projections onto the images in pixels are connected via several geometric relations that lead to constraints between corresponding image points. The geometry is based on the camera's internal parameters (known as intrinsic parameters consisting of the focal length, optical centre and skew coefficient of the camera) and the relative pose $<R,t>$ between the frames. As illustrated in Figure 1.1, the standard epipolar geometry setup involves two camera views of the same 3D point P, whose projection on each of the image planes (left and right) in pixel unit is located at P_l and P_r respectively. The

camera centers are located at C_l and C_r, and the *baseline* between them is denoted by a translation, t. The plane defined by the two camera centers and P is the *epipolar plane*, π, and C_l, C_r, P_l, P_r and P are coplanar. The rays projected from P_l and P_r intersect at P and lie on π. It is this property that is of importance while searching for pixel correspondences. The points where the baseline intersects the left and right images are the *epipoles* denoted by e_l and e_r respectively. In this geometry there lies a constraint such that the point $P_{r\,r}$ corresponding to P_l lies on the epipolar line, I_{pr} (the line of intersection between the image plane and π) and is essentially the point where the ray from P intersects I_{pr} in the right-hand view. The benefit of the two-view setup is that the search for the point corresponding to P_r need not cover the entire image plane but can be limited to the line I_{pr}.

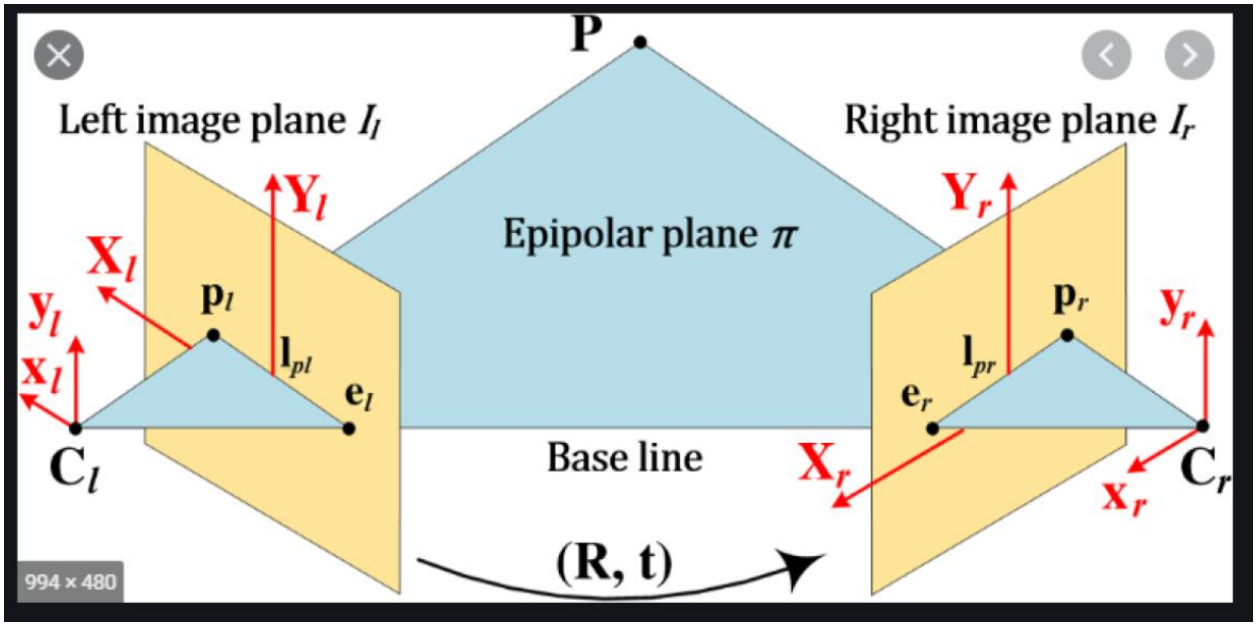

Figure 1.1: The Epipolar Geometry

As the position of the 3D point P in Figure 1.2b varies, the epipolar plane rotates about the baseline. All epipolar lines intersect at the epipole. Figure 1.2a shows the importance of the two-view geometry in depth perception. The point P can lie anywhere on the ray projecting from P_1 to

the 3D point in the world, thereby giving rise to different possibilities for the depth value (such as P_1, P_2, P_3). A second frame of the same scene disambiguates this uncertainty via the epipolar constraint.

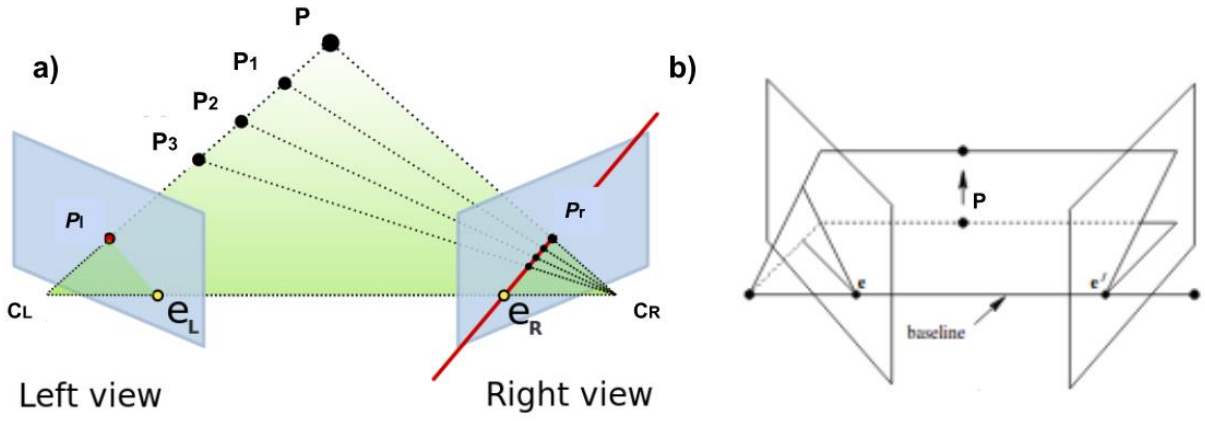

Figure 1.2: Importance of two view geometry in depth perception [16].

1.1.2 Stereo Vision

In a stereo rig setup (as shown in Figure 1.3), two calibrated cameras (with optical centers at C_L and C_R) separated by a known baseline are employed to view the same scene to detect a common 3D point P from different 2D locations. As the cameras are positioned parallel to each other along the x-axis, a point P_L in the left image will differ from its corresponding point P_R in the right image only along the width of the image. The difference in pixels along the x-axis between these two points is known as the disparity ($x_L - x_R$), which is used to calculate the depth, Z of the 3D point, provided the camera intrinsic parameters are known which can be easily obtained through an off-line calibration step.

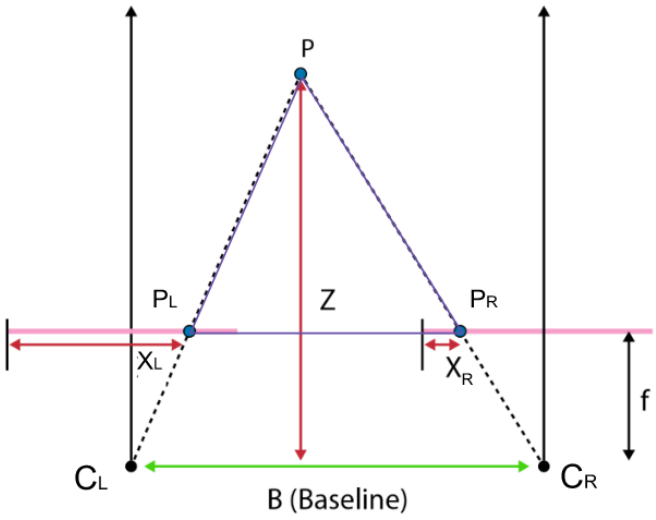

Figure 1.3: A Stereo Setup

The depth, Z of the 3D point P can be calculated in metric scale using similar triangles as:

$$\frac{B}{Z} = \frac{B-(x_L-x_R)}{Z-f}, \tag{1.1}$$

$$Z = \frac{fB}{x_L-x_R}, \tag{1.2}$$

where, f is the focal length of the camera and B is the baseline in meters.

1.2 Problem Outline

Most monocular SIDE methods only consider temporal constraints for training, where a geometric constraint between two consecutive color images is imposed to minimize a photometric reconstruction loss after warping one image to the other. This means that different scene scales are equally likely from the network's perspective. A change only in the scale of a depth map results in the same 2D image, and thus, the photometric loss of two images whose depth maps only differ in scale will be the same. For instance, an image of a cube with each side measuring 5 meters captured from 10 meters away will appear identical to an image of a cube with sides of 10 meters taken from 20 meters away. This leads to scale-inconsistent depth predictions over different

frames, resulting in a globally inconsistent map, which does not impact single image-based tasks but is crucial for video-based applications such as AR and Visual Odometry (VO) as shown in Figure 1.4. In such cases, the only difference is the regularization term in the loss function which is used for the smoothness of the predicted map (further discussed in section 3.1.3). The term is formulated with derivatives of the inverse depth maps and as the goal is to minimize this loss, a smaller inverse depth scale will always result in a smaller loss value. As a result, the network learns to predict smaller and smaller scaled depth maps and eventually the training diverges. Consequently, when the neural network-based depth is incorporated in a VO system, the estimated trajectory (green) in the figure differs significantly in scale from the ground-truth trajectory (red) as shown in the figure below.

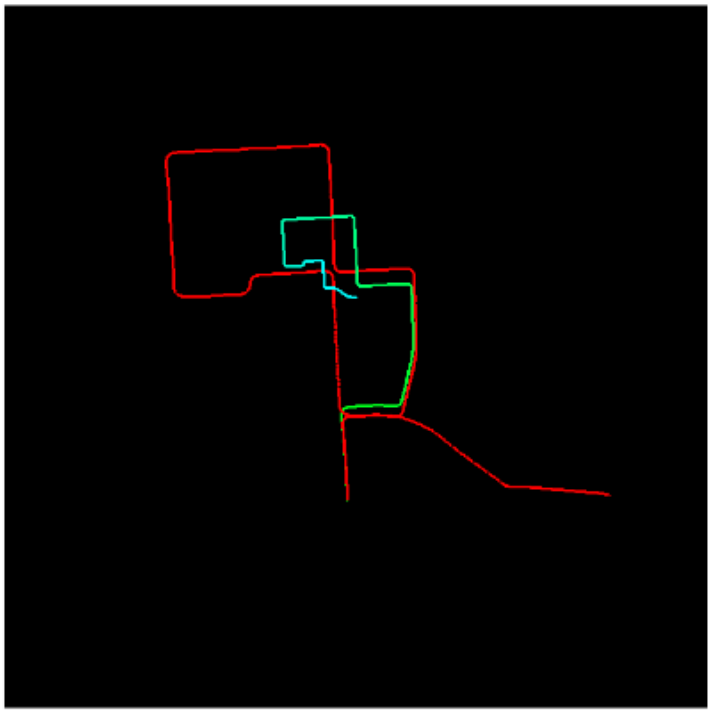

Figure 1.4: Testing visual odometry on KITTI sequence 00 using the predicted depth from [14].

1.3 Thesis Statement and Motivation

To address the scale inconsistency issue in depth predictions using unsupervised learning from video sequences, this thesis constraints the framework in [14] using a geometric loss, which minimizes the difference between the predicted depth and transformation and the corresponding synthesized ones. The smoothness loss in [14] nearly converges to 0 as shown in Figure 1.5a. Adding this spatial constraint alongside the temporal constraints also prevents the network from predicting smaller inverse depths, therefore causing the smoothness loss to increase (Figure 1.5b) and result in sharper depth maps (as shown in Figure 3.8). Later, the scale consistent predicted depths are integrated into a visual monocular SLAM system, namely ORB SLAM [17] using the KITTI dataset [18]. Although, this boosts the robustness and accuracy of the odometry components compared to traditional monocular SLAM systems, the learned depth exhibits significant noise during 3D reconstruction from multiple frames, i.e., the predictions still suffer from per frame uncertainty due to the lack of metric data during training. As a result, a novel optimization step is introduced in the SLAM framework using probabilistic approach to handle the uncertainty in the predictions, where the map points and keyframe poses are locally optimized while simultaneously determining which depth estimations are correct within a single integrated Bayesian framework.

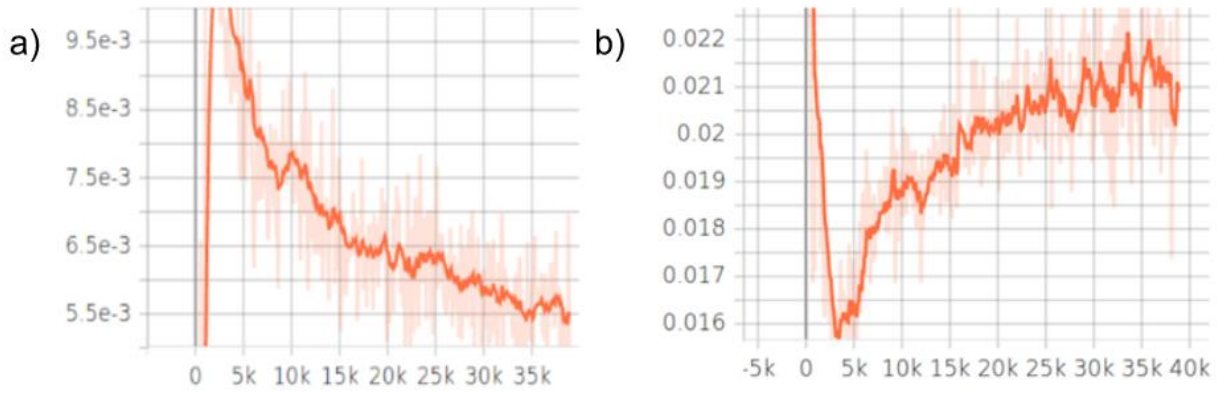

Figure 1.5: The effect of the added geometric constraint on smoothness loss.

1.4 Thesis Outline

- Chapter 2 reviews the SIDE problem and the current state-of-the-art methods for solving it. The effects of different sensors in training the depth network are reviewed comparing their advantages and disadvantages. The chapter discusses how limitations of SIDE methods have been modeled in the literature, especially in achieving scale consistent depth predictions in unsupervised learning. Lastly, different SLAM systems that have been integrated with a few of the discussed prediction models are revisited.

- Chapter 3 introduces the proposed geometric constraint to achieve scale-consistent depth predictions. The effects of the added constraint are evaluated and compared with previous unsupervised and semi-supervised methods in both indoor and outdoor settings. The depth results are compared with those from other vision-based sensors.

- Chapter 4 integrates the scale consistent depth predictions into a visual monocular SLAM system. A novel local optimization step called the sparse bundle adjustment (SBA) is introduced in the framework to handle noises in the predictions. Evaluation on the KITTI odometry dataset demonstrates that the proposed RGB-PD SLAM generates more consistent and accurate trajectories compared to monocular SLAM or SfM-Learner [14]. The importance of scale consistent pose estimation is shown using an application of Augmented Reality.

- Chapter 5 concludes this thesis. A summary of this work is presented, and future directions for this research are discussed.

2. Literature Review

Estimation of depth from a single image is a fundamentally ill-posed problem without having at least another image of the same scene from a different viewpoint to allow estimation of the epipolar geometry between them (section 1.1.1). Even so, humans can assume plausible depth estimates for novel scenes from single images using a priori information of the environment, and they do so by constantly learning as they navigate through and interact with the real world. Inspired by this, previous works estimate depth from single images by combining prior information and relating different structural cues such as the sky, buildings, and ground [19]. With rapid development in deep learning, neural networks have demonstrated remarkable performance in depth estimation alongside image classification, object detection and semantic segmentation. Extensive overviews on single image depth estimation (SIDE) can be found in [20]-[22].

2.1 How Neural Networks Perceive Depth

Research on (SIDE) using deep neural networks was first conducted by Eigen et al. [10], following which there has been active research in the field achieving high accuracy, which leads to the conclusion that these networks can indeed perceive depth from single images. But the question is: how? Monocular SIDE methods play an important role in comprehending spatial geometry. When only a single image is available, algorithms exploit depth cues, and according to [23] a study on human depth perception [24] provides insight into the following pictorial cues that could be used in perceiving depth from single views:

- Position of objects in an image: Objects that are farther away are generally closer to the horizon. Besides, very far objects have less contrast.

- Occlusion: Occlusion provides necessary information about the order in which objects are placed in the image, but not the distance. Objects that are closer to the camera tend to block those behind them.
- Texture density: Textured surfaces closer to the camera appear more coarse-grained in the image.
- Linear perspective: Parallel lines in a 3D world seem to converge in the distance as they retreat to a vanishing point on the horizon into the image plane.
- Apparent size of objects: Objects that are closer to the camera appear bigger and vice versa.
- Changes in illumination and shading: Surfaces appear more luminous when their surface normal points towards the light source. Shading gives perspective to scenes and provides information about the change in depth within a surface (for instance, the edges of a cylinder are typically shaded to give it a round effect).
- Focus: Objects that lie in front or behind the focal plane appear blurred.
- Edge or outline of an object.

However, a thorough investigation in [23] using four previously published models shows that neural networks primarily depend on the vertical position of the objects from the ground contact point and less on their apparent size. The vertical position, in turn, depends on the camera's position and orientation (pose) with respect to the world. Moreover, these networks can successfully recognize objects that do not appear in the training set but rely on factors such as the presence of shadow under the object as well as textures and contrast between adjacent regions.

2.2 Effect of sensors in training the network

In order to increase the accuracy of estimation, different network structures, loss functions and training methods have been proposed over the years. One major challenge of SIDE with deep

learning is the lack of ground truth depth maps needed for training which is often impractical to acquire for a large variety of scenes. This section thus categorizes depth estimation methods from the aspect of using ground truth: supervised learning, unsupervised learning and semi-supervised learning, where each uses different visual sensory data for training. Even though the training process of unsupervised and semi-supervised methods depends on sets of multiple images (monocular video sequences or stereo image pairs), the trained model predicts depth maps for single RGB images during testing.

2.2.1 Supervised Monocular Depth Estimation

The supervisory signal of supervised methods depends on the ground truth of depth maps obtained from 3D visual sensors such as LiDAR or depth cameras so that depth estimation can be stated as a regressive problem due to the continuous property of depths. The difference between the ground truth depth and the inferred depth is used to supervise the training of networks, the L2 loss being:

$$\mathcal{L} = \frac{1}{N}\sum_{\rho \epsilon N}\left\| D_\rho - \widehat{D}_p \right\|^2, \tag{2.1}$$

where D_ρ is the predicted depth value of pixel ρ and $\widehat{D}_p$ stands for the corresponding ground truth value. *N* denotes the total number of pixels for which real depth data is available. Eigen et al. [10] were the first to show that dense, per-pixel depth maps can be estimated by means of a multi-scale deep neural network consisting of: a coarse-scale network that predicts the depth of the scene based on the entire image, and a fine-scale network that refines the predictions locally. Unlike most other prior works in SIDE, they do not rely on hand-crafted features or initial over-segmentation. Instead, they learn a representation directly from raw pixel values by training the network on images and their respective depth values. Several works followed inspired by this idea where they

propose to improve accuracy by using techniques like Markov Random Fields (MRF) [25], Conditional Random Fields (CRF) [26], changing supervisory signal from regression to classification [27], using more robust loss functions [28], and infusing strong scene priors to address problems related to surface normal estimation [29]. Nevertheless, these approaches depend on having high-quality, pixel aligned, ground truth depth data at training time.

2.2.2 Unsupervised Monocular Depth Estimation

To address the above issue, few works [14][15] have proposed to estimate depth based on neural networks that use epipolar geometric constraints between frames as the supervisory signal instead of requiring ground truth depth at training time. These methods are trained using monocular image sequences and geometric constraints are established based on the projection between neighbouring frames:

$$\rho_n \sim K T_{n\to n-1} D_n(\rho_n) K^{-1} \rho_n, \tag{2.2}$$

where ρ_n stands for a pixel on image I_n, and ρ_{n-1} refers to the corresponding pixel of ρ_n on image I_{n-1}. K is the camera intrinsic parameters matrix that transforms the camera coordinates to the image plane, and the inverse, likewise projects the pixels back to the camera coordinate frame. It is defined by:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

where (c_x, c_y) are the coordinates of the principal point (the intersection of the optical axes with the image plane) f_x and f_y are the ratio between the focal length f of the lens and the dimension of the pixel in the x and y directions respectively. $T_{n\to n-1}$ is the 3D transformation between images I_n

and I_{n-1}. $D_n(\rho_{n)}$ is the depth value at pixel ρ_n and thus if the depth and image transformation are known, the corresponding pixel on I_{n-1} can be estimated using Eq 2.2. Based on this constraint, Zhou et al. [14] jointly train a pose network to regress the transformation between target image I_n and a source image I_{n-1}, and a depth network to predict a depth map of the target image. The framework is shown in Figure 2.1.

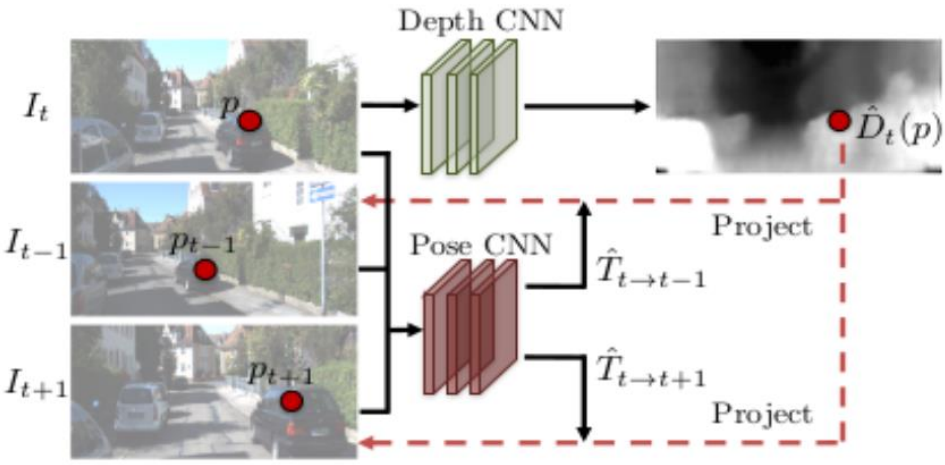

Figure 2.1: The SfM-Learner framework with unsupervised learning [14].

Following contemporary works [12][30], they use view reconstruction as the supervisory signal, where the loss is based on warping nearby views to the target using bilinear interpolation. A photometric error between the reconstructed image, $\hat{I}_t$ and the real image, I_t based on the L1 or L2 loss is computed as the key loss term (details in section 3.1.1):

$$\mathcal{L} = \sum_{\rho} \left| I_t(\rho) - \hat{I}_t(\rho) \right|_1, \tag{2.4}$$

Despite being trained together, the depth model and the pose model can be used independently at test time.

2.2.3 Semi-supervised Monocular Depth Estimation

Semi-supervised methods, on the other hand, are designed to obtain higher estimation accuracy with absolute scale information while reducing the dependence on expensive ground truth. Training on stereo image pairs instead of monocular videos benefits from the known transformation between the left and right images. Thus, some studies consider this framework as unsupervised methods, while others regard them as semi-supervised methods (because of the known camera pose). Such methods trained on stereo image pairs estimate the disparity maps (inverse depth maps) between the left and right images, which is then used to synthesize the left image from the right image by inverse warping. Like the unsupervised methods, the photometric difference between the reconstructed image $\hat{I}_l$ and the real image I_l is used as the supervisory signal to the constrain the training process:

$$\mathcal{L} = \sum_{\rho} \left| I_l(\rho) - \hat{I}_l(\rho) \right|_1 \tag{2.5}$$

$$\mathcal{L} = \sum_{\rho} |I_l(\rho) - I_r(\rho + d(\rho))|^2, \tag{2.6}$$

where I_r is the right image, I_l is the left image and $d(\rho)$ is the predicted disparity. This concept was first used by Garg et al. [31], where they also incorporated a smoothness loss to refine the coarse disparities and improve their continuity, as done by many supervised methods. Godard et al. [12] improved the final loss function by adding a left-right consistency term, which enforces consistency between the disparities produced relative to both the left and right images as shown in Figure 2.2. Garg's architecture was extended to predict left and right disparity maps together from only the left image and the generated maps were then used to reconstruct the left image from the right and vice versa. They also added SSIM to strengthen the structural similarity between the reconstructed images and real images. Zhou et al. [32] additionally regresses the pose between

temporally adjacent frames to increase the accuracy. However, collecting binocular stereo images still requires special instruments and acquiring a large number of readily available stereo images for training (especially in an indoor setting) is difficult. This is why, a lot of research has been done using monocular video sequences, including ours.

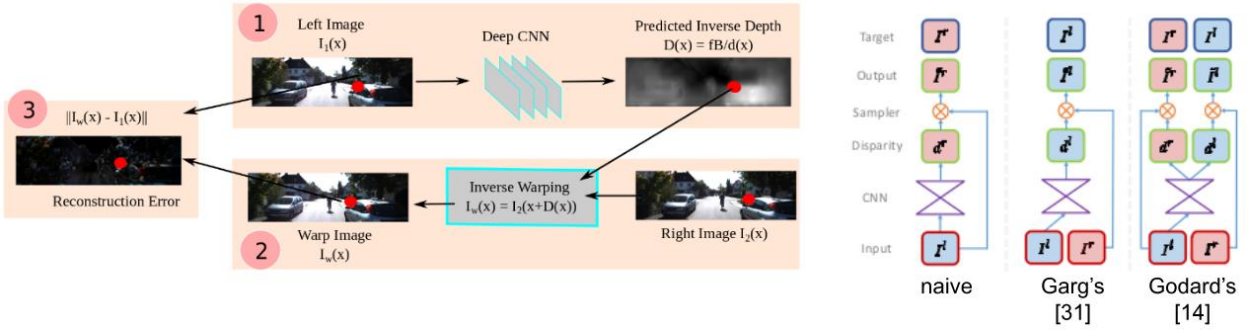

Figure 2.2: General framework of semi-supervised depth estimation using stereo image pairs.

2.3 Modeling different limitations of SIDE methods

2.3.1 Masking schemes

View reconstruction using interpolation works well when the position of objects in consecutive frames does not change and there is no occlusion between them. This means that the consecutive frames in the training sequence cannot be too far apart in time. To overcome this issue, most methods based on view synthesis mask out occlusions and non-rigid objects in the scene during training. Zhou et al. [14] designed an explainability mask network that predicts the target pixels for which view synthesis will be successful. On the downside, training a separate network for the mask increases the computational cost and thus makes the training strenuous. Therefore, Bian et al. [33] used geometry-based masks instead of predicting them. They restate that dynamic objects in a scene create a non-rigid flow, which cannot be explained by depth-based mapping, and that a stationary camera (e.g., two or more images that are captured at the same

position) creates identical flow that is independent to the depth prediction. Thus, they implement the binary mask, M_a proposed in [34] to remove invalid points in static pairs. It compares the photometric losses between the mapping by predictions (depth and pose) and the identity mapping, and removes the points where the identity mapping leads to a lower loss:

$$M_a(\rho) = \begin{cases} 1, & if\ |I_t - \hat{I}_t| < |I_t - I_s| \\ 0, & othewise \end{cases}, \tag{2.7}$$

Yin et al. [35] proposed Geo-Net to jointly estimate depth, pose and optical flow. The occlusion and non-Lambertian surfaces are identified as the regions where forward/backward of optical flows contradict to a great extent.

2.3.2 Combining SIDE with other vision-based tasks

Methods based on traditional visual odometry also exist in the literature. Some work [36]-[37] argue that the pose network employed by [14] does not completely exploit the relation between depth prediction and the camera pose. Therefore, instead of using a separate pose network, [36] directly regresses the pose from traditional direct visual odometry (DVO), which is then used to facilitate depth estimation. DVO takes the depth map generated by the network as well as a sequence of three consecutive frames to estimate the poses between them by minimizing a photometric error. The calculated poses are then sent back to the training network. Now that the depth network is supervised by more accurate poses, the quality of depth estimation is improved. Luo et al. [38] leverage geometric constraints extracted using conventional multi-view reconstruction methods and use them to fine-tune a pre-trained monocular SIDE network to produce more geometrically consistent depth estimates across an entire video.

Besides, techniques based on integrating SIDE with other vision-based tasks such as semantic image segmentation have also been explored to strengthen the loss function of the network [39]. Depth estimation and semantic image segmentation strongly correlate and are mutually beneficial tasks. Semantic labels allow depth predictions to be aligned to the actual size of the physical world, and inversely depth enhances the semantic estimation via increasing the capacity of the classifier to distinguish ambiguities. Ramirez et al. [40] proposed a semi-supervised framework (Figure 2.3), where they use semantically labeled images together with unsupervised signals based on the geometric constraint in Eq 2.3. A single encoder is shared between a depth and semantic decoder and a crossdomain discontinuity (L_{cdd}) term is introduced to reduce the error metrics. Similarly, [41] jointly performs depth and semantic predictions via neural networks, where the proposed improves feature extraction by separating features specific for one task from those which are common for both (depth and segmentation).

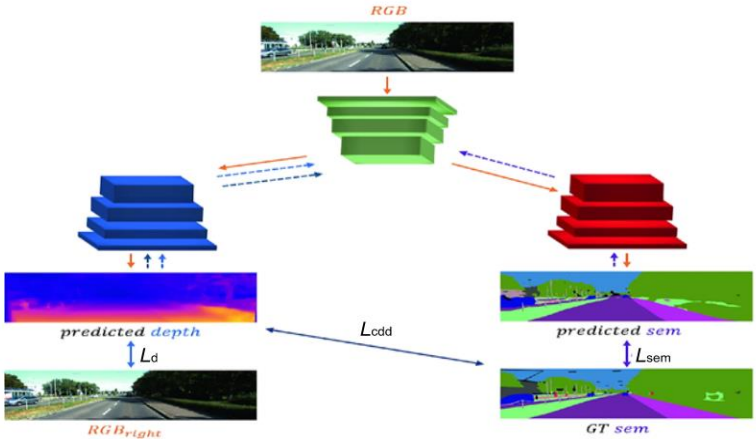

Figure 2.3: Schematic representation of the proposed architecture by [40].

The photometric loss often fails in challenging environments such as indoor areas with non-texture regions. Some works have imposed geometric constraints on optical flow to make predictions robust to dynamic and reflective objects, lack of texture, complex ego-motion of

handheld cameras and transparent surfaces [30][42]-[43]. SfM-Net [30] predicts the depth, segmentation, rigid body motions along with the camera extrinsic parameters to calculate the optical flow, which is then used to warp each frame to the next. Forward-backward consistency checks between estimated scene depth in different frames constrain the network. Zhou et al. [43] used an additional module to generate the optical flow to supervise the depth and pose prediction processes.

2.3.3 Use of additional information and constraints

Most SIDE methods formulate the optimization problem as either pixel-wise regression or classification. To improve the performance, some efforts have been made to enforce other constraints besides minimizing a pixel-wise error. To factor in higher order geometric constraints in 3D space, [44] designs a loss term that employs a geometric constraint, specifically, virtual normal directions determined by three randomly sampled points in the reconstructed scene. This leads to more accurate depth predictions that allows recovery of high-quality 3D features, such as point clouds and surface normals, directly from the depth, thus removing the need to train new sub models. Few work [45]-[46] apply geometric constraints based on surface normals. Surface normals and depth are mutually beneficial as surface normals are determined by the local surface tangent planes of the 3D points, and inversely depths are constrained by the local surface tangent plane determined by their surface normals.

Chen et al. [47] establish spatial and temporal constraints to resolve depth and pose scale factors using stereo pairs and neighbouring frames, which are enforced inside the supervisory reconstruction loss function during training time. They also predict the camera intrinsics alongside depth maps and ego-motion to reduce human effort and manual intervention. Mahjourian et al. [48] designed a 3D loss based on the consistency of point clouds generated from the estimated

depth of adjacent images and aligned the point clouds using Iterative Closest Point (ICP) algorithm. The loss comprises transition loss and registration error. Meanwhile, Prasad et al. [49] improve the photometric loss with epipolar constraints by adding weights to each pixel according to the fitness of epipolar geometry.

2.3.4 Scale consistent depth predictions

In comparison to supervised and semi-supervised methods, unsupervised methods rely on monocular images instead of the ground truth or calibrated binocular pairs to learn the depth information. Due to the lack of absolute scale information, unsupervised methods still suffer from scale ambiguity and inconsistency and thus the performance is inferior to that of supervised and semi-supervised methods. To solve the problem of inconsistent and diminishing depth (due to the scale-sensitive property of the regularization term in Eq 3.7), Wang et al. [36] proposed to apply a non-linear operator to normalize the output of the depth network before feeding it to the final loss term. The operator divides the inverse depth, d_i for the i^{th} pixel in the target image by its mean:

$$\eta(d_i) = \frac{N d_i}{\sum_{j=1}^{N} d_j}, \tag{2.8}$$

Closest to ours, Bian et al. [15][33] proposed a geometry consistency loss that penalizes the depths between adjacent frames, where the depth differences are normalized by their summation. They support that this normalization treats points at different absolute depths equally during optimization as well as makes the training more stable as the outputs are naturally ranging from 0 to 1. However, the outputs need to be scaled by the ground truth to be used in SLAM applications. Zhao et al. [50] constrains the cost function further by adding a structural dissimilarity term between the predicted and synthesized depths. However, implementation of the proposed strategy led to deteriorated results in our case. This is because the paper measures SSIM for interpolated

depth maps (whose values are not in order), whereas it is a measure of the similarity between two ordered images. Mahjourian et al. [48] penalize the misalignment of predicted depths using 3D ICP loss, where the gradients for depth and pose networks are estimated independently as ICP in non-differentiable. Zou et al. [51] propose a consistency loss that enforces corresponding points in two images to have identical depth predictions. However, in theory, this is incorrect as scene depth is view-dependent in the case of video-based learning. Yin et al. [52] recover the scale of ego-motion in VO by incorporating depths estimated from monocular images using deep convolutional neural fields, where Convolutional Neural Networks (CNN) are concatenated with CRF. The CNN regresses a coarse depth map from input images and the CRF refines the depth further, which aids in scale recovery and thus solves the scale drift problem in VO. Both depth refinement and scale recovery are done interactively. Zhao et al. [53] obtain the relative pose by solving the fundamental matrix from predicted optical flow correspondences and then recover an up-to-scale 3D structure via a two-view triangulation module. The scale of the predicted depth map is then aligned with the triangulated point cloud and the transformed depth map is used for depth error computation, thereby solving the scale inconsistency problem.

2.3.5 Frameworks and Architectures

Encoder-decoder networks are often used for regenerating the images in monocular depth estimation [10][12][14]. The encoder learns rich feature representation by increasing the receptive field of the network while reducing input dimension and computational cost. VGG [54] and ResNet50 [55] are popular choices of encoders. The decoder restores the original input resolution by means of up-sampling operators. Besides, Zhou et al. [14] point out that photometric loss is uninformative in texture-less regions and can be resolved using a convolutional encoder-decoder

architecture with a small bottleneck to constrain the output to be globally smooth and for gradients to propagate from meaningful regions to neighboring regions.

Recurrent Neural Networks (RNN) have been explored in depth predictions to understand the temporal coherence between adjacent video frames. Zhang et al. [56] take the advantage of convolutional long short-term memory (CLSTM) and propose a novel spatial-temporal CSLTM (ST-CLSTM) structure that can capture not only the spatial features but also the temporal correlations/consistency among consecutive video frames with slight increase in computational overhead. The incorporation of a conLSTM module into the depth and pose prediction networks in [57] (which are trained with re-projection error along with optical flow consistency) has shown to enhance the prediction performance in both supervised and unsupervised learning.

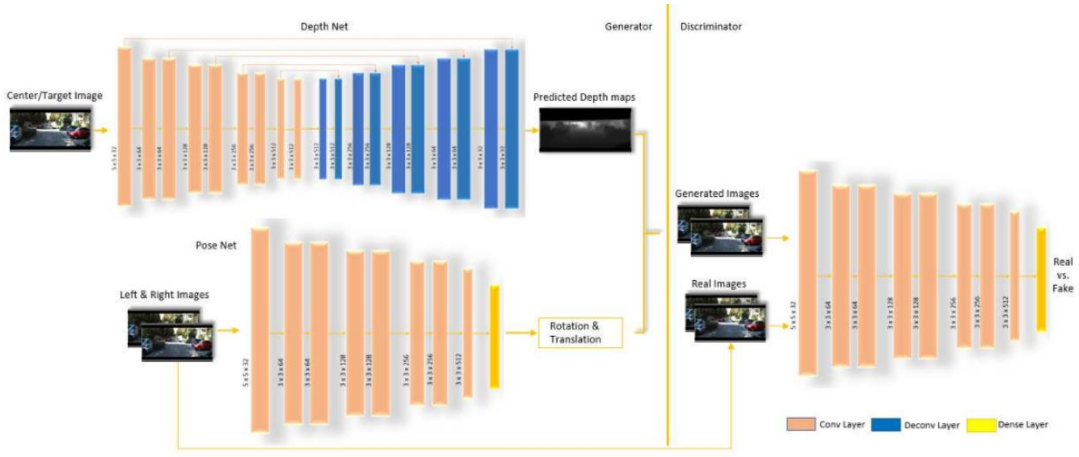

Figure 2.4: Overview of the unsupervised SIDE framework proposed by [60] using GAN.

Many SIDE methods [13] [58]-[61] leverage recent advances in adversarial learning for depth estimation as GANs are very useful where generating globally coherent images is vital. To introduce adversarial training into the process, a discriminator is used to distinguish between the real and reconstructed images. The confrontation between the generator and the discriminator networks facilitates the training of the framework and helps maintain the global coherence in the

predicted images. A vanilla GAN framework can be described as a min-max optimization process and the loss term can be defined as:

$$\mathcal{L}_{adv} = \min_{G} \max_{D} E_{x \sim P_{gt}}[logD(x)] + E_{\hat{x} \sim P_G}\big[\log\big(1 - D(\hat{x})\big)\big], \tag{2.9}$$

where x is the real image and $\hat{x}$ is the generated one using photometric reconstruction. Eq 2.9 is added to the final loss term during training of the networks. Kumar et al. [60] have shown that this approach works well in both supervised and unsupervised settings and their framework is shown in Figure 2.4. Pilzer et al. [58] further improved the predictions via a cycled generative network paradigm as shown in Figure 2.5.

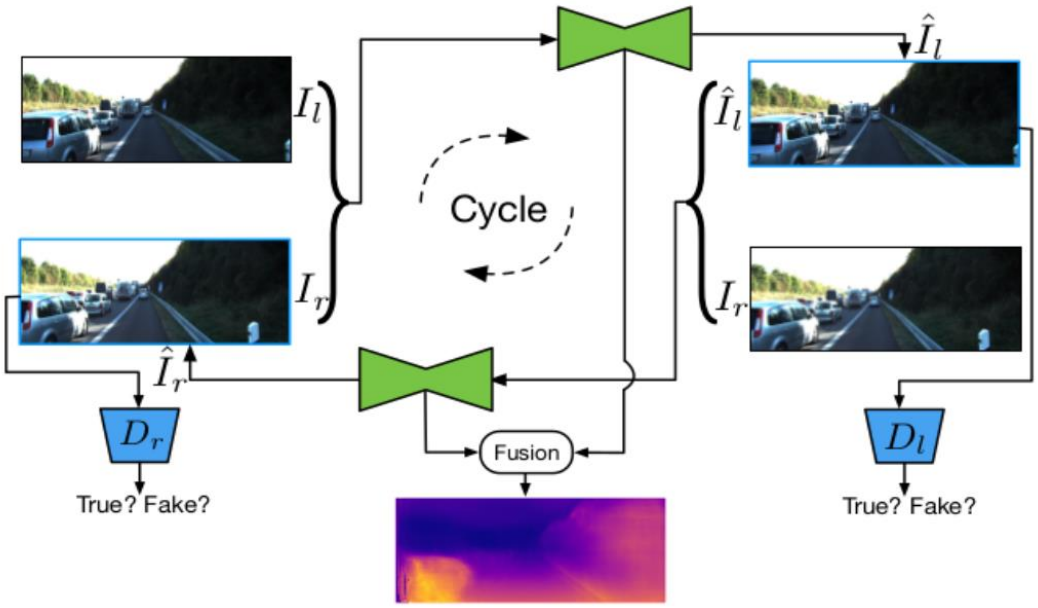

Figure 2.5: Overview of the cycled generative network for adversarial semi-supervised depth estimation using stereo pairs [58].

2.4 Integration of SLAM with predicted depth

Integration of SLAM with depth prediction via deep neural network is a promising direction to solve major limitations of traditional monocular SLAM in terms of estimating the

absolute scale, yielding dense reconstruction, especially along low textured regions and in cases of pure rotational motions. Greene et al. [62] proposed a monocular SLAM approach that combines metric information inferred from deep neural network with a factor graph-based geometric SLAM. The CNN is trained to predict metric depth from monocular images in 7 pyramid levels using calibrated stereo pairs as training data. They argue that the learned depth predictions only need to be coarse in image space for scale estimation so it is computationally tractable during real-time inference. At runtime, they divide the SLAM problem into a local visual odometry module, which performs conventional monocular SLAM over a small sliding window and a global pose graph module that incorporates metric depth measurements from the network as unary scale factors in a SIM(3) pose graph to constrain the solution scale.

Tateno et al. [63] proposes CNN-SLAM that uses predicted depth map from neural networks as initial guess to recover blurred depth borders which is then refined via a direct key-frame SLAM scheme based on small-baseline stereo matching (by enforcing color consistency minimization between a key-frame and associated input frames). The absolute scale information from predicted depth maps leads to more accurate pose trajectory and scene reconstruction. As in most cases, the camera used in SLAM differs from the one used in training the depth network, the authors also propose to normalize the depth maps so that the method is robust towards different intrinsic camera parameters. Additionally, they fuse CNN predicted semantic segmentation with the final global reconstruction model.

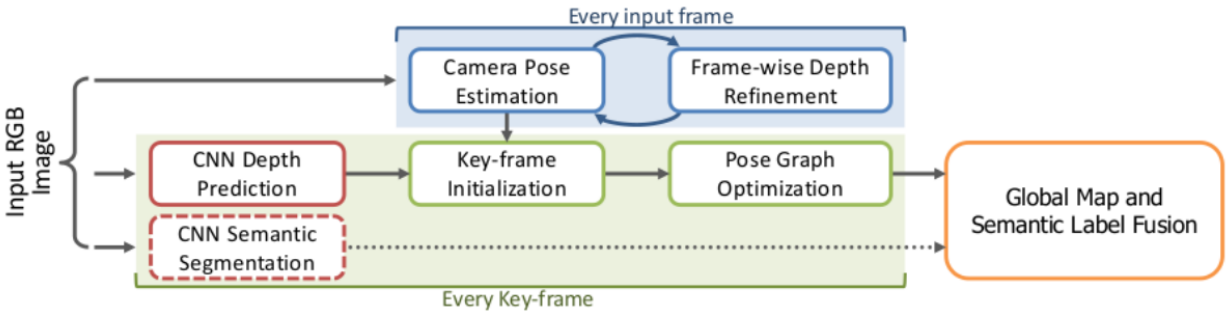

Figure 2.6: Overview of CNN-SLAM by [63].

Bian et al. [33] shows that integrating their scale-consistent depth predictions (SC-Depth) into ORB-SLAM2 [1] results in more robust and accurate tracking. They propose a hybrid pseudo-RGBD SLAM (as shown in Figure 2.7), where their scale-consistent predicted depths are used to initialize the 3D map at the beginning and in optimization by minimizing the difference between the predicted depth and the reprojected depth from the map back to the image plane. This extends the reprojection error from 2D into 3D, thereby improving the performance. The predicted pose from the CNN is used as the initial pose during tracking and is optimized through Bundle Adjustment (BA) [64].

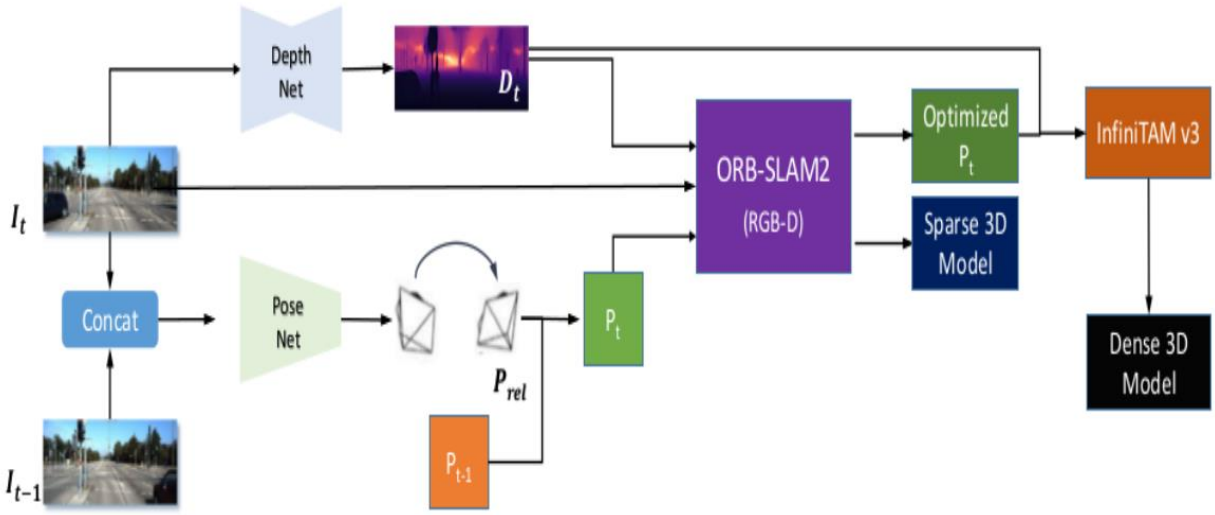

Figure 2.7: Pipeline of the Pseudo RGBD-SLAM proposed by [33].

3. Geometrically Consistent Depth Predictions

In this chapter, a self-supervised learning framework is proposed for the task of monocular depth and ego-motion estimation from unstructured sequences of monocular images in the hopes of achieving higher depth accuracy and consistency. Alongside the temporal constraint, a spatial geometric consistency loss is added that minimizes the difference between the predicted and synthesized depth and translation to enforce global scale consistency throughout an advancing video-based application such as vSLAM. In common with recent work [14]-[15], both the depth and pose networks are trained in an end-to-end manner with view synthesis as the key supervisory signal. The method uses single-view depth and multi-view pose networks with a loss based on warping nearby source images to the target using predicted depths and poses. Despite being jointly trained, the depth and pose networks can be tested independently during inference. The overview of the supervision pipeline is shown in Figure 3.1.

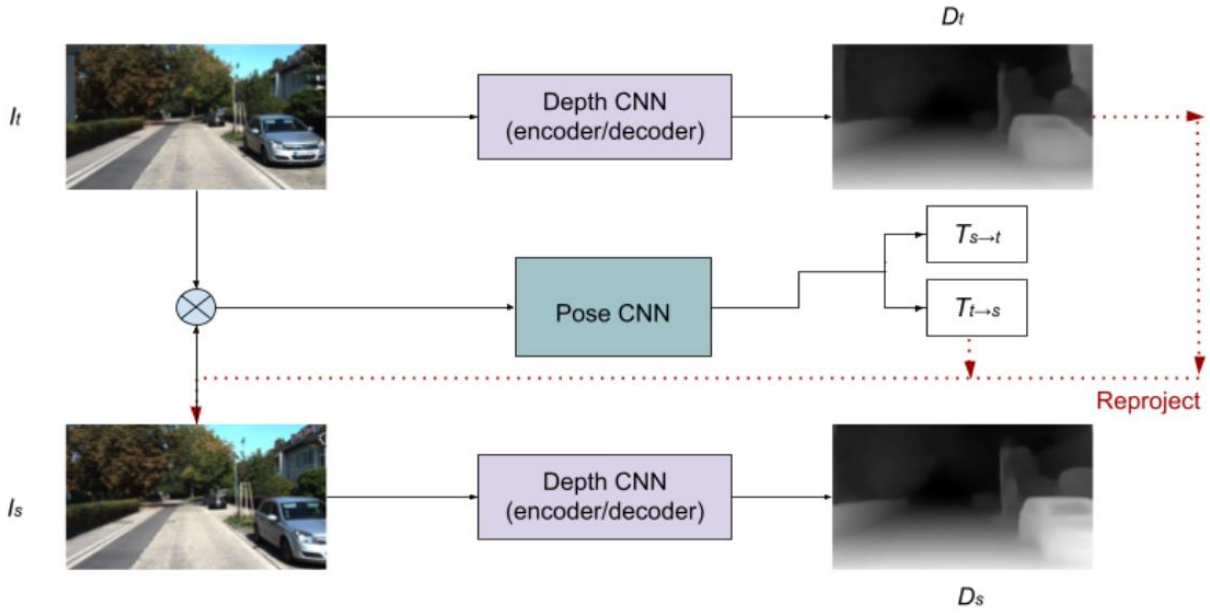

Figure 3.1: Overview of the supervision pipeline.

3.1 Modeling depth network to enforce scale consistency

Given the target image I_t at time t and its two adjacent source images, I_{t-1} and I_{t+1} at times $t-1$ and $t+1$ (denoted as I_s in Figure 3.1 for simplification) randomly sampled from a video during training, the depth CNN predicts the depth maps D_t and D_s for the target and each of the source images, respectively. The pose CNN takes in the sequence of images and outputs the relative poses $T_{t\to s}$ and $T_{s\to t}$ (3 euler angles in radians and 3 translation offsets along X, Y and Z components for each pose) that transforms I_t to the coordinate frame of I_s and vice versa. Alongside the temporal reconstruction loss, a spatial geometric consistency loss is proposed to produce more scale consistent depth maps.

3.1.1 Photometric reconstruction loss as key supervisory signal

The goal is to synthesize the target image I_t from each of its adjacent views and minimize the photometric difference between the reconstructed and the original target image. The L1 view synthesis loss can be formulated as:

$$\mathcal{L}_{vs} = \sum_{\rho} \left| I_t(\rho) - \widehat{I_t}(\rho) \right| \tag{3.1}$$

where ρ indexes over the pixel coordinates and I_t is the reconstructed target view obtained by warping the source view I_s. The 2D homogeneous coordinates of a pixel $\rho_t = [u_t, v_t, 1]$ in the target image can be projected onto the source view pixel $\rho_s = [u_s, v_s, 1]$ via a 3D rigid transformation:

$$\rho_s = K T_{t\to s} D_t(\rho_t) K^{-1} \rho_t \tag{3.2}$$

where, K denotes the rescaled parameters of the original intrinsic matrix. The projection results in continuous values for ρ_s. To sample from $I_s(\rho_s)$ in order to fill the value of $I_t(\rho_t)$, the differentiable bilinear sampling mechanism proposed in the spatial transformer networks (STN)

[65] is used. It linearly interpolates the values of the 4 neighboring pixels (top-left, top-right, bottom-left, bottom-right) of ρ_s by taking the weighted sum of the pixel intensities at these four points to approximate $I_s(\rho_s)$ as shown in Figure 3.2.

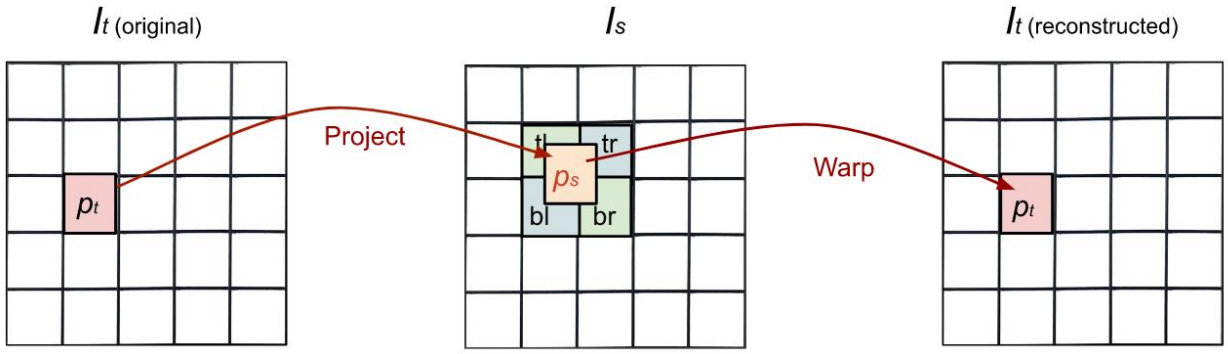

Figure 3.2: The differentiable warping process.

A weighted combination of Eq. 3.1 and a single scale structural dissimilarity index (1-SSIM) [66] is used as the final reconstruction loss:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_s \sum_\rho \alpha \frac{1-SSIM(I_t(\rho),\widehat{I_t}(\rho))}{2} + (1-\alpha)\left|I_t(\rho) - \widehat{I_t}(\rho)\right|, \tag{3.3}$$

where *N* is the total number of pixels. SSIM normalizes pixel illumination and is therefore applied to handle complex illumination changes, which is vital in estimating photometric losses because the formulation otherwise assumes that the surface is Lambertian. The SSIM between image x and image y is calculated using the formula below:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}, \tag{3.4}$$

where μ and σ are the mean and variance of the local image color, respectively which are obtained with a 3x3 block filter (using average pooling). C1, C2 and α are constants and we set their values as C1=0.0001, C2=0.0003 and α=0.85, following [12].

3.1.2 Proposed geometric consistency loss

To ensure geometric coherence, a constraint is enforced such that D_t and D_s, related by the geometric transformation $T_{t\to s}$, conform to the same 3D scene. A differentiable depth inconsistency operation is proposed to compute the pixel-wise inconsistency between the predicted depth map D_s and the synthesized depth $\widehat{D}_s$ of the source image. $\widehat{D}_s(\rho_s)$ corresponds to the Z coordinate of the point $\mathrm{P}_s = [X, Y, Z]$ in the camera coordinate frame, such that:

$$\mathrm{P}_s = T_{t\to s} D_t(\rho_t) K^{-1} \rho_t, \tag{3.5}$$

However, computing the difference directly between D_s and $\widehat{D}_s$ is incorrect as the synthesized map does not essentially lie on the grid of the predicted map. Therefore, D_s is aligned with $\widehat{D}_s$ using bilinear interpolation to form Ds'. The geometric consistency loss is then formulated as the average of the difference in depth between the interpolated map Ds', and $\widehat{D}_s$:

$$\mathcal{L}_{depth} = \frac{1}{N} \sum_s \sum_\rho \left| \widehat{D}_s(\rho) - D'_s(\rho) \right| \tag{3.6}$$

Alongside, an inverse relative transformation $T_{s\to t}$ is predicted using the pose network that transforms each of the source images to the target image. The inverse transformation of $T_{t\to s,}$ denoted as $\widehat{T}_{s\to t}$ is also computed. The difference between the predicted relative translation, t (from each source, s to the target along X, Y and Z axes) and the computed translation $\hat{t}$ is then penalized to further constrain the loss term and generate more accurate and consistent predictions:

$$\mathcal{L}_{pose} = \sum_s (|t_x - \hat{t}_x| + \left|t_y - \hat{t}_y\right| + |t_z - \hat{t}_z|) \tag{3.7}$$

In their study, [67] concluded that rotation behaves as noise, whereas translation contributes to salient supervisory signals during training and thus, we only considered the latter in the loss term.

By imposing the above loss terms across different samples in a batch, more scale consistent maps are produced over the entire video sequence, such that, depth of image I_1 concurs in scale with the depth of I_2 in a batch, and similarly the depth map of I_2 is coherent with the depth of I_3 in a different batch. Subsequently, all the depth maps produced for the entire sequence are consistent with each other.

3.1.3 Smoothness loss

Following the work of [14], multi-scales prediction along with a smoothness loss is used to allow gradients to propagate from distinguished regions to low textured regions. The gradients are derived from the pixel intensity loss between $I_t(\rho_t)$ and the interpolated pixels of $I_s(\rho_s)$, which may hinder training if the correct ground-truth pixel (ρ_s) lies in a region with low texture or far from the current estimation. In simpler words, the photometric loss is not informative in low texture scenes. Zhou et al. [14] found two strategies to be effective in solving this issue: 1) employing a convolutional encoder-decoder architecture with a small bottleneck for the depth predictor to enforce the output to be globally smooth and allow gradients to flow from informative regions to surrounding regions; 2) integrating smoothness loss and multi-scale prediction. Both strategies are taken into consideration in overcoming the gradient locality. The norm of the first order gradients d_x and d_y (along both x and y direction of the disparity image, respectively) is penalized to encourage smoothly changing depth values. To avoid depth discontinuities that frequently occur at image gradients, the cost is weighted with an edge-aware smoothness term using the image gradients δ_{I_x} and δ_{I_y}:

$$\mathcal{L}_{smooth} = \frac{1}{N}\sum_{\rho}|\delta_{dx}|\, e^{-|\delta_{I_x}|} + \left|\delta_{dy}\right|e^{-\left|\delta_{I_y}\right|}, \tag{3.8}$$

3.1.4 Masking Scheme

When applied to monocular videos, the view synthesis formulation in section 3.1.1 assumes that 1) the scene does not contain dynamic objects; 2) there is no occlusion between the target and the source images. Therefore, the presence of moving objects and occlusions violate the geometry consistency assumption. Considering the factors, the regions are masked out via an explainability mask *E* [14]. Like in [14], the network is jointly trained with the depth and pose networks to output a per-pixel soft mask *E* for each source-target pair that predicts the target pixels for which view synthesis will be successful (e.g., masks out pixels that are occluded and/or are non-rigid). The reconstruction and geometric consistency losses in Eqs. 3.3 and 3.6 respectively are then weighted by the predicted mask, giving:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_s \sum_{\rho \epsilon N} \mathrm{E}(\rho)\left(\alpha \frac{1-SSIM(I_t(\rho),\widehat{I_t}(\rho))}{2} + (1-\alpha)\left|I_t(\rho) - \widehat{I_t}(\rho)\right|\right), \tag{3.9}$$

$$\mathcal{L}_{depth} = \frac{1}{N}\sum_s \sum_{\rho} E_\rho \left|\widehat{D}_s(\rho) - D'_s(\rho)\right| \tag{3.10}$$

It is important to note that *E* has no direct supervision and predicting its value as zero would perfectly minimize the above loss terms. Therefore, a regularization term $\mathcal{L}_{\text{reg}}$ is added to constrict zero predictions by reducing the cross-entropy loss with constant labels [0,1] at each pixel location. The network thus still learns to minimize the loss terms weighted by the mask but is allowed a certain amount of slack for discounting the factors not considered by the model. The loss term $\mathcal{L}_{\text{c}}$ at each output scale *c*, forming the final total loss as $\mathcal{L}_{total} = \sum_{c=0}^{3} \mathcal{L}_c$ is defined as:

$$\mathcal{L}_c = \alpha_r \mathcal{L}_{rec} + \alpha_s \mathcal{L}_{smooth} + \alpha_d \mathcal{L}_{depth} + \alpha_p \mathcal{L}_{pose} + \alpha_e \mathcal{L}_{reg}(E) \tag{3.11}$$

3.2 Implementation Details

3.2.1 Training Details: Hyperparameters and System Details

All the deep-learning models have been implemented using the publicly available Tensorflow [68] framework. For the learning-based algorithm we chose SfM-Learner [14] as our baseline. The original implementation is available on their Github page. Batch normalization was applied to all the layers, except the output layer and Adam Optimizer was used with a learning rate of 0.0001 and its momentum terms β_1 and β_2 were set to 0.9 and 0.95 respectively. The parameters α_r, α_e and α_s in Eq 3.11 were set to 1, 0.2 and 0.1 respectively. For choosing weights for the geometric losses ($\mathcal{L}_{depth}$ and $\mathcal{L}_{pose}$), hyperparameter tuning was done and a value of 0.1 and 1.0 yielded the best result for α_d and α_p respectively. A batch size of 4 was used and all the experiments were conducted with image sequences captured from a single RGB camera. The images were resized to 128 x 416 (height x width) during training like the baseline for a fair evaluation. The training typically converges after 120K iterations. The experiments were conducted on an Intel Core i7-7700 CPU (64-bit with a base frequency of 3.60 GHz) and a pre-installed Nvidia GeForce GTX 1050 GPU that has a memory of 4 GB.

3.2.2 Training Datasets and Data Preprocessing

I. KITTI: Eigen Split

In its raw form, the KITTI dataset [18] consists of RGB images (rectified stereo pairs) from 61 outdoor scenes along with their corresponding sparse point clouds captured while driving using car-mounted cameras and a rotating Velodyne LiDAR scanner respectively. Originally, a typical image is about 375 x 1242 pixels in dimension and the maximum depth present in the dataset is 80 meters. The Eigen split is more frequently used by the research community as the ground truth

values are not part of the official KITTI training data. Instead, the ground truth is generated by reprojecting the 3D points as observed from the LiDAR onto the left input RGB image of the stereo pair. The split uses all 61 scenes from the 'residential', 'city', and 'road' categories of the KITTI dataset, which are split into 28 for testing and the rest for training and validation. Besides, the KITTI dataset provides ground truth poses for 11 visual odometry sequences and thus is also a common benchmark to evaluate VO-based algorithms.

The training has been done using all 33 scenes in the training set. Following the baseline, we discarded images from the testing set as well as removed static sequences with a mean optical flow of less than 1 pixel for training. Images from both cameras of the stereo setup have been used but were considered independently in the formation of the training sequences. This resulted in a total of 44,222 sequences, out of which we used 39,835 for training and the rest for validation. The length of the image sequence is 3, where the central image is the target and ±1 frame is the source (as shown in Figure 3.3).

Figure 3.3: Example of training data with 3 consecutive images in a sequence.

II. NYUv2 DEPTH

The NYUv2 dataset [69] consists of densely labeled monocular RGB images, each being 640 x 480 in resolution taken in indoor environments with 464 different scenes. Unlike the KITTI dataset, which collects ground truth depth data with a laser scanner, this dataset takes ground truth

of depth using an RGB-D camera (Microsoft Kinect). The scenes are split and distributed as 249 for training and 215 for testing. It is not a one-to-one correspondence between depth maps and RGB images due to different frame rates between the two sensors. In order to align the two, each depth map is associated with the closest RGB image at first. Then, with the geometrical relationship provided by the dataset, the camera projections are used to align depth and RGB pairs. It is the common benchmark and the primary training dataset in supervised monocular depth estimation methods. However, the loss term fails to converge when all the images in the training set are used for training in an unsupervised manner. Bian et al. [67] demonstrate that complex ego-motion exhibited in handheld settings impede learning. Pure rotational cases (which is generally the dominant camera motion in videos captured using handheld devices) in image warping do not rely on depth. Moreover, incorrect estimation of pure rotation contributes to noisy gradients during training. Thus, they rectified the training images by removing the pure rotational cases as a preprocessing step. We downloaded the rectified dataset directly from their Github page. The images are resized to 416 x 312 before training.

III. Data Augmentation

Random scaling was applied to the training images and their respective intrinsic parameters by sampling from uniform distributions in the range [1,1.5], followed by a crop also using uniformly distributed random numbers. Such augmentation generates new and different samples to train the datasets, which facilitates performance and lowers the error metrics. Following [12], an additional color augmentation is done on the input RGB images with a 50% probability, where random gamma, brightness, and color shifts were applied in the ranges [0.8,1.2] for gamma, [0.5,2.0] for brightness, and [0.8,1.2] for each color channel separately.

3.2.3 Network Architecture

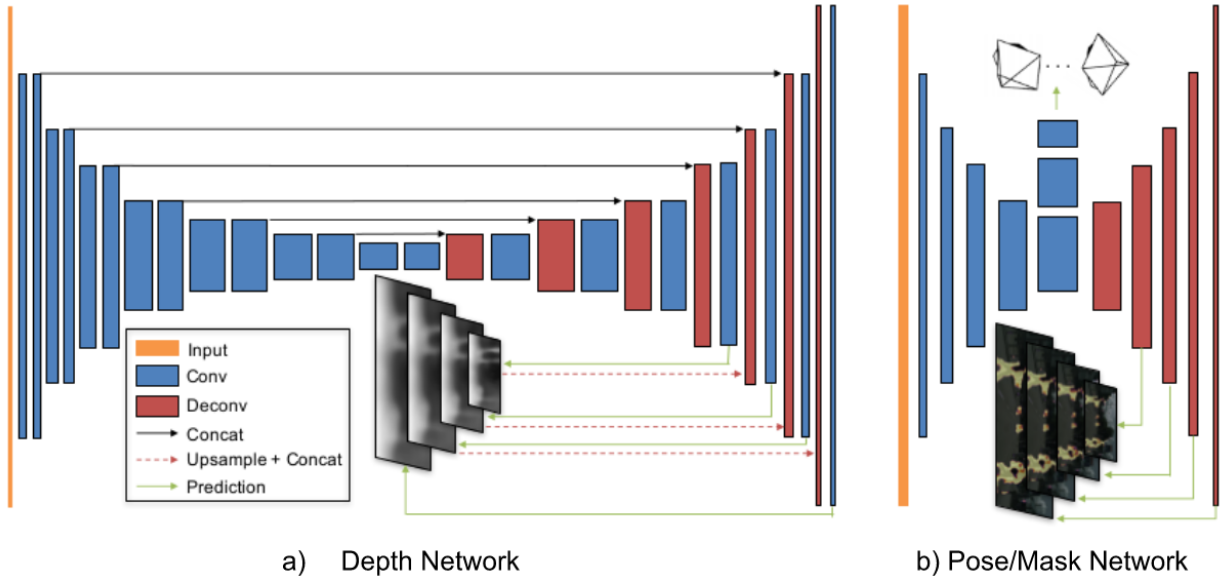

Figure 3.4: Network architecture for depth/pose/explainability mask modules [14].

I. Single-view depth network

The baseline [14] uses the DispNet architecture for single image depth estimation, which is based on an encoder-decoder design with skip connections and multi-scale depth predictions. The kernel size is 7,7,5,5 for the first 4 convolution layers respectively and 3 for all the other layers. The number of output channels for the first convolutional layer is 32. All the layers are followed by ReLU activation except for the output layers, where it uses $1/(\alpha * sigmoid(x) + \beta)$ so that the predicted depth is always positive (in front of the camera) and within a reasonable range. The scaling factor, α is set to 10 and the minimum value for the predicted depth is constrained to be 0.01.

II. Pose network

The network takes in the target image concatenated with one source image (along the color channels) at a time and outputs a 6-DoF relative pose (3 Euler angles and 3-D translation) between

the target and the source view. It consists of 7 stride-2-convolutions followed by a 1 x 1 convolution with 6 output channels. Global average pooling is applied to total predictions at all spatial locations. All convolutional layers are followed by ReLU activation, apart from the last layer, with no non-linear activation. The kernel size is 3 for all the layers except for the first 2 convolutional layers with 7 and 5 respectively. The number of output channels for the first convolutional layer is 16.

III. Explainability mask

The mask prediction network shares the first 5 encoding layers with the pose network (therefore the number of output channels for the first convolutional layer is also 16), followed by 5 deconvolution layers with multi-scale side predictions. All the layers are succeeded by ReLU except for the output layers, where no activation is applied. The number of output channels for each prediction layer is 2, with every 2 channels normalized by softmax to obtain the explainability prediction for the corresponding source-target pair (the second channel after normalization is E) that is used to weigh the reconstruction and geometric losses in Eqs. 3.9 and 3.10 respectively. The kernel size is 3 for all the layers except for the first 2 conv and last 2 deconv layers with 7,5,5 and 7 respectively.

3.3 Experimental Results

In this section we compare our results with prior state-of-the-art unsupervised SIDE methods. Our method will be referred to as '*proposed*', indicating the addition of the geometric consistency loss (as explained in 3.1.2) to the baseline [14].

3.3.1 Evaluation Criteria

To compare and evaluate the performance of different depth estimation networks, a few commonly accepted evaluation metrics are categorized into accuracies and errors, the errors being root mean squared error (RMSE), root mean squared logarithmic error (RMSE log), mean absolute relative error (Abs Rel), and squared relative error (Sq Rel). For error metrics, lower is better, and higher is better for accuracy.

- $RMSE = \sqrt{\frac{1}{N}\sum_{\rho \epsilon N}\left\|D_\rho - \widehat{D}_\rho\right\|^2}$, (3.12)
- $RMSE\ \log = \sqrt{\frac{1}{N}\sum_{\rho \epsilon N}\left\|\log\left(D_\rho\right) - \log\left(\widehat{D}_\rho\right)\right\|^2}$, (3.13)
- $Abs\ Rel = \frac{1}{N}\sum_{\rho \epsilon N}\frac{\left|D_\rho - \widehat{D}_\rho\right|}{D_\rho}$, (3.14)
- $Sq\ Rel = \frac{1}{N}\sum_{\rho \epsilon N}\frac{\left|D_\rho - \widehat{D}_\rho\right|^2}{D_\rho}$, (3.15)
- $Log10 = \frac{1}{N}\sum_{\rho \epsilon N}\left|\log_{10}\left(D_\rho\right) - \log_{10}\left(\widehat{D}_\rho\right)\right|$, (3.16)
- $Accuracies: percentage\ of\ d_\rho\ such\ that \max\left(\frac{D_\rho}{\widehat{D}_\rho}, \frac{\widehat{D}_\rho}{D_\rho}\right) = \delta < thr$

where $\widehat{D}_\rho$ is the predicted depth value of pixel ρ, and D_ρ is the ground truth of depth (attained from LiDAR). *N* denotes the total number of pixels for which real-depth values (ground truth) are available, and *thr* denotes the minimum threshold within which the maximum ratio between the predicted depth and ground truth depth lies. Therefore, it indicates the number of pixels for which depth has been inferred within an accurate range.

3.3.2 Evaluation of the depth network

I. Evaluation on the Eigen Test Split

The evaluation of our unsupervised SIDE method is done on the KITTI dataset using the 697 images from the Eigen test split and the depth value was capped at 50m as standard practice. For a fair evaluation of unsupervised methods that cannot retrieve the absolute scale, the predicted depth maps are multiplied by a scalar that matches the median with that of the ground truth as proposed in [14]. As shown in Table 3.1, our method performs on par with several unsupervised methods. Results of other methods were taken from their respective papers. Learning with semantic labels or optical flow has been shown to improve the performance of monocular methods. However, this thesis is more interested in exploring monocular methods that do not use additional information, and hence, does not include those findings in the study. Figure 3.5 provides examples of visual comparison amongst the proposed method, the semi-supervised method by [12] and the baseline [14]. It is evident from the figure that the results obtained are comparable to the state-of-the-art semi-supervised method, albeit trained in an unsupervised manner. Moreover, the proposed method outputs more distinct and sharper depth maps than the baseline even in cases where the latter fails such as, in vast open spaces (row 2 of Figure 3.5) and in presence of objects too close to the camera (row 3).

II. Analyzing the effects of added geometric consistency

To check for the geometric consistency of the depth maps, following [33] we used the Open3D library [70] that takes in two point clouds (of the source and target) as well as the relative transformation between them as inputs and computes two main metrics: 1) RMSE of the inlier correspondences of two aligned point clouds using a constant threshold of 0.02, and 2) Fitness which measures the overlapping area of two point clouds (number of inliers / number of points in

target point cloud). The depth and relative pose for every adjacent frame in KITTI sequence 09 are predicted where the depth maps from [14] and [33] are also resized to 416 x 128 before converting them to point clouds for fair evaluation. Table 3.2 shows the results.

Table 3.1: Comparison of the proposed depth prediction method with different state-the-art SIDE methods that use stereo images (stereo), monocular images (mono) or a combination of both (stereo + mono) for training.

Method	**Supervision**	**Error metric ↓**				**Accuracy metric ↑**		
		AbsRel	**Sq Rel**	**RMSE**	**RMSlog**	**$\delta<1.25$**	**$\delta<1.25^2$**	**$\delta<1.25^3$**
Monodepth [12]	Stereo	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Pilzer et al. [58]	Stereo	0.144	1.007	4.660	0.240	0.793	0.923	0.968
Zhou et al. [71]	Stereo + Mono	0.195	1.754	6.505	0.271	0.717	0.898	0.961
Chen et al. [47]	Stereo + Mono	0.141	1.227	5.629	0.239	0.809	0.927	0.962
SfM-Learner [14]	Mono	0.208	1.768	6.586	0.283	0.678	0.885	0.957
Kumar et al. [60]	Mono	0.211	1.980	6.154	0.264	0.732	0.898	0.959
Bian et al. [15]	Mono	0.149	1.137	5.771	0.230	0.799	0.932	0.973
Bian et al. [33]	Mono	0.132	0.982	5.226	0.209	0.835	0.947	0.978
proposed	Mono	0.153	1.143	5.728	0.231	0.796	0.927	0.975

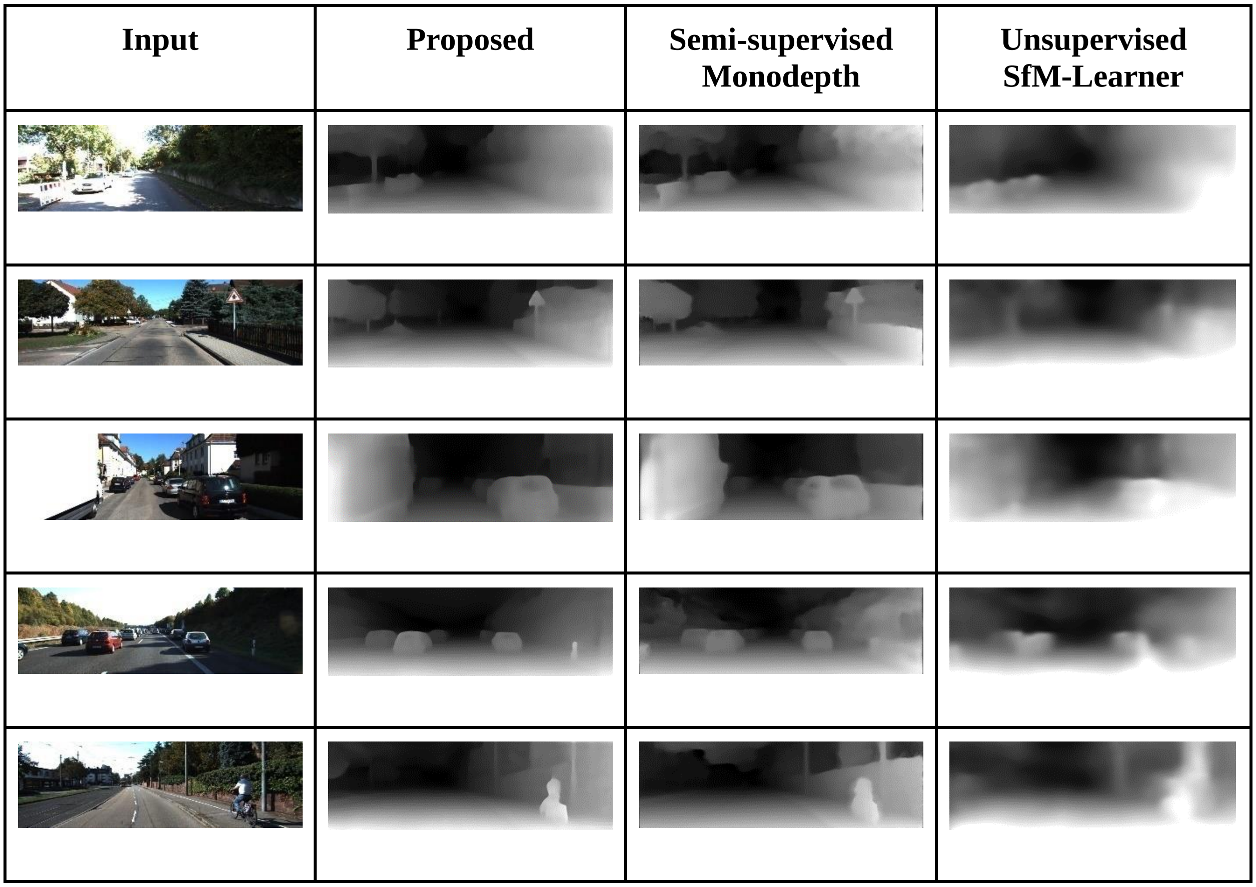

Figure 3.5: Comparison of depth estimation between Monodepth [12] (with ground-truth pose supervision), SfM-Learner [14] (unsupervised) and proposed (unsupervised).

Besides, Figure 3.6 shows that the proposed model trained with the geometric constraint results in more consistent depth predictions compared to the one with a lack thereof [14]. The latter showcases larger differences in the number of pixels that correspond to a certain depth range when compared with the ground truth, especially in the mid-range (Figure 3.6a). The network tends to overestimate or underestimate the depth values to a slightly greater extent when compared to the proposed method (Figure 3.6b). Such inconsistency does not impact single image-based tasks but is crucial in estimating the camera pose (which will be discussed in Chapter 4). Enforcing the proposed geometric constraint has proven to reduce the differences, therefore, results in more globally consistent depth maps. Moreover, as can be seen from Table 3.1, the percentage of pixels

lying within an accurate range increased from 67.8% to 79.6% when the threshold is set to 1.25. This further supports that the network benefits from the geometric constraint.

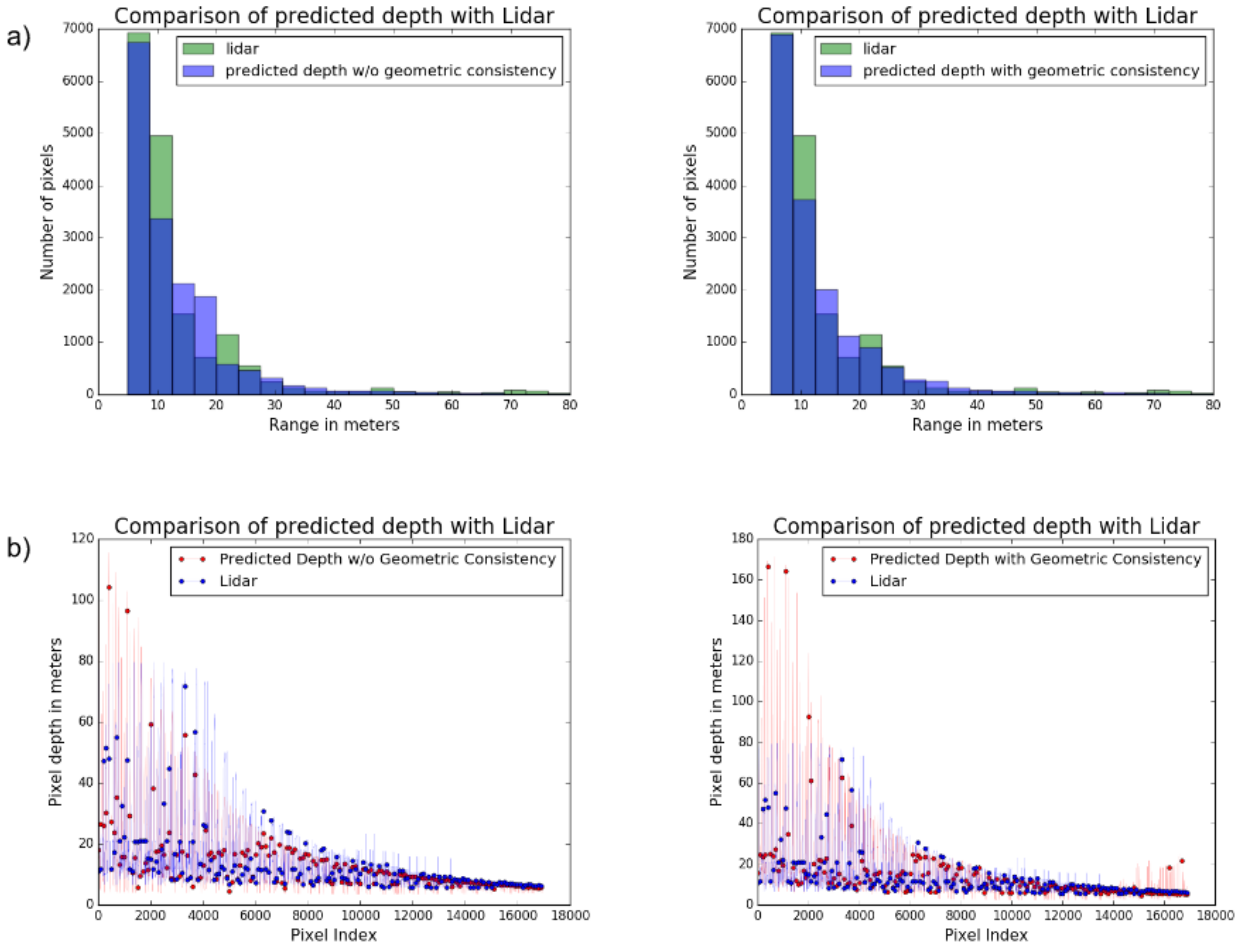

Figure 3.6: Plots illustrating the effect of the geometric constraint.

Table 3.2: Depth consistency evaluation on KITTI sequence 09.

Methods	**Fitness ↑**	**RMSE ↓**	**No. of Correspondences ↑**
Zhou et al. [14]	0.354	0.00988	75.776K
Bian et al. [33]	0.689	0.00871	134.956K
proposed	0.541	0.00890	115.659K

III. Evaluation on the NYU dataset

The evaluation is done on the commonly used test subset of 654 images officially provided in NYUv2. Removal of relative rotation [68] between target-source pairs during training significantly improves as rotation is the dominant ego-motion in videos from hand-held devices. This dataset has mostly been used by methods that used ground truth depth for supervision. The lack of stereo images in the dataset limits semi-supervised methods from using it for training. However, they can use it to check the generalization ability of their models.

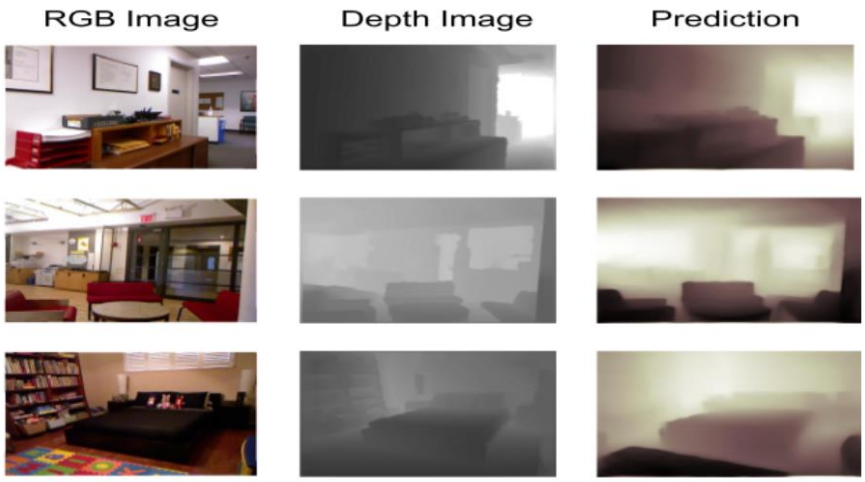

Figure 3.7: Comparison of depth map between proposed and the ground truth using NYU dataset.

Table 3.3: SIDE results on NYUv2.

Method	Supervision	Error metric ↓			Accuracy metric ↑		
		AbsRel	**RMSE**	**log10**	**$\delta<1.25$**	**$\delta<1.25^2$**	**$\delta<1.25^3$**
Laina et al. [11]	Depth	0.127	0.573	0.055	0.811	0.953	0.988
Yin et al. [44]	Depth	0.108	0.416	0.048	0.875	0.976	0.994
SfM-Learner [14]	Mono	0.193	0.682	0.088	0.754	0.932	0.956
Bian et al, 2021 [33]	Mono	0.157	0.593	0.067	0.780	0.940	0.984
Proposed	Mono	0.173	0.639	0.075	0.734	0.937	0.983

3.3.3 Generalizing to other datasets

Here the generalization ability of the models outside their training domains is evaluated. The model trained on KITTI is applied to the Make3D dataset [72] that was unseen by the network during training. The Make3D dataset contains 134 test samples of color images of size 2272 x 1704 with corresponding depth information from a laser scanner, provided at a low resolution of 55 x 303 in an outdoor setting, thus making it suitable for evaluation. However, as they do not contain stereo images, semi-supervised methods that depend on stereo images for training can only use this dataset to test the generalization ability of their networks. Due to the different aspect ratio of the dataset, evaluation is done on a central crop of the images as standard practice. The images are resized to a resolution of 512 x 384, and then cropped at the center before testing. As the network is fully convolutional, it can be applied to arbitrary resolutions. However, due to the specific structure of the KITTI scenes, the network learns a strong prior of the scene scale that does not work well for inputs of very different resolutions.

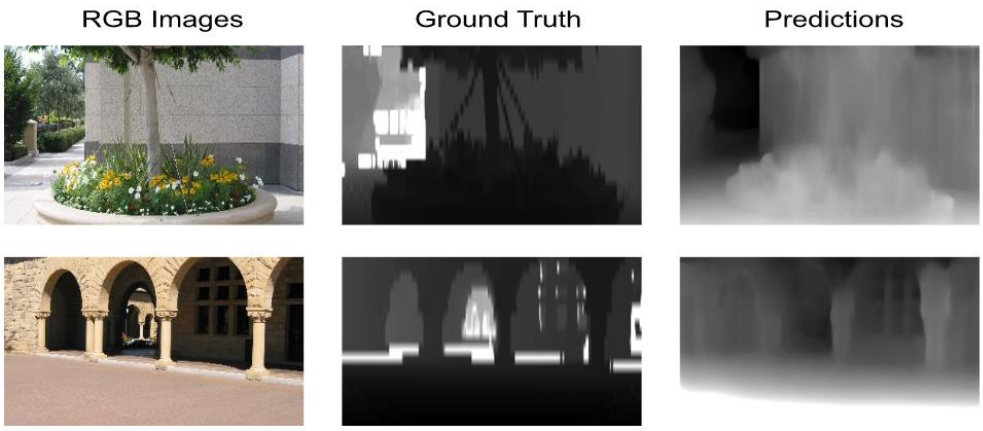

Figure 3.8: Sample predictions on datasets outside the training domain.

Following the standard evaluation protocol of [10], the errors are only computed where the depth is less than 70 meters. It can be seen from Table 3.4 that the proposed method does not contribute to a significant improvement when compared to the baseline [14] with a 70m cap. However, we notice a considerable decrease of 28.16% in the RMSE when the depth values are limited to 50 meters. This shows that most errors are accounted for from depths that have a higher value, meaning the model can generalize well to scenes closer to the camera. Besides, Figure 3.8 shows that our predictions can capture the global scene layout reasonably well on the dataset.

Table 3.4: Results on Make3D dataset.

Method	**Abs Rel**	**Sq Rel**	**RMSE**	**log10**
Laina et al. [1]	0.198	1.665	5.461	0.082
Godard et al. [12]	0.443	7.112	8.860	0.143
Zhou et al. [14]	0.383	5.321	10.47	0.478
Proposed (70m cap)	0.385	4.744	10.37	0.203
Proposed (50m cap)	0.368	3.792	7.449	0.183

3.3.4 Comparison with other depth sensors

I. Range and Accuracy

Infrared depth cameras such as Intel RealSense and Microsoft Kinect usually operate indoors and allow short distance measurements, namely up to 9 meters with an error of 0.5% of the measured range [73]. However, the practical ranging limit is between 1 and 4 meters, where the sensors can reliably measure depth. Stereo cameras, on the other hand, can sense depths at as far as 500 meters, depending on the camera resolution and the length of the baseline. Commercially available stereo cameras can accurately calculate distances up to 120 meters with an accuracy of 1.4 to 2.8 mm for every 1 meter measured. LiDARs are generally used outdoors, and their range can be up to a few kilometers. Meanwhile, neural networks can estimate depth up to 50 meters with an accuracy of 1% of the measured depth.

II. Coverage

The stereo system can cover up to 87% of the image area, meaning it can provide useful depth information for about 87% of the pixels leaving out a part of the image at the edge that cannot be seen from its corresponding stereo pair, as well as a few occasional holes in the disparity map, where the disparity is nearly 0 (corresponding to points that have not moved in the images). In contrast, the velodyne LiDARs only covers up to 4% of the image area as shown in Figure 3.9a (pixels with no depth values are assigned a value >3500m), thereby supporting that the depth information from such scanners is indeed quite sparse compared to that from an RGB-D camera or a neural network, for which a per-pixel depth map can be obtained.

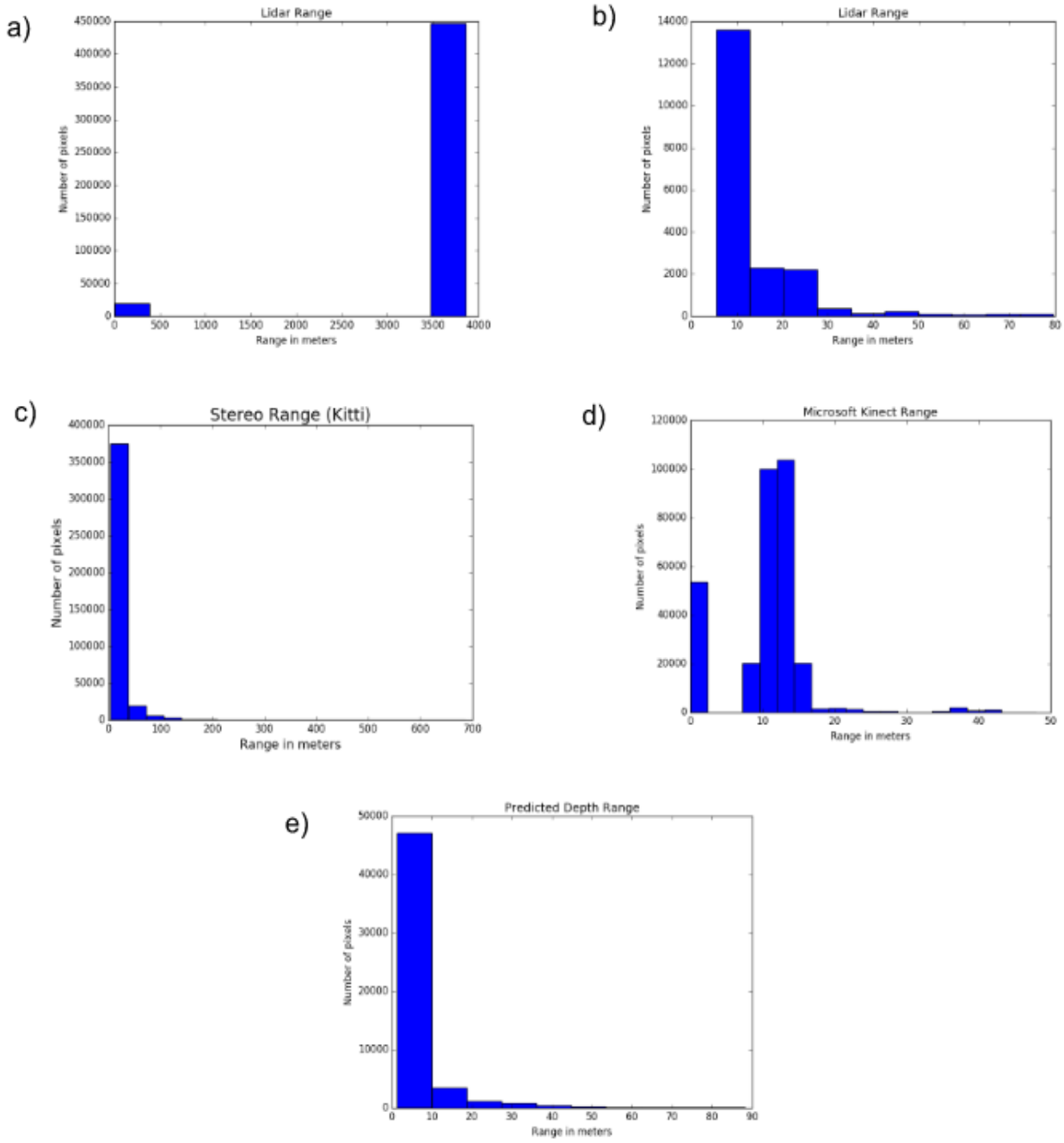

Figure 3.9: Bar graph illustrating the maximum range measurement using stereo cameras, RGB-D camera, LiDAR and proposed CNN.

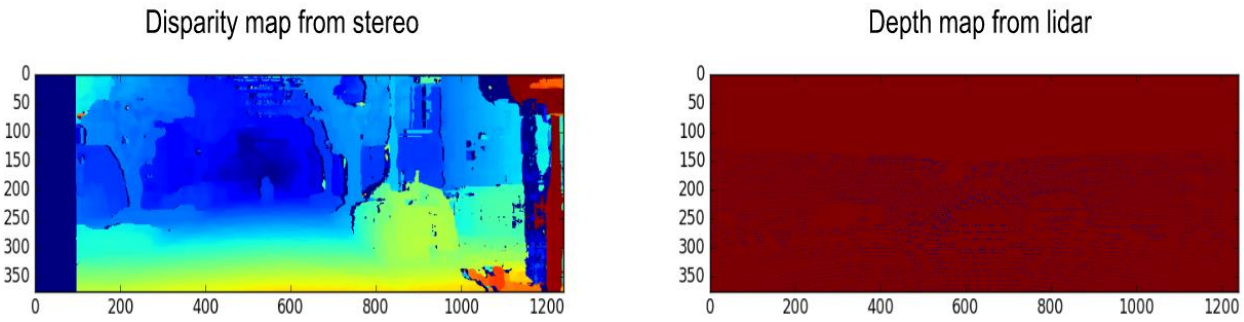

Figure 3.10:Depth maps from stereo camera and velodyne LiDAR.

3.3.5 Evaluation of the pose network

The pose network predicts a relative 6DOF ego-motion for each short sequence of images- 3 euler angles in radians and 3 translation offsets along the X, Y, Z directions with respect to the central image (target). The length of input image sequences is set to 3 at times *t-1*, *t* and *t+1*, where the pose at *t-1* is adjusted as the reference coordinate frame. For the evaluation of the network, the model is tested on sequences 09-10 of the official KITTI odometry split with ground-truth odometry obtained via IMU/GPS readings (more on the dataset in Chapter 4). For evaluation, the ground truth trajectory is also broken down into 3-frame snippets, where the first frame in the sequence is fixed as the reference. Besides, the ego-motion is compared to those from monocular ORB SLAM [17] (short), which is also run on 3-frame snippets as the input setting. To resolve the scale ambiguity during evaluation, the scale factor for the predictions is optimized to align with the ground truth using Eq 3.17. The Absolute Trajectory Error (ATE) is computed on each short sequence and averaged over the full sequence using Eq 3.19 like in [14]. The results are shown in Table 3.5. Note that the root mean squared error (RMSE) is calculated without considering the first pose in each short sequence as it is always set as the identity for both the ground truth and predictions and thus corresponds to zero error.

$$scale = \frac{\sum(gt_{xyz} * pred_{xyz})}{\sum {pred_{xyz}}^2}, \tag{3.17}$$

$$RMSE = \sqrt{\frac{1}{2}\sum_{n=1}^{2}(scale_n * pred_n - gt_n)^2}, \tag{3.18}$$

$$ATE = \frac{1}{N}\sum_{n=1}^{N} RMSE_n, \tag{3.19}$$

Table 3.5: Absolute trajectory error with standard deviation on the KITTI odometry split averaged over all 3-frame snippets.

Methods	Seq 09	Seq 10
Baseline	0.032 ± 0.017	0.030 ± 0.015
Proposed	0.020 ± 0.006	0.024 ± 0.008
ORB SLAM 2 (short)	0.064 ± 0.141	0.064 ± 0.130

3.4 Ablation Study

To test the efficacy of different components of the network, a few ablation experiments are conducted using the KITTI dataset. The ablation study results in Table 3.6 show that training with the proposed geometric constraint results in a significant improvement in the performance. Besides, the importance of masking inexplicable regions and using multi-scale depth predictions for training is also discussed later in the section. Figure 3.11 compares the trend in the loss terms during training between models 2 and 4 from Table 3.6. As discussed earlier, the geometric constraint causes the smoothness loss to increase, which results in sharper depth maps (Figure 3.5). Reinforcement of the geometric loss terms also leads to an overall decrease in the reconstruction loss and the total loss. However, the loss for the explainability mask is seen to slightly increase.

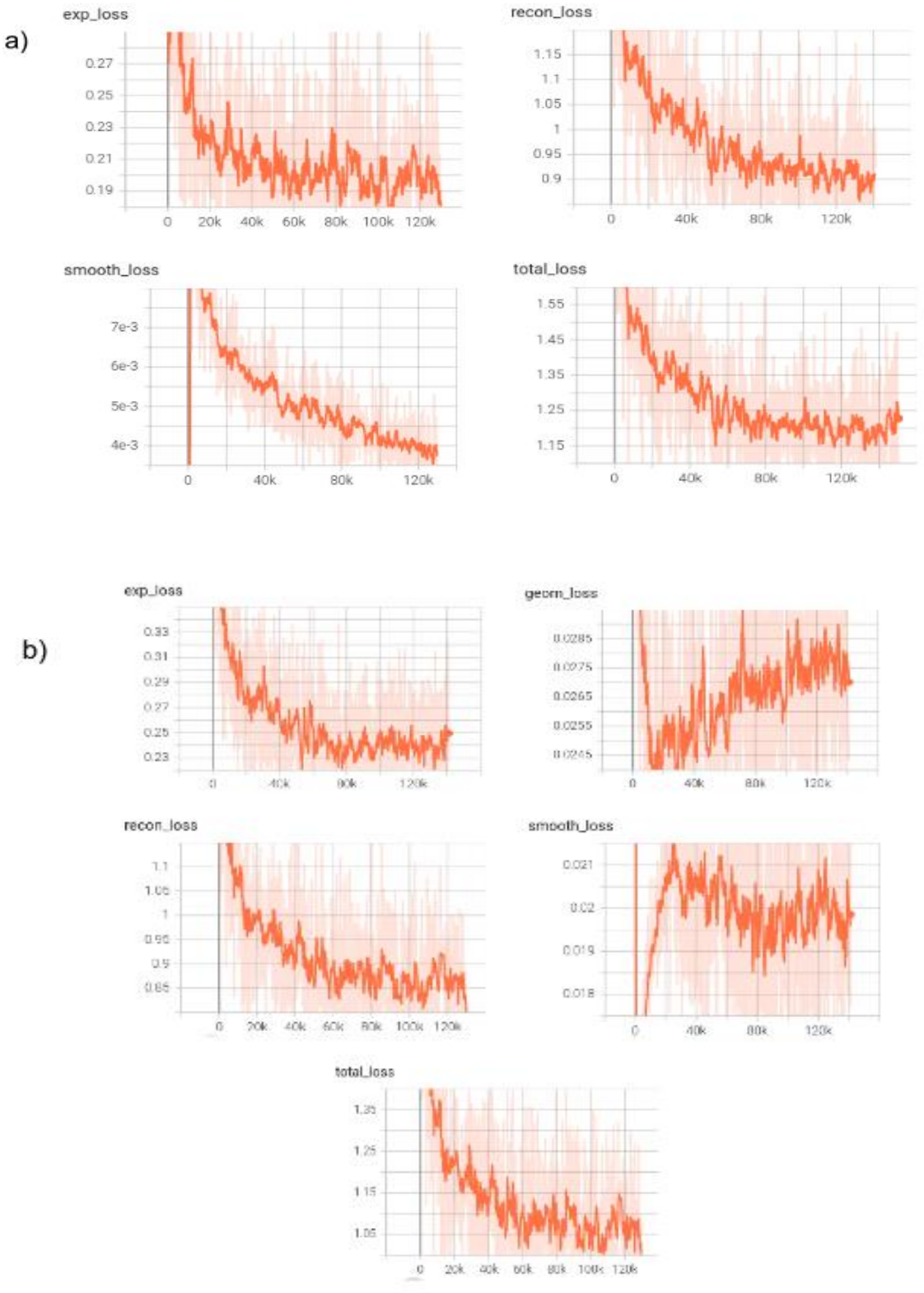

Figure 3.11: Trend in loss terms during the training of a) model 2 b) model 4 over 120k iterations.

Table 3.6: Ablation study results on KITTI.

No.	Models	AbsRel	δ<1.25
1.	Baseline (B) [14]	0.208	0.695
2.	B + SSIM + Edge Aware Smoothness	0.195	0.741
3.	B+ SSIM + Smoothness + $\mathcal{L}_{depth}$	0.159	0.773
4.	B+ SSIM + Smoothness + $\mathcal{L}_{depth} + \mathcal{L}_{pose}$	0.153	0.796
5.	4. + Single scale predictions	0.161	0.768

3.4.1 Effect of Structural Similarity and Edge-Aware Smoothness

The baseline minimizes an L1 norm of the second-order gradients for the predicted depth maps. In the proposed framework, this is replaced with an L1 norm of the first-order gradients (Eq 3.8) along the spatial directions weighted by an edge-aware term using the image gradients as in [3]. This encourages smoothness to be guided by image edges by filtering out noise or texture while retaining sharp edges. Row 2 of Figure 3.12 shows the effect of adding SSIM loss and an edge-aware smoothness loss to the baseline which shows to preserve edges better than when the network is trained without them (row 3). Besides, the loss terms result in a 6.25% decrease in the absolute relative error as shown in Table 3.6.

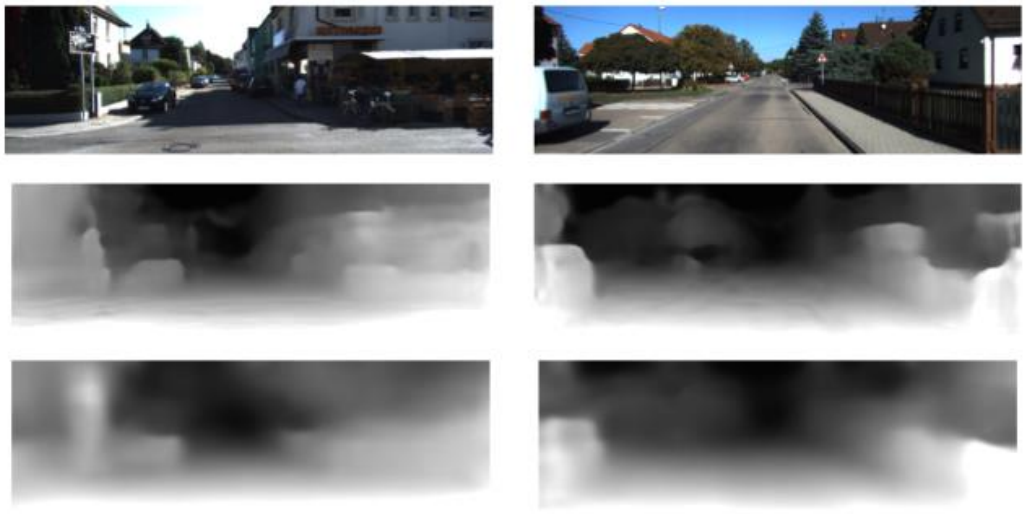

Figure 3.12: Effect of edge-aware smoothness and SSIM.

3.4.2 Effect of Explainability Mask

Samples of the explainability mask predicted by the network is illustrated in Figure 3.13. As can be seen, the network has learned to identify occlusions that disappear from the frame in subsequent views (Figure 3.13a) as well as dynamic objects (such as the car in Figure 3.13b) as inexplicable pixels that could have affected the reconstruction loss. Both the geometric loss (depth component) and the reconstruction loss are weighted by the mask during training.

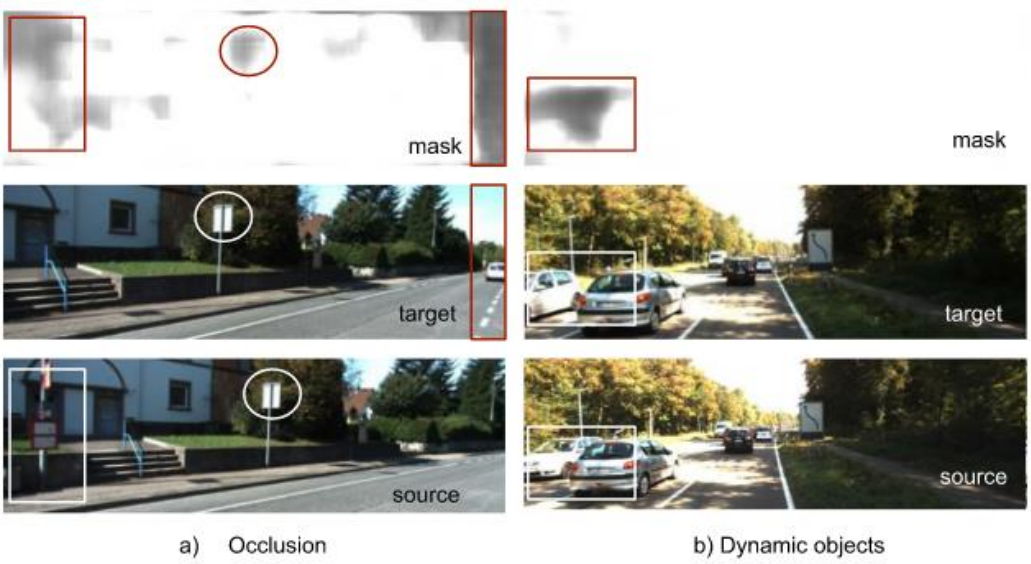

Figure 3.13: Ability of the mask in recognizing occlusions and dynamic objects.

3.4.3 Effect of multiscale depth predictions

The baseline computes losses on image pyramid, i.e., it outputs the predictions at four different scales that halve in spatial resolution at each of the subsequent scales starting from the original resolution. The same object appears different at different distances provided that the camera position does not change. As the training dataset does not contain images of the same scene from different distances, a similar situation can be simulated by changing the resolution of the images for the SIDE task. By working in multiple scales, the network looks at different cues at each scale and therefore becomes more robust in perceiving the depth of the scene. Table 3.6 shows that computing the loss using single scale supervision, i.e., at the original resolution only degrades the performance of the network. However, the multi-scale predictions introduce computational overhead during training. Besides, Figure 3.14 shows that the network begins to understand the depth cues at a lower resolution first during the initial epochs of training.

Figure 3.14: Depth prediction for the same image at different resolutions in a multi-scale framework during the first epoch of training.

3.5 Discussion

In this chapter, a novel geometric consistency loss term is incorporated in the framework of [14] to produce more scale consistent depth predictions, where the difference in the predicted and computed depth and transformation is minimized. Additionally, a structural similarity index is used in the reconstruction loss term to handle illumination changes. The proposed framework results in a 13.03% decrease in the RMSE obtained from the depth predictions compared to the baseline and 11.8% more pixels have manifested to be accurate within a threshold of 1.25. Besides, the proposed method performs comparatively with other state-of-the-art SIDE methods. The neural network has been trained and evaluated on both indoor (NYUv2) and outdoor datasets (KITTI). Range and accuracy of the predictions are further compared to other sensors such as stereo cameras, Microsoft Kinect, and LiDAR.

4. RGB-PD SLAM and Augmented Reality

In this chapter the scale consistent depth predictions introduced in Chapter 3 are integrated into a well-established visual monocular Simultaneous Localization and Mapping (SLAM) system, namely the ORB SLAM [17] using the KITTI dataset. The proposed approach estimates the camera poses from a sequence of monocular RGB images, where the 3D scene is reconstructed using depth predictions from the CNN model. Therefore, the SLAM approach is termed as RGB-PD SLAM, where *PD* stands for predicted depth. SLAM is often formulated as a maximum-likelihood inference problem on a factor graph, where variable nodes in the graph that represent the camera poses and the landmark locations are related through factor nodes and edges that encode the geometric relationship between them. The residual error models are assumed to be simple unimodal Gaussian during the optimization step. However, as the neural network is trained using unstructured monocular videos in an unsupervised manner, the predictions still suffer from per-frame ambiguity. Therefore, a novel sparse local bundle adjustment (SBA) process using probabilistic approach is introduced in the framework to handle the noises in the predictions, where the map points and keyframe poses are locally optimized while simultaneously determining which depth estimations are correct within a single integrated Bayesian framework. Later, we demonstrate how the improved quality of the reconstruction and pose estimation from consistent depth predictions facilitate seamless Augmented Reality (AR) applications.

4.1 SLAM: Pipeline Overview

The system follows a keyframe-based paradigm and the parallel tracking and mapping strategy in [74] to achieve real time performance. A schematic illustration of the pipeline is shown in Figure 4.1. The tracking thread (frontend) estimates the camera pose at frame rate while the mapping thread (backend) updates the reconstructed map with every keyframe at a lower

frequency and minimizes the geometric residuals in a graph optimization step, where all the map points and keyframe poses in the global map are refined using bundle adjustment (BA) [64]. To deal with large drifts in trajectories, the framework also runs loop closure detection in a separate thread, where potential candidates are identified using visual appearances between features. Each component of the system is discussed further in the following subsections.

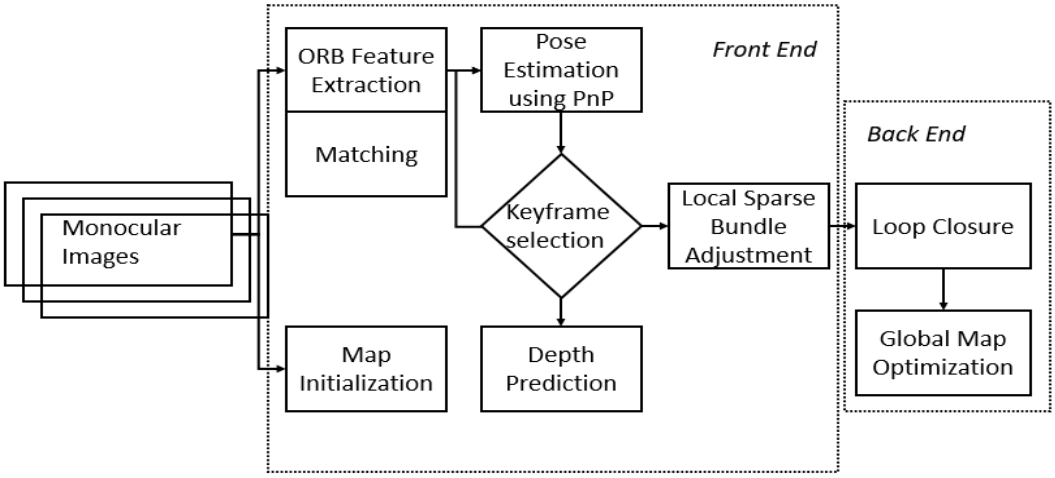

Figure 4.1: A schematic illustration of the RGB-PD SLAM pipeline.

4.1.1 Feature Extraction and Matching

The pipeline begins by extracting ORB (Oriented FAST and Rotated BRIEF) [75] features and storing their descriptors to establish correspondences between images, which are priorly resized to a dimension of 416 x 128 to fit the depth prediction model. Feature detection methods, such as SIFT [76], SURF [77], or ORB, extract distinct points of interest (referred to as key points) from images in the form of corners, blobs and edges that are invariant to scaling, translation and rotation. Their descriptors can then be matched across multiple frames using a feature matching technique such as the Brute-Force matcher or the FLANN matcher. Figure 4.2 illustrates the result of extracting and matching ORB features between two images.

Figure 4.2: Feature extraction and matching using KITTI dataset [18].

4.1.2 Map Initialization

An initial 3D map of the environment is created using the CNN-predicted depth map of the first image in the sequence as opposed to the sophisticated bootstrapping mechanism required in traditional monocular SLAM, where the latter decouples the chicken-egg problem by initially retrieving the relative pose either via a homography [16, p. 325] using DLT [16, p. 88] assuming scene planarity, or by computing the fundamental matrix [16, p. 329] with the eight-point algorithm [16, p. 281] in case of non-planar scenes. This leads to multiple solutions for the camera pose that needs to be disambiguated via a cheriality test [78], which often fails under low parallax. In contrast, a per-pixel depth prediction of the first image initializes the system by reconstructing a part of the environment as an initial map. Given, the camera intrinsic parameter matrix *K*, and the depth *D* of a homogeneous pixel point $\rho = [u\ v\ 1]$ in an image, its 3D projection *P* in the camera frame, *C* can be computed as:

$$P^C = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = DK^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{4.1}$$

A sparse 3D map is created using the extracted ORB features only. However, a dense map of the scene can also be stored at the expense of computational power as shown in Figure 4.3.

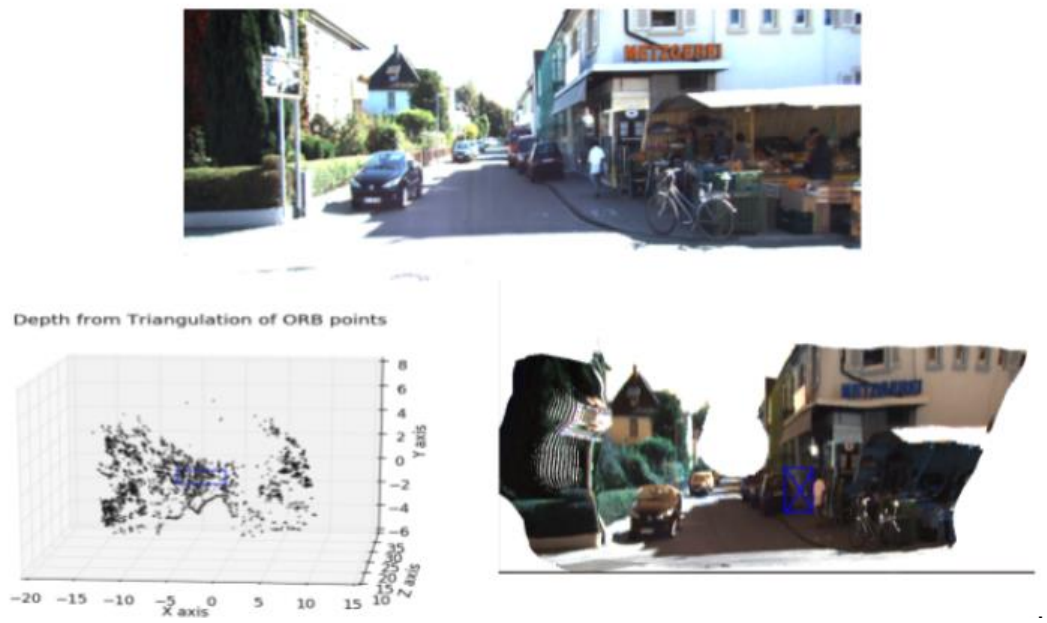

Figure 4.3: Sparse and dense point cloud generation from triangulation and CNN depth predictions respectively.

4.1.3 Camera Pose Estimation

The system defines the global reference frame, *W* at the camera pose for the first frame of the sequence. The pose of each frame is estimated via 2D-3D correspondences between the image and the map. Poor associations are detected and removed using the Lowe's ratio test [76] during matching. As the map size increases linearly with the traveled distance, matching every point in the map becomes infeasible. As a result, only map points that are shared between the current frame and its neighbouring keyframes are considered. The points are filtered using camera frustum culling, where each visible map point is projected into the viewing frustum of the current camera. Besides, points whose descriptors are detected from a radically different angle of view ($\theta > 45°$) have also been discarded. The 6DOF pose for each frame is then estimated with respect to the map using the *Perspective-n-Point (PnP)* algorithm, given a set of *n* ($n \geq 3$) 3D points in the world and their corresponding 2D projections in the image plane such that:

$$\rho = KTP^W, \tag{4.2}$$

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, \tag{4.3}$$

where, $T \in \mathrm{SE}(3)$ is a 4x4 rigid body motion in the homogeneous coordinate system that gives the transformation from the world to the body frame for each timestamp. T is expressed as a pair $\langle R, t \rangle$, where $R \in SO(3)$ is a 3x3 rotation matrix and $t \in \mathbb{R}^3$ is a 3x1 translation vector in the Euclidean space. $SO(3)$ is the group of all rotations, such as, $\forall R \in SO(3)$, where $R^T R = RR^T = I$ and $R^T = R^{-1}$. The absolute pose of the camera in W can be computed as the inverse of the transformation matrix, T. The Random Sample and Consensus (RANSAC) algorithm is commonly used with PnP to make the solution robust to outliers in the set of point correspondences.

4.1.4 Keyframe Selection and Map Expansion

Depth maps are predicted for keyframes only to reduce the computational complexity that arises from the time taken for the network to infer these images. Therefore, the map is expanded only upon the insertion of a keyframe, which is a visually distinct image with informational cues stored in the system when less than 90% of its extracted points are tracked in the last keyframe. Each k_i in a set of keyframes $\kappa = \{k_1 \dots k_n\}$, stores the keyframe pose ${T_k}^W \in SE(3)$, its 2D measurements ρ_k, and the corresponding depth map D_k. The new keyframe, k_i is added as a node in the co-visibility graph that consists of keyframes as nodes and an edge between two keyframes exists if they observe the same map points as shown in Figure 4.4. The remaining unmatched features in k_i are projected to create new map points from its depth map D_k. Additionally, a match with other measurements in the connecting keyframes is searched for each unmatched point in k_i. Matches that fulfill the epipolar constraint are triangulated to form new points in the map. A novel local SBA is then performed in the neighbourhood of the current frame, where all points in the co-

visible area and the keyframe poses are optimized for a fixed window of keyframes. The formulation is elaborated in the following section.

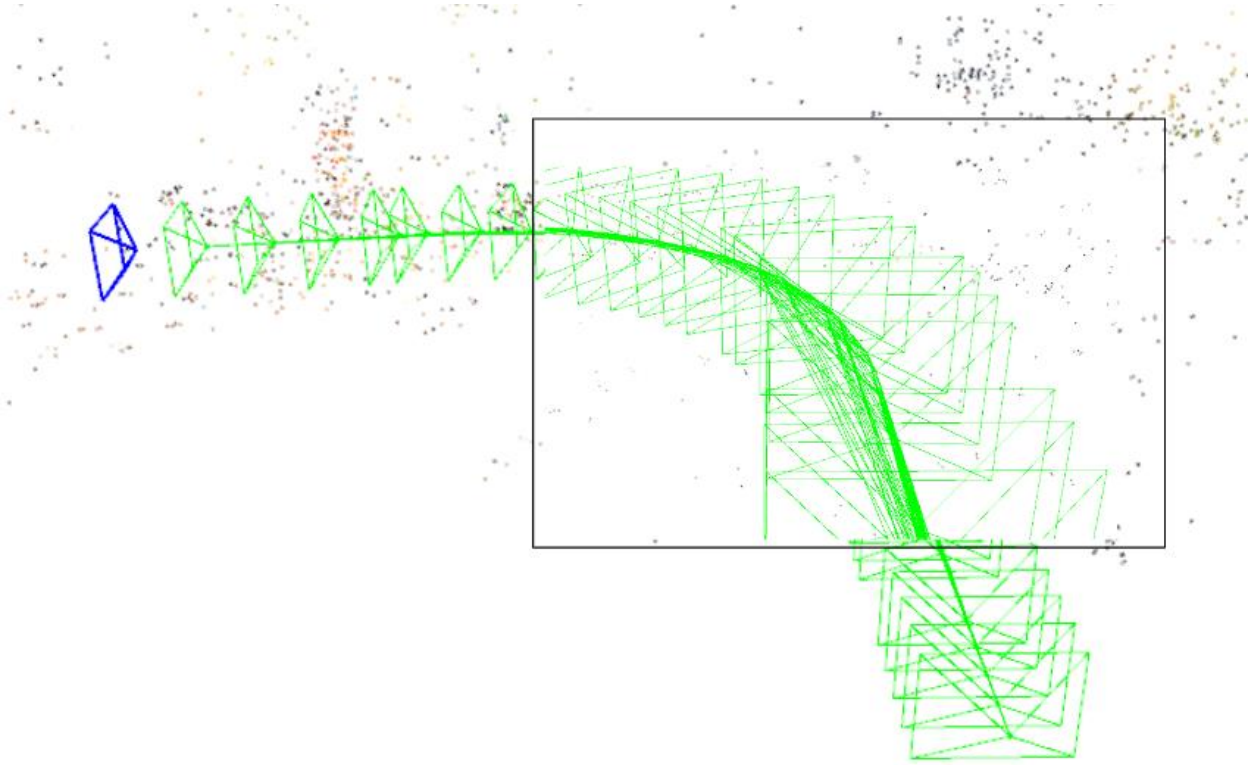

Figure 4.4: Co-visibility graph: edges between the keyframes encode the geometric relationship between them.

4.1.5 Local Sparse Bundle Adjustment (SBA)

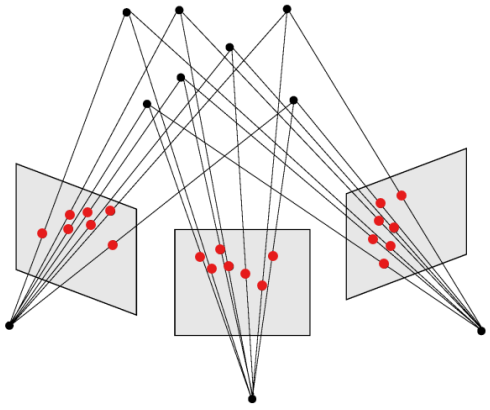

Figure 4.5: Illustration of m=7 3D points and their 2D projections on n=3 keyframes.

Despite extensive validation of the edges in the factor graph prior to optimization, some of them have large errors due to the complex distribution of the predicted depths. In an experiment using the KITTI dataset, 3D positions of 28 ORB points, which could be tracked in 10 consecutives images in that sequence were estimated using their respective depth maps and the ground truth camera positions in the world reference frame. The boxplots in Figure 4.6 illustrate the noise distributions of the depth measurements for those 28 points in the experiment. As can be seen, the points that are closer tend to have smaller variances compared to the ones that are farther away.

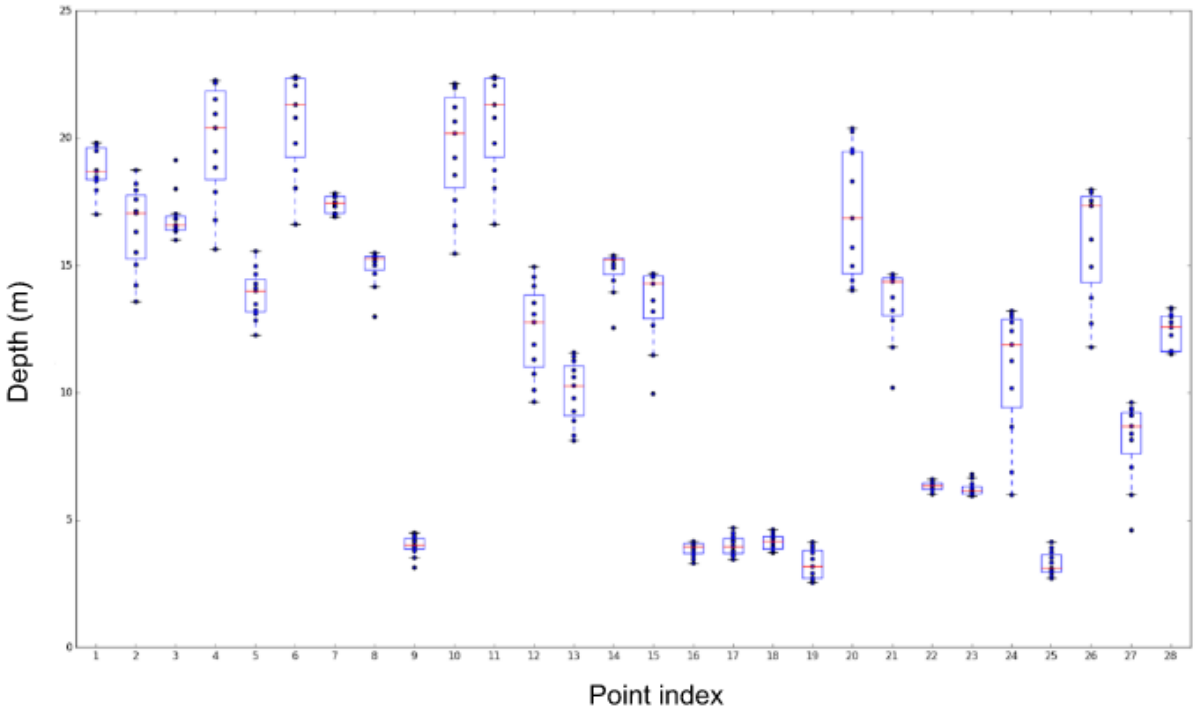

Figure 4.6: Boxplot illustrating the noise distribution of depth predictions.

The proposed approach, therefore, models the uncertainty of measurements using a mixture of Gaussians to jointly optimize the current keyframe k_i, all the keyframes connected to it in the covisibility graph and the shared landmark positions using a least-squares estimation technique called the sparse bundle adjustment (SBA). Given an initial set of *N* keyframe poses $\{{T_1}^W \dots {T_N}^W\}$

viewing m 3D landmark locations $\{{P_1}^T \dots {P_m}^T\}$ from triangulation and m locations $\{{P_1}^p \dots {P_m}^p\}$ from depth predictions, and a family of measurement sets $\{z_1 \dots z_N\}$, where each set z_j contains the measurement $\rho_{i,j}$ of the i^{th} point in the j^{th} keyframe, the goal is to locally optimize the variables by minimizing the below reprojection error:

$$e_{i,j} = \rho_{i,j} - \pi(T_i P_j) \tag{4.4}$$

where, π is the projection function that transforms the points from world reference frame to the image frame. The error function is assumed to be a Gaussian distribution. However, in practice the residuals may not be always Gaussian distributed. A Gaussian Mixture model (GMM) can be used in this case to represent any arbitrary distribution. The joint posterior distribution of the state variables $x = (\Theta, P)$ containing the keyframe poses and 3D locations, where $\Theta = (t_{x,} t_y, t_{z,} \theta^{roll}, \theta^{pitch}, \theta^{yaw})$ is the vector representation of the motion matrix T in the Lie Group (SE3), conditioned on the sensor observations z can then be written as the product of the factor potentials in the factor graph such that:

$$p(x|z) \sim \prod_i \prod_j p\left(z_{i,j}\middle|x\right) \tag{4.5}$$

where, the likelihood function $p\left(z_{i,j}\middle|x\right)$ is the measurement model corresponding to the probability density of observing the j^{th} 3D point from the i^{th} keyframe at the pixel location $z_{i,j}$.

$$p\left(z_{i,j}\middle|x\right) = \max_n \phi_n N(e_n, \mu_n, \sigma_n) \tag{4.6}$$

$N(e, \mu, \sigma)$ represents a different Gaussian distribution of the error function in Eq 4.6 resulting from P^T and P^p. ϕ_n is the probability of the n^{th} component in the given mixture such that $0 \leq \phi_n \leq 1$ and $\sum_{n=1}^{2} \phi_n = 1$. The proposed mixture model type is based on the max operator from [79]

instead of a sum-mixture, because by using the sum-mixture, the maximization of the log-likelihood cannot be converted to a proper least square problem. The summation prevents the logarithm to be pushed all the way into the individual Gaussian component in Eq 4.5. The max operator selects a single Gaussian model (either the one from triangulation or from predictions based on their probability distribution). Once the mixture component with the maximum likelihood is known, the effect of the other mixture component becomes insignificant. The resulting reprojection error, the respective Jacobian, and information matrices for that component are scaled according to the weight ϕ_n and combined in to a least-squares problem, which is then solved using the well-known Levenberg-Marquardt algorithm [80]. If the likelihoods of the two components are tied—a case where the Jacobian is evaluated at a non-differentiable point—one of the two components is selected arbitrarily.

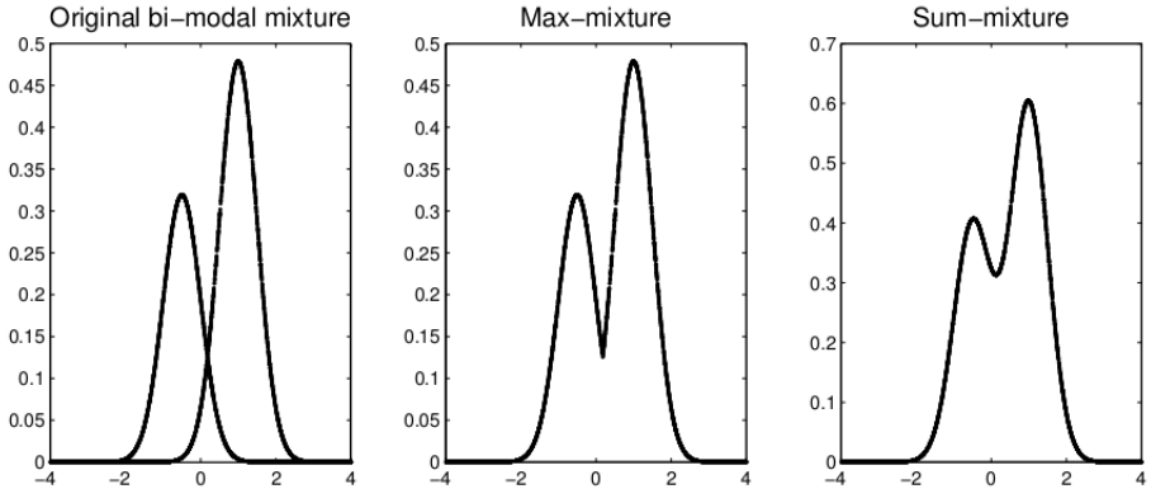

Figure 4.7: Mixture overview: Given two mixture components, the max and the sum mixtures produce different distributions. In both cases, arbitrary distributions can be approximated [79].

The system then looks for loop closure candidates and subsequently performs full BA, where all the map points and the keyframe poses in the global map are optimized to correct

accumulated drift error over a large timescale in the motion estimation. The nearest loop closure candidates can be found by performing a random search throughout the sequence proposed in [81].

4.2 Error Metrics

The RGB-PD SLAM is assessed based on two commonly used error metrics [82]. The absolute trajectory error (ATE) is the average deviation of the estimation from the ground truth trajectory per frame that evaluates the overall performance of the SLAM system. The ATE in meters is calculated as the root-mean-squared-error (RMSE) over a video sequence consisting of N frames such as:

$$ATE_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}|\Delta t_i|^2} \tag{4.7}$$

$$\Delta t = \sqrt{(x-\hat{x})^2 + (y-\hat{y})^2 + (z-\hat{z})^2} \tag{4.8}$$

where, Δt_i is the Euclidean distance between the absolute ground truth and estimated camera positions in the Euclidean space. As a pre-processing step, the true and estimated keyframe poses are aligned using the timestamps [82]. However, it is possible for the trajectories to be specified in arbitrary coordinate frames, for instance when the ground truth reference frame does not necessarily start in the origin. In such cases, the estimated trajectory is mapped onto the ground truth trajectory via a rigid-body transformation obtained using singular value decomposition [83]. For monocular methods, the alignment is followed by a similarity transformation with a scale factor as the scale is unknown. However, the manual alignment process in [18 section VIII E] is not practical and was therefore, skipped during evaluation. Another prominent method to evaluate the performance of visual SLAM systems is the relative pose error (RPE), which measures the drift of the trajectory over a fixed interval of time, frame, or distance Δ. Choosing $\Delta = 1$ is intuitive for systems that match consecutive frames, which then gives the drift per frame, meanwhile $\Delta =$

30 gives the drift per second on a sequence recorded at 30Hz. The relative pose error matrix can be defined as:

$$E_i = (T_i^{-1} T_{i+\Delta})^{-1} (\hat{T}_i^{-1} \hat{T}_{i+\Delta}) \tag{4.9}$$

where, T and $\hat{T}$ are the ground truth and estimated poses. The relative position error (RPE_{pos}) in meters and the relative rotation error (RPE_{rot}) in radians are then calculated as the RMSE of E_i such that:

$$RPE_{pos} = \sqrt{\frac{1}{m} \sum_{i=0}^{m} |trans(E_i)|^2} \tag{4.10}$$

$$RPE_{rot} = \frac{1}{m} \sum_{i=0}^{m} \angle(rot(E_i)) \tag{4.11}$$

$$\angle(rot(E_i)) = \cos^{-1}\left(\frac{Tr(rot(E_i)) - 1}{2}\right) \tag{4.12}$$

where, $rot(E_i)$ and $trans(E_i)$ are the rotation and translation components of E_i respectively. m denotes the number of relative pose error matrices calculated from a sequence of *N* camera poses such that, $m = N - \Delta$. The error is averaged over all possible pairs in both translation and rotation components. Generally, calculating the translation errors is sufficient as rotational errors are estimated as translation errors when the camera is moving.

4.3 SLAM Experiments on KITTI dataset

The odometry benchmark from the KITTI dataset [18] consists of 11 driving sequences with accurate ground truth camera poses obtained via IMU/GPS readings. The dataset consists of color images from a stereo pair and contains depth information from a LiDAR scanner, making it suitable for evaluation. The experiments were conducted using 6 of these sequences covering different scenes but not limited to fast rotations, areas with identical structures, and dynamic objects. We compared the RGB-PD SLAM with traditional monocular ORB SLAM, stereo SLAM

and RGB-D SLAM. Qualitative visual odometry (VO) results in Figure 4.11 shows how the monocular SLAM accumulates drift in scale over time compared to stereo or RGB-PD SLAM due to the unknown scale factor.

Figure 4.8 shows the effect of the number of extracted features on the absolute trajectory error obtained by RGB-PD SLAM running in on-line mode with ORB/ORB features. A total of 6 experiments are conducted, increasing the number of extracted features by 500 each time. Increasing the number of features reduces the error, but a significant reduction is observed after ~2000 key points. Therefore, rest of the experiments was conducted using 2000 ORB features. Tracking fails for less than 1000 features.

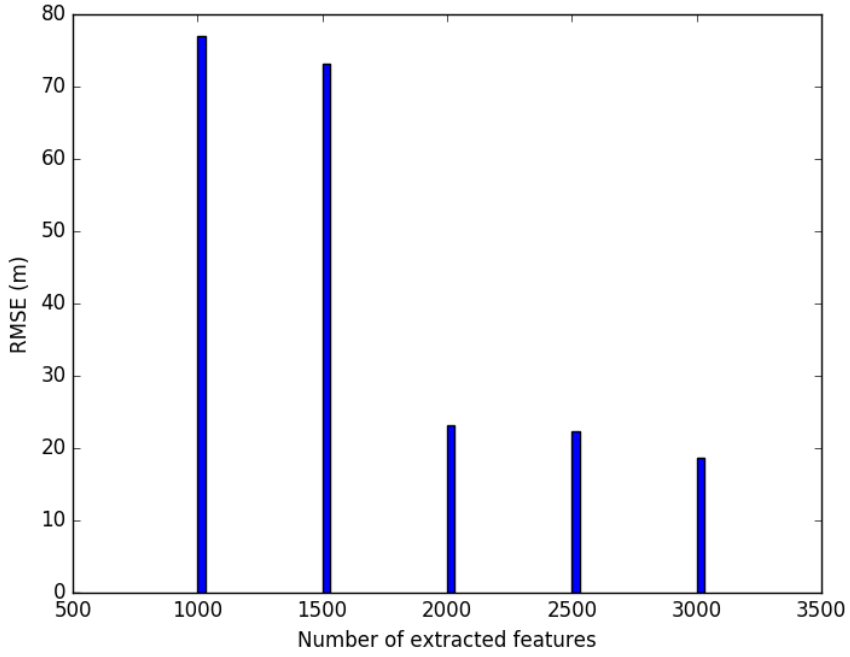

Figure 4.8: ATE obtained by changing the number of ORB features to be extracted on sequence 05.

4.3.1 Comparison with other SLAM systems

Table 4.1 shows the visual odometry results between RGBD-SLAM, monocular SLAM, and stereo SLAM. In all sequences of KITTI odometry dataset, regardless of loop existence, RGB-PD SLAM improves trajectory tracking accuracy compared to the monocular SLAM. Sequences 00, 05, 07 and 09 contain loops, which the systems can detect and successfully close, except for sequences 07 and 09, where the loops are seen only in a few frames at the end of the sequences which the systems cannot always detect. Qualitative results in Figure 4.11 demonstrate how monocular SLAM fails to maintain a consistent scale for KITTI odometry sequences 08 as it could not detect a loop to correct the drift in the trajectory. On the contrary, RGB-PD SLAM can maintain a consistent scale even for a long video by leveraging the scale-consistent depth predictions.

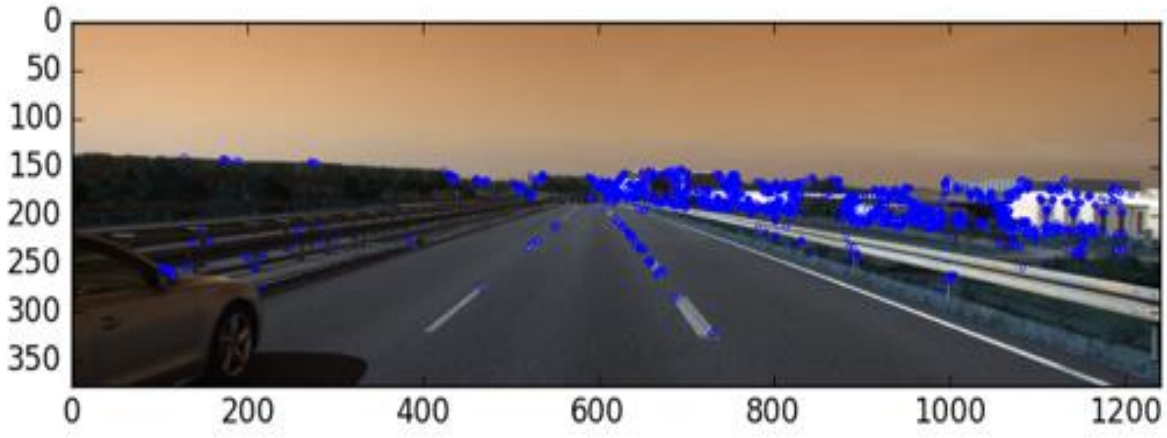

Figure 4.9: Extracted ORB points for sequence 01.

Besides, the proposed method can process entire sequence 01, for which the monocular SLAM fails to detect features as it consists of a highway with few trackable close objects as shown in Table 4.1. The ability of RGB-PD SLAM to create points from a single frame instead of the delayed initialization of the monocular SLAM, where it strives to find matches between two

keyframes, is vital to not lose tracking. However, the estimated trajectory is grossly incorrect as can be seen in Figure 4.12. This is because, the tracked points in a highway are mostly at a distance, and as discussed in the previous chapter, the predictions are generally accurate for points that are closer, precisely, within a depth range of 50m from the sensor. As the vehicle is moving at a high speed, more points need to be tracked faster for the system to not lose tracking. Thus, taking the distant points into consideration, leads to inaccuracy in the estimated trajectory. However, as the stereo system estimates the map and the trajectory with metric scale, it does not suffer from scale drift as much as the RGB-PD SLAM.

Table 4.1: Visual Odometry results on KITTI dataset.

Sequences	**ORB Monocular SLAM**			**Stereo SLAM**			**RGB-PD SLAM**		
	ATE (m)	$\boldsymbol{RPE_{pos}}$ **(m)**	$\boldsymbol{RPE_{rot}}$ **(deg/s)**	**ATE (m)**	$\boldsymbol{RPE_{pos}}$ **(m)**	$\boldsymbol{RPE_{rot}}$ **(deg/s)**	**ATE (m)**	$\boldsymbol{RPE_{pos}}$ **(m)**	$\boldsymbol{RPE_{rot}}$ **(deg/s)**
00	94.34	25.46	4.14	11.50	0.70	0.25	34.21	6.46	3.28
01	X	X	X	8.32	0.41	0.13	314.70	12.34	0.74
05	140.68	37.91	3.26	1.57	1.10	0.86	22.89	9.88	1.20
07	36.70	2.53	1.63	2.69	0.17	0.17	17.21	1.18	1.14
08	193.4	5.63	1.64	7.97	0.26	0.17	21.20	0.30	0.65
09	68.61	4.79	1.27	13.10	0.27	0.23	32.14	1.98	0.51

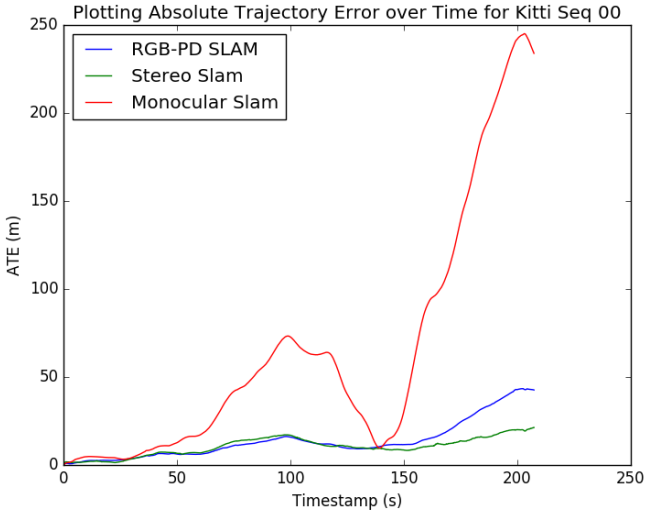

Figure 4.10: Absolute Trajectory Error over time for sequence 00.

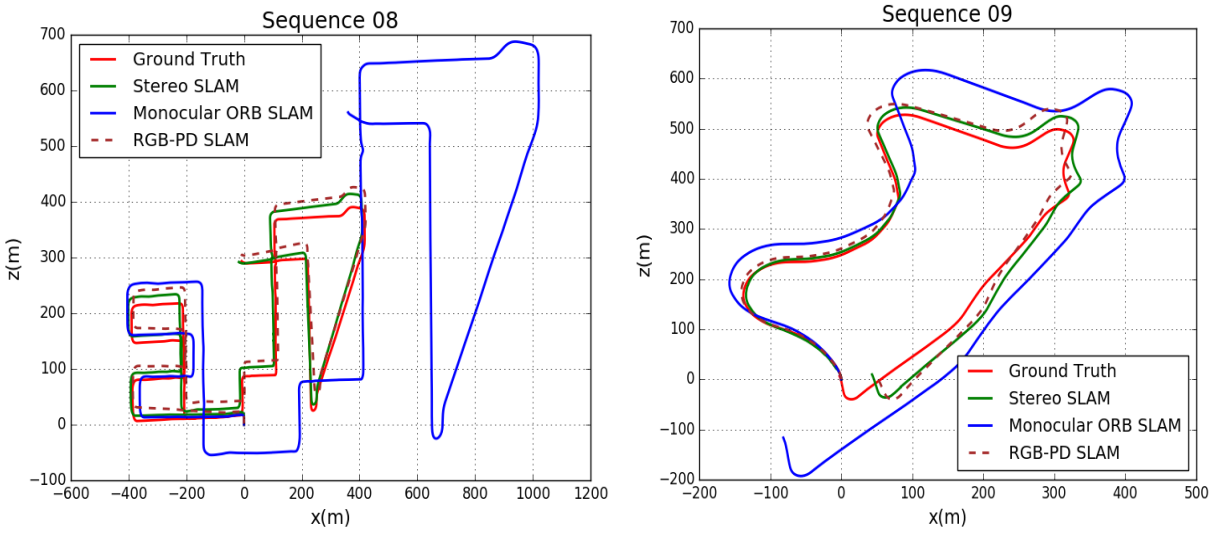

Figure 4.11: Estimated trajectory on sequences 08 and 09.

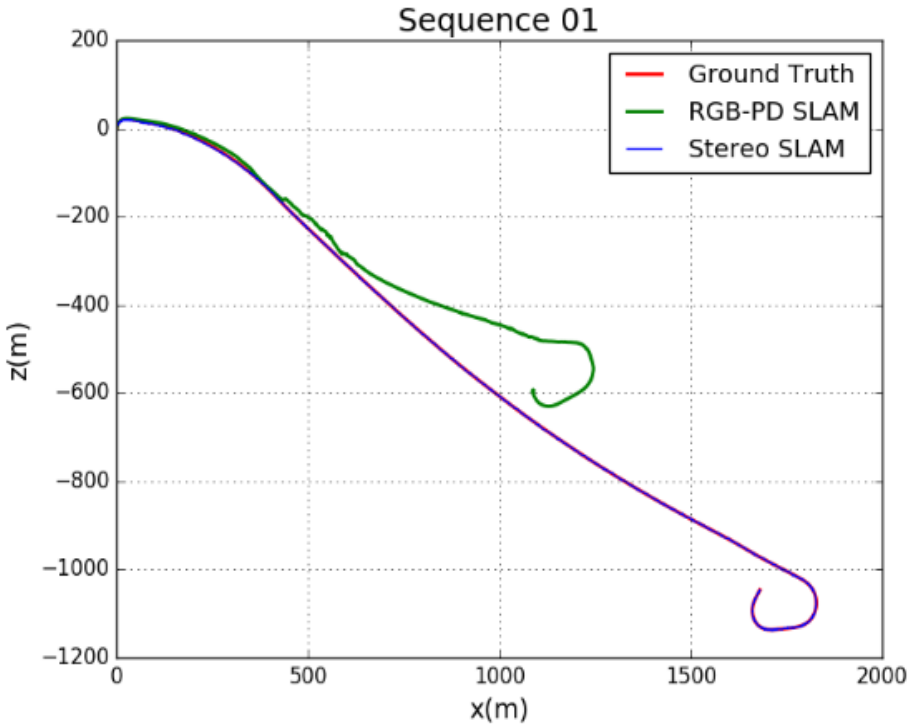

Figure 4.12: Estimated trajectory on sequence 01.

Figure 4.13 illustrates the relative translation error and relative rotation error of the three systems with time. The RPE_{tran} for monocular SLAM is seen to increase near the end of the sequences due to the drift in scale. In most cases, the RPE obtained from RGB-PD SLAM is tenfold higher than stereo SLAM but much less compared to that of monocular SLAM.

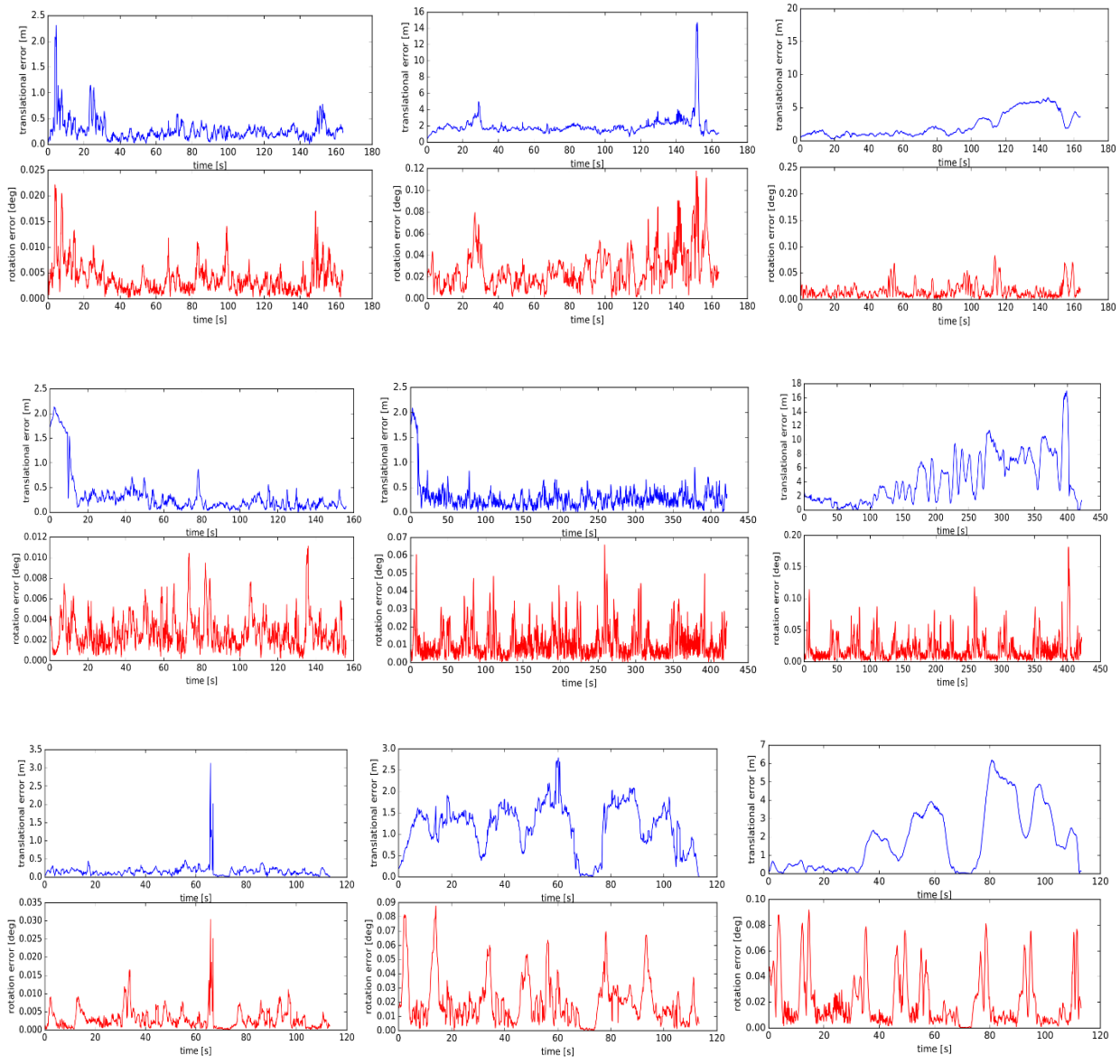

Figure 4.13: Relative translation error (blue) and relative rotation error (red) between stereo, RGB-PD and monocular SLAM (from left to right) on KITTI sequences 09, 08 and 07 (top to bottom).

4.3.2 Comparison with contemporary SIDE methods

Adding the geometric constraint solved the diminishing scale problem of the unsupervised SfM-Learner [14] as discussed earlier in this thesis. However, even after their depth predictions are scaled by a scalar s that matches the median with the ground truth, the proposed method

performs better in estimating the camera pose in the SLAM system as shown in Figure 4.14 and Table 4.2. Semi-supervised SIDE methods such as Monodepth [12], on the other hand, exhibit even superior results in pose estimation as they benefit from the known transformation between the cameras in the stereo rig during training. Nevertheless, their inference time is much higher compared to unsupervised depth predictions. This is because semi-supervised methods suffer from disparity ramps on both the left side of the image and of the occlusion as an effect of stereo disocclusions. As a remedy, a final post-processing step is required, where the input, at the time of inference, is horizontally flipped and a disparity map for the flipped image is also computed. Both the disparity maps are then combined to form the final disparity map. This final step leads to better accuracy but grossly increases computation time.

Stereo trained methods can estimate depths at metric scale, which are readily optimized in existing SLAM frameworks. By contrast, monocular trained methods suffer from scale inconsistent issue making the post-optimization step non-trivial. However, incorporating the scale-consistent depth predictions allows the optimization of map points and poses and thus lowers the ATE significantly.

Table 4.2: Odometry results between SfM-Learner, Monodepth and RGB-PD SLAM.

Sequences	SfM-Learner			Monodepth			RGB-PD		
	ATE (m)	RPE_{pos} (m)	RPE_{rot} (deg/s)	ATE (m)	RPE_{pos} (m)	RPE_{rot} (deg/s)	ATE (m)	RPE_{pos} (m)	RPE_{rot} (deg/s)
00	38.49	7.21	4.02	16.32	3.52	2.22	34.21	6.46	3.28
05	80.86	7.56	4.84	17.89	2.80	1.09	22.89	9.88	1.20
07	59.08	8.89	5.23	7.49	0.86	0.97	17.21	1.18	1.14

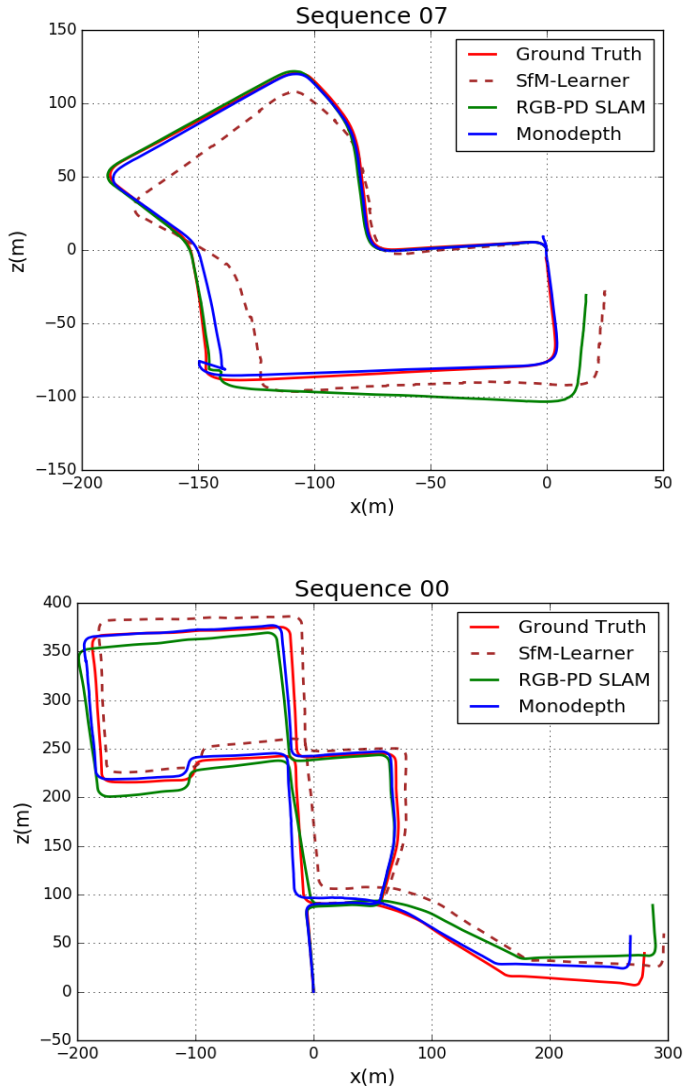

Figure 4.14: Estimated trajectory on sequences 07 and 00 using contemporary SIDE methods.

4.4 Augmented Reality

Augmented Reality (AR) is an advanced visualization technology, where virtual or computer-generated objects are superimposed on a user's view of the real world. AR has been widely used in gaming and entertainment sectors, now expanding its usage in education, medical fields, robot path planning, repair, maintenance, as well as in military aircraft navigation and aiming. For a seamless AR experience, it is imperative that synthetic elements are rendered and aligned in the scene with accuracy. The solution to this problem can be related to the camera localization problem. A widely acknowledged definition of AR and related taxonomies can be found in Azuma's survey pertaining to the topic [84]. Billinghurst et al. [85] later survey almost 50 years of research and development in the field of AR. However, most of these definitions emphasize SLAM, or principally, ego-motion estimation as a basic problem in the integration of real and virtual worlds.

4.4.1 Overview of the Problem

Given a camera frame F_c and a world frame F_w in the real world, the goal is to align the virtual world, F_{vw} with the real world. To create an image of the virtual world with the real camera current view, the virtual camera, F_{vc} and F_c should be located at the same position and have the same intrinsic parameters. Once the real and virtual cameras are aligned, a compositing step provides the resulting augmented image. Within this process, the only unknown is the real camera position with respect to the world frame, and thus, vision-based AR is restricted to pose estimation problems. The rigid transformation, ${}^{c}T_{w}$ that defines the position and orientation of the camera with respect to the world can be expressed using Eq 4.3.

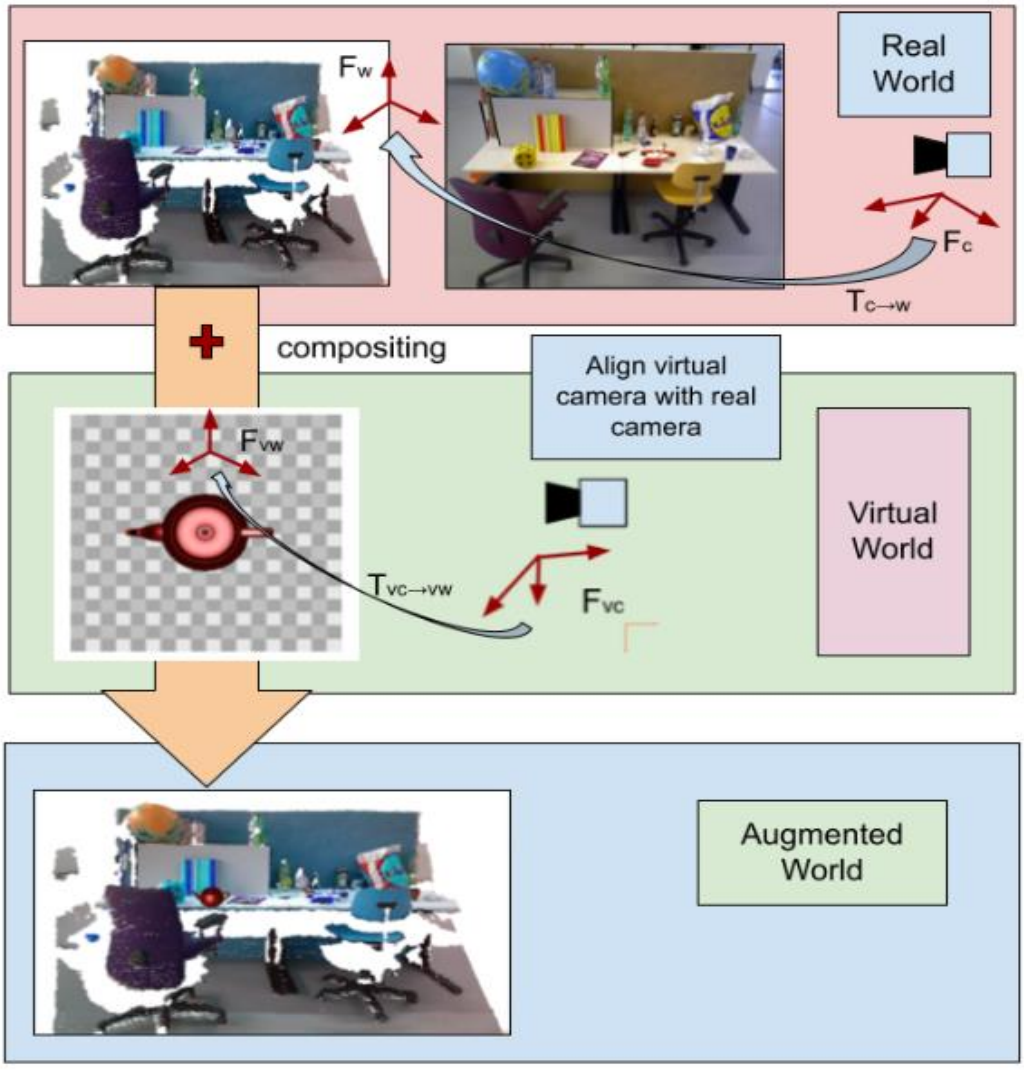

Figure 4.15: General principle of the registration process in AR.

For N number of 3D points $P_i, i = 1..N$, whose coordinates in the homogeneous system are expressed as $P_i = (X_i, Y_i, Z_i, 1)^T$ in the world frame, their corresponding perspective projection $p_i = (u_i, v_i, 1)^T$ in the image plane can be defined using Eq 4.2. Therefore, any error in the estimation of the absolute camera pose appears as an inconsistency to the user. The fundamental idea of the registration process is illustrated in Figure 4.15.

4.4.2 Camera Pose Estimation in AR

I. Pose Estimation using Fiducial Markers

Until the early 2000s, almost all vision-based techniques relied on fiducial markers (Figure 4.16), which were used as points of reference for the positional tracking of AR devices. The locations of these markers in the world are known as a priori and the camera's poses are computed with respect to those positions. Later, various marker-less approaches such as SLAM and object detection using feature matching emerged in the literature, which allowed easy estimation of the camera pose and registration of objects. Amongst various solutions, two classical methods for pose estimation are most considered in augmented reality: coplanar based techniques and 3D model based techniques. More information on the methods can be found in [86].

Figure 4.16: Fiducial Markers for AR.

II. Pose Estimation using planar scenes or 2D models

Taking planar scenes, *i.e.*, without depth information, into account simplified the pose estimation problem in AR. Considering a 2D model on which the virtual object needs to be augmented as the reference image (as illustrated in Figure 4.17), the objective narrows down to the retrieval of camera transformation between the captured and the reference images. An efficient solution is through the estimation of homography between point correspondences.

Figure 4.17: Reference image that needs to be augmented (top left). Tracking the reference image in the environment using feature extraction and matching to compute homography (top right). Replacement image (bottom left). Augmented scene (bottom right).

A point p_1 in image I_1 is related to a point p_2 in the reference image I_2 by the homography H_2^1 as follows:

$$p_1 = H_2^1 p_2, \tag{4.13}$$

H_2^1 can be estimated using the DLT algorithm or by minimizing a cost function, that is the geometric distance using a gradient approach such as Gauss-Newton. The homography can be decomposed to retrieve the pose. As a result, in the estimation of the initial pose, it assumes that all the points lie in the plane $Z = 0$. In such cases, the i^{th} 3D point can be written as $P_i = (X_i, Y_i, 0,1)^T$. The projection in Eq 4.2 thus becomes:

$$p_i = KTP_i = K[r_1 \quad r_2 \quad r_3 \quad t]\begin{bmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{bmatrix}, \tag{4.14}$$

where r_j is the j[th] column of the rotation matrix that can be re-written as:

$$p_i = H[X_i \quad Y_i \quad 1]^T, \tag{4.15}$$

$$H = [r_1 \quad r_2 \quad t]^T, \tag{4.16}$$

Considering that the rotational matrix is orthogonal, the third column of the matrix is computed as $r_3 = r_1 \times r_2$. Like the pose, homographies can be chained between consecutive frames:

$$^{N}H_0 = \prod_{n=2}^{N} {}^{n}H_{n-1}\,{}^{1}H_0. \tag{4.17}$$

Nevertheless, this leads to error accumulation over time. The simplicity of this approach made its use in AR a standard, however many researchers still consider this as a marker-based approach, given that a reference model needs to be available in the environment. Besides, object detection without sufficient texture or with repetitive patterns is challenging. Figure 4.18 displays how the lack of extracted features (ORB) from the reference image causes feature matching, and therefore pose estimation to be unsuccessful. As a result, the algorithm fails. With regards to that, active research is being conducted to estimate the pose from a 3D model of the environment that the camera is viewing.

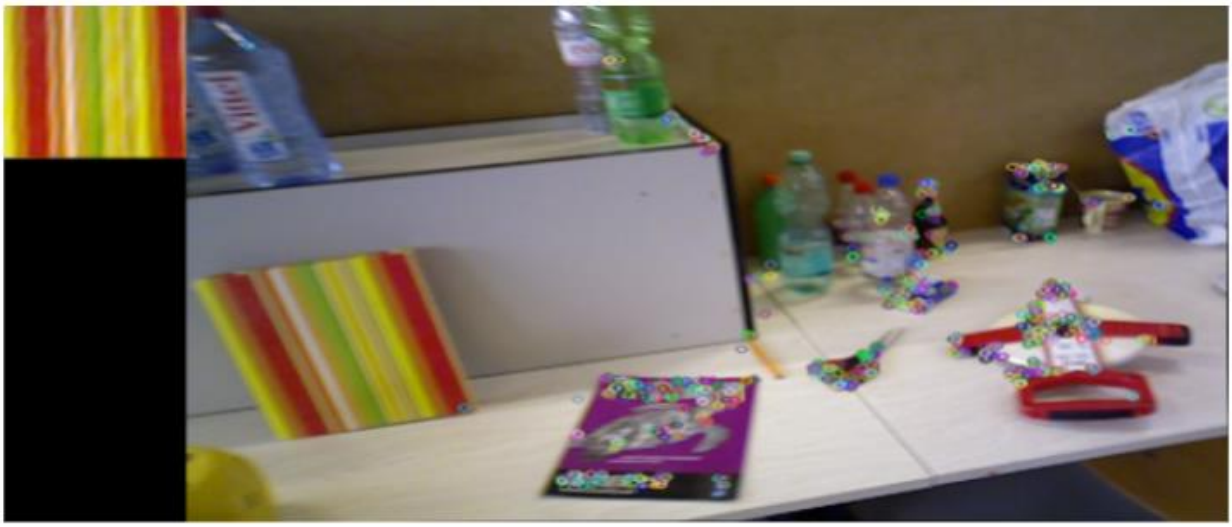

Figure 4.18: Feature extraction fails for objects with repetitive patterns.

III. Pose Estimation using the 3D environment

The pose is estimated via the constraints between 2D information in the images and the 3D features of the environment or more generally using the *PnP* solution introduced in section 4.1.2. Often a comprehensive or even a sparse 3D model of the environment is not known beforehand. The idea is to then reconstruct the scene and perform camera localization within the same framework via SLAM as mentioned earlier. In this case, we estimate a dense structure of the long_office_household sequence from the TUM dataset [82] using the depth prediction model trained on NYUv2 mentioned in Chapter 3. A region of interest (ROI) expressed in pixels is selected in the initial image and the virtual object (red teapot) is rendered at the center of the ROI in 3D space using Eq 4.2. Once the absolute camera pose is calculated, the position and orientation of the virtual object (M_{model}) w.r.t to the camera is updated as:

$$M_{model} = c_{T_w} \times M_{model}, \tag{4.18}$$

The final augmented image is visualized using the OpenGL library and Pangolin and is shown in Figure 4.19. It is worth noting that reconstruction, camera pose estimation and registration can be done for real time images using the proposed depth prediction models, given that the camera intrinsic parameters are known.

Figure 4.19: The result of rendering a synthetic teapot in the reconstructed scene of an indoor office environment from TUM dataset [82] using the proposed depth prediction model trained on NYUv2 [69] dataset.

4.4.3 Effect of Inconsistent Depth and Camera Trajectory Estimations

It is important that the scale of the reconstructed scene is consistent for the virtual object to be rendered realistically. This is particularly important in AR applications, where the scale of the virtual object needs to be taken into consideration, for instance, in medicine and engineering that have an impact on human lives and safety. One example is the detection of buried facilities, where computer-generated images of the pipelines are superimposed on the user's view of the real world for accurate tracking. Municipalities and utility companies maintain a wide mesh of underground infrastructures, such as sewers and electric cables for public services which are often densely packed and difficult to access. This combination of complexity and latency frequently leads to increased costs and labor for any excavation-related task. AR facilitates locating the

occluded underground utilities by projecting holograms of pipelines, conduits, and electric cables on the real-time image of the site.

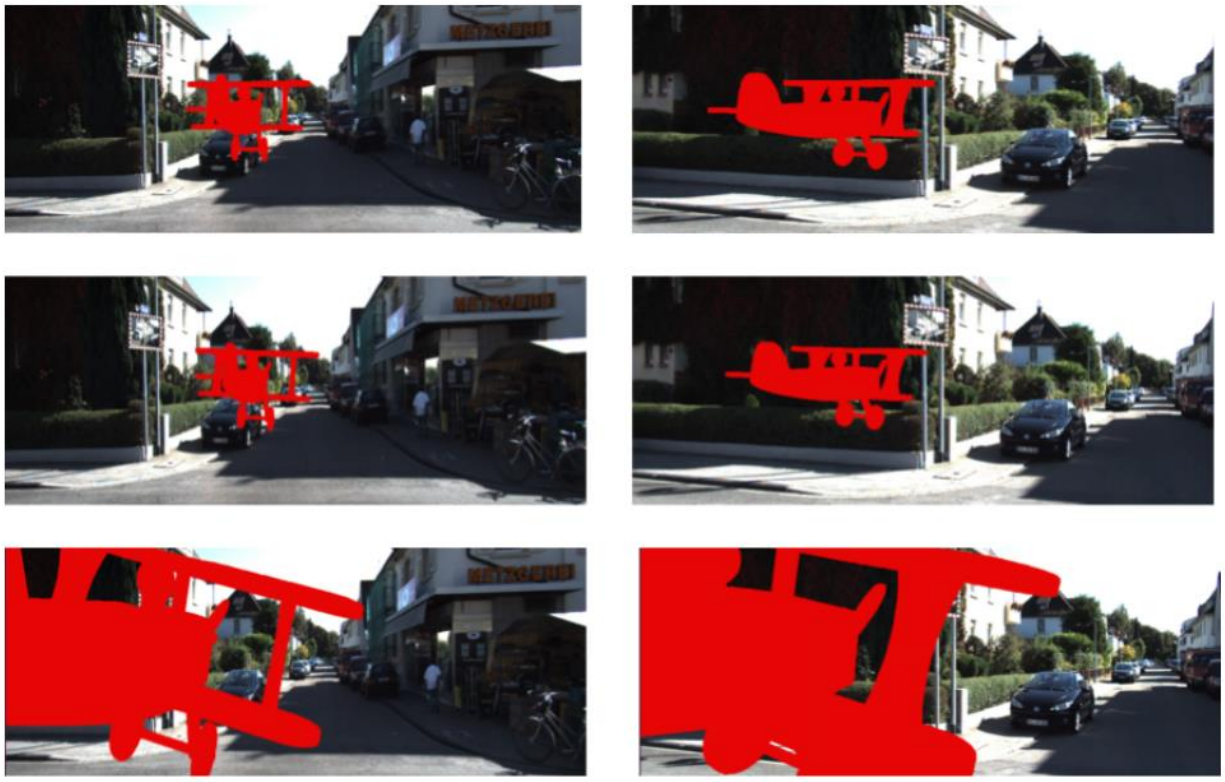

Figure 4.20: Rendering a synthetic toy plane using pose estimations from RGB-PD SLAM and traditional monocular SLAM.

In a simple experiment, a virtual toy plane object is rendered onto the images from sequence 07 of the KITTI odometry dataset. The object is set to have a fixed position and orientation in the world reference with respect to the first frame, i.e., it's position in the environment is fixed. Figure 4.20 exemplifies excerpts from the video that illustrates how the toy plane appears with respect to different camera poses as the camera moves in the environment. The pose of the plane is updated with respect to the current camera pose at framerate. Row 1 of the figure is the result of using ground truth poses, while row 2 uses the RGB-PD SLAM and row 3 uses monocular SLAM. The plane appears slightly smaller in row 2 due to the error in the

estimation of pose. However, the scale uncertainty associated with monocular SLAM leads to a large drift in scale and as a result, to a substantial positioning error as seen in the figure- the sensor is estimated to be much closer to the object than it is in reality. The part of the trajectory used for the illustration of this AR application is plotted in Figure 4.21.

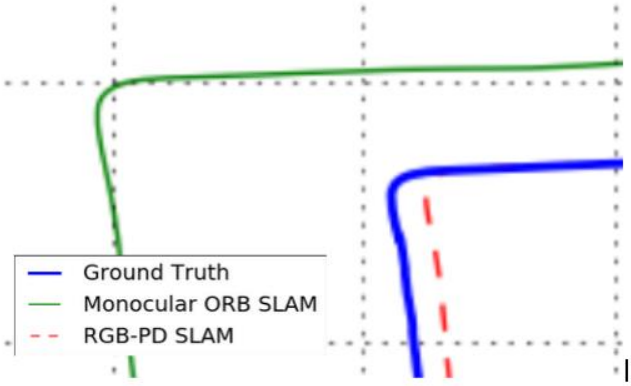

Figure 4.21: Part of trajectory from sequence 07, where large drift in scale is seen for monocular odometry compared to RGB-PD SLAM.

4.5 Discussion

Across most of the sequences from the KITTI dataset, the proposed RGB-PD SLAM shows competitive performance when compared to stereo SLAM without the need for additional sensors. The monocular SLAM system infused with scale-consistent depth predictions leverages the benefits of deep learning and has multiple advantages over existing monocular solutions:

- It enables easy initialization of the system compared to conventional monocular SLAM that depends on triangulation of corresponding features between multiple frames with sufficient parallax between them. In contrast, the proposed system can initialize at any time by directly forecasting a per-pixel depth map for the input image like a depth sensor but without the need for additional devices.

- Traditional monocular SLAM suffers from drift in scale especially in long sequences as the scale is ambiguous. SLAM with the proposed SIDE method can maintain a consistent scale over the entire video sequence as a benefit of the added geometric constraint during the training of the network.
- The tracking system in traditional feature-based SLAM often fails in the presence of repetitive patterns or in the lack of trackable features (when the 3D map is over sparse).
- Monocular ORB-SLAM can only provide a sparse map of the environment. Meanwhile, the proposed method enables dense spatial reconstruction of the scene by leveraging the per-pixel depth predictions.

Besides, the proposed method not only solves the diminishing problem of the unsupervised SfM-Learner, but also improves accuracy of the overall trajectory. Semi-supervised methods leverage the known translation between the cameras in the stereo rig and therefore, performs nearly as well as stereo SLAM. Later, the importance of scale in SLAM is shown using a simple application of Augmented Reality, where the large drift in monocular SLAM leads to inaccurate positioning of the virtual object in the real environment.

5. Conclusion

5.1 Summary

This thesis explores the idea of using scale consistent depth predictions from a neural network to achieve more accurate SLAM and AR. Previous literature has used unsupervised learning to predict depth from single images. However, most such methods only leverage temporal constraints during training by minimizing a photometric loss between adjacent images, often disregarding the scene geometry or the spatial constraints. Therefore, Chapter 3 introduces a novel geometric consistency loss that is mindful of the global structure of the environment by penalizing the misalignment in pose and depth predictions between multiple frames during training. This enforces the network to predict more scale consistent depth maps over a sequence of images, where the predictions in a single sequence are coherent with each other. This is imperative in map fusion and in the estimation of the sensor pose (the sensor being a monocular camera in this case) in SLAM systems for applications such as autonomous navigation and AR. The network has been trained and tested on both indoor (NYUv2) and outdoor datasets (KITTI). Incorporating the consistency loss has been shown to generate sharper maps compared to the baseline [14] with an overall 13.03% decrease in the RMSE, trained and evaluated on the KITTI dataset.

The efficacy of the depth prediction model in real time applications is further shown in Chapter 4 by fusing it into a visual monocular ORB SLAM system to provide direct range measurements. However, as the neural network is trained using unstructured monocular videos in an unsupervised manner, the predictions still suffer from per-frame ambiguity. As a result, the fused 3D structures from multiple images contains significant noise. A novel local optimization step called the sparse bundle adjustment (SBA) using probabilistic approach is, therefore,

introduced in the framework to handle the noises in the predictions. The map points and keyframe poses are locally optimized while simultaneously determining which depth estimations are accurate within a single Bayesian framework. The proposed RGB-PD SLAM is seen to improve the accuracy and robustness of the visual odometry components compared to traditional monocular SLAM, which suffers from large drifts due to the unknown scale factor associated with it. However, SLAM using binocular cameras can triangulate depth in metric scale by exploiting the known transformation between them, and therefore produces more accurate odometry results. Besides, adding the geometric constraint solved the diminishing scale problem of the unsupervised SfM-Learner, which follows from the regularization term in the smoothness loss and therefore, greatly reduces the error in the absolute and relative trajectories. The chapter further outlines the current pose estimation problems in AR and discusses the importance of scale consistent estimations using a simple example.

5.2 Future Work

Although this thesis has addressed the key issues in the problem of unsupervised depth estimation from single images, there are several limitations that still need to be addressed. Firstly, most self-supervised methods underperform in nighttime scenarios. This is because, multiple light sources cause drastic changes in illumination between adjacent frames that make the photometric loss futile in supervising the networks. A potential solution is to devise a feature metric loss that can aid the models to learn more robust feature representations that are invariant to changes in lighting conditions.

The pose network in the proposed framework is principally used to train the depth network in an unsupervised manner. It will be interesting to fuse the predictions from the pose network into the SLAM system instead of using an *N-point* algorithm or the concept of constant motion model

to estimate an initial pose from the previous frame, which simply assumes that the camera motion is the same as the previous frame. This assumption is often violated in real scenarios involving abrupt camera motions. In such cases, the relative pose predictions can be used to find an initial pose for the current frame during tracking before performing local bundle adjustment.

Thirdly, despite the ongoing progress in unsupervised SIDE methods, they can not be fully deployed in real-world applications yet. Performance of self-supervised methods is still inferior compared to supervised or semi-supervised SIDE methods. However, they can be used to achieve high level computer vision tasks such as 3D object detection and reconstruction as they can learn from a large number of unlabelled data that is often available. Furthermore, these models can be easily transferred to new scenarios by fine tuning. A fascinating prospect in the future will be to insert the unsupervised SIDE model into more high-level applications and discover new scopes to improve the overall performance.

Bibliography

[1] Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, *33*(5), 1255-1262.

[2] Klein, G., & Murray, D. (2007, November). Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality* (pp. 225-234). IEEE.

[3] Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, *9*(1),

[4] Latif, R., & Saddik, A. (2019, April). SLAM algorithms implementation in a UAV, based on a heterogeneous system: A survey. In *2019 4th World Conference on Complex Systems (WCCS)* (pp. 1-6). IEEE.

[5] Lee, S., & Lee, S. (2013). Embedded visual SLAM: Applications for low-cost consumer robots. *IEEE Robotics & Automation Magazine*, *20*(4), 83-95.

[6] Hofmann-Wellenhof, B., Lichtenegger, H., & Collins, J. (2012). *Global positioning system: theory and practice*. Springer Science & Business Media.

[7] Ahmad, N., Ghazilla, R. A. R., Khairi, N. M., & Kasi, V. (2013). Reviews on various inertial measurement unit (IMU) sensor applications. *International Journal of Signal Processing Systems*, *1*(2), 256-262.

[8] Cole, D. M., & Newman, P. M. (2006, May). Using laser range data for 3D SLAM in outdoor environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* (pp. 1556-1563). IEEE.

[9] Kim, H. D., Seo, S. W., Jang, I. H., & Sim, K. B. (2007, October). SLAM of mobile robot in the indoor environment with digital magnetic compass and ultrasonic sensors. In *2007 International Conference on Control, Automation and Systems* (pp. 87-90). IEEE.

[10] Eigen, D., & Fergus, R. (2015). 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).

[11] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., & Navab, N. (2016, October). Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)* (pp. 239-248). IEEE.

[12] Godard, C., Mac Aodha, O., & Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 270-279).

[13] Aleotti, F., Tosi, F., Poggi, M., & Mattoccia, S. (2018). Generative adversarial networks for unsupervised monocular depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops* (pp. 0-0).

[14] Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1851-1858).

[15] Bian, J., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M. M., & Reid, I. (2019). Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems, 32*.

[16] Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[17] Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, *31*(5), 1147-1163.

[18] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, *32*(11), 1231-1237.

[19] Hoiem, D., Efros, A. A., & Hebert, M. (2005). Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers* (pp. 577-584).

[20] Zhao, C., Sun, Q., Zhang, C., Tang, Y., & Qian, F. (2020). Monocular depth estimation based on deep learning: An overview. *Science China Technological Sciences*, *63*(9), 1612-1627.

[21] Mertan, A., Duff, D. J., & Unal, G. (2022). Single image depth estimation: An overview. *Digital Signal Processing*, 103441.

[22] Li, Q., Zhu, J., Liu, J., Cao, R., Li, Q., Jia, S., & Qiu, G. (2020). Deep learning based monocular depth prediction: Datasets, methods and applications. *arXiv preprint arXiv:2011.04123*.

[23] Dijk, T. V., & Croon, G. D. (2019). How do neural networks see depth in single images?. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2183-2191).

[24] Gibson, J. J. (1950). The perception of the visual world.

[25] Saxena, A., Chung, S., & Ng, A. (2005). Learning depth from single monocular images. *Advances in neural information processing systems*, *18*.

[26] Hua, Y., & Tian, H. (2016). Depth estimation with convolutional conditional random field network. *Neurocomputing*, *214*, 546-554.

[27] Cao, Y., Wu, Z., & Shen, C. (2017). Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(11), 3174-3182.

[28] Eigen, D., & Fergus, R. (2015). 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).

[29] Weerasekera, C. S., Latif, Y., Garg, R., & Reid, I. (2017, May). Dense monocular reconstruction using surface normals. In *2017 IEEE international conference on robotics and automation (ICRA)* (pp. 2524-2531). IEEE.

[30] Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., & Fragkiadaki, K. (2017). Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*.

[31] Garg, R., Bg, V. K., Carneiro, G., & Reid, I. (2016, October). Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision* (pp. 740-756). Springer, Cham.

[32] Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., & Brox, T. (2017). Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5038-5047).

[33] Bian, J. W., Zhan, H., Wang, N., Li, Z., Zhang, L., Shen, C., ... & Reid, I. (2021). Unsupervised scale-consistent depth learning from video. *International Journal of Computer Vision*, *129*(9), 2548-2564.

[34] Godard, C., Mac Aodha, O., Firman, M., & Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 3828-3838).

[35] Yin, Z., & Shi, J. (2018). Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1983-1992).

[36] Wang, C., Buenaposada, J. M., Zhu, R., & Lucey, S. (2018). Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2022-2030).

[37] Zhao, W., Liu, S., Shu, Y., & Liu, Y. J. (2020). Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9151-9161).

[38] Luo, H., Gao, Y., Wu, Y., Liao, C., Yang, X., & Cheng, K. T. (2018). Real-time dense monocular SLAM with online adapted depth prediction network. *IEEE Transactions on Multimedia*, *21*(2), 470-483.

[39] Lin, X., Sánchez-Escobedo, D., Casas, J. R., & Pardàs, M. (2019). Depth estimation and semantic segmentation from a single RGB image using a hybrid convolutional neural network. *Sensors*, *19*(8), 1795.

[40] Zama Ramirez, P., Poggi, M., Tosi, F., Mattoccia, S., & Di Stefano, L. (2018, December). Geometry meets semantics for semi-supervised monocular depth estimation. In *Asian Conference on Computer Vision* (pp. 298-313). Springer, Cham.

[41] Sanchez-Escobedo, D., Lin, X., Casas, J. R., & Pardas, M. (2018, April). Hybridnet for depth estimation and semantic segmentation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1563-1567). IEEE.

[42] Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., & Black, M. J. (2019). Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12240-12249).

[43] Zhou, J., Wang, Y., Qin, K., & Zeng, W. (2019). Moving indoor: Unsupervised video depth learning in challenging environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 8618-8627).

[44] Yin, W., Liu, Y., Shen, C., & Yan, Y. (2019). Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5684-5693).

[45] Qi, X., Liao, R., Liu, Z., Urtasun, R., & Jia, J. (2018). Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 283-291).

[46] Li, B., Shen, C., Dai, Y., Van Den Hengel, A., & He, M. (2015). Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1119-1127).

[47] Chen, K., Pogue, A., Lopez, B. T., Agha-Mohammadi, A. A., & Mehta, A. (2021, September). Unsupervised monocular depth learning with integrated intrinsics and spatio-temporal constraints. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2451-2458). IEEE.

[48] Mahjourian, R., Wicke, M., & Angelova, A. (2018). 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).

[49] Prasad, V., & Bhowmick, B. (2019, January). Sfmlearner++: Learning monocular depth & ego-motion using meaningful geometric constraints. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 2087-2096). IEEE.

[50] Zhao, C., Yen, G. G., Sun, Q., Zhang, C., & Tang, Y. (2020). Masked GAN for unsupervised depth and pose prediction with scale consistency. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(12), 5392-5403.

[51] Zou, Y., Luo, Z., & Huang, J. B. (2018). Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 36-53).

[52] Yin, X., Wang, X., Du, X., & Chen, Q. (2017). Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields. In *Proceedings of the IEEE international conference on computer vision* (pp. 5870-5878).

[53] Zhao, W., Liu, S., Shu, Y., & Liu, Y. J. (2020). Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9151-9161).

[54] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[55] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[56] Zhang, H., Shen, C., Li, Y., Cao, Y., Liu, Y., & Yan, Y. (2019). Exploiting temporal consistency for real-time video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1725-1734).

[57] Wang, R., Pizer, S. M., & Frahm, J. M. (2019). Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5555-5564).

[58] Pilzer, A., Xu, D., Puscas, M., Ricci, E., & Sebe, N. (2018, September). Unsupervised adversarial depth estimation using cycled generative networks. In *2018 international conference on 3D vision (3DV)* (pp. 587-595). IEEE.

[59] Jung, H., Kim, Y., Min, D., Oh, C., & Sohn, K. (2017, September). Depth prediction from a single image with conditional adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)* (pp. 1717-1721). IEEE.

[60] CS Kumar, A., Bhandarkar, S. M., & Prasad, M. (2018). Monocular depth prediction using generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 300-308).

[61] Gwn Lore, K., Reddy, K., Giering, M., & Bernal, E. A. (2018). Generative adversarial networks for depth map estimation from RGB video. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 1177-1185).

[62] Greene, W. N., & Roy, N. (2020, May). Metrically-scaled monocular slam using learned scale factors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 43-50). IEEE.

[63] Tateno, K., Tombari, F., Laina, I., & Navab, N. (2017). Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6243-6252).

[64] Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (1999, September). Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms* (pp. 298-372). Springer, Berlin, Heidelberg.

[65] Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. *Advances in neural information processing systems*, *28*.

[66] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, *13*(4), 600-612.

[67] Bian, J. W., Zhan, H., Wang, N., Chin, T. J., Shen, C., & Reid, I. (2021). Auto-Rectify Network for Unsupervised Indoor Depth Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[68] Abadi, M. (2016, September). TensorFlow: learning functions at scale. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming* (pp. 1-1).

[69] Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012, October). Indoor segmentation and support inference from rgbd images. In *European conference on computer vision* (pp. 746-760). Springer, Berlin, Heidelberg.

[70] Zhou, Q. Y., Park, J., & Koltun, V. (2018). Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*.

[71] Zhou, S., Zhang, J., Pan, J., Xie, H., Zuo, W., & Ren, J. (2019). Spatio-temporal filter adaptive network for video deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2482-2491).

[72] Saxena, A., Sun, M., & Ng, A. Y. (2008). Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, *31*(5), 824-840.

[73] Horaud, R., Hansard, M., Evangelidis, G., & Ménier, C. (2016). An overview of depth cameras and range scanners based on time-of-flight technologies. *Machine vision and applications*, *27*(7), 1005-1020.

[74] Pire, T., Fischer, T., Castro, G., De Cristóforis, P., Civera, J., & Berlles, J. J. (2017). S-PTAM: Stereo parallel tracking and mapping. *Robotics and Autonomous Systems*, *93*, 27-42.

[75] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision* (pp. 2564-2571). Ieee.

[76] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91-110.

[77] Bay, H., Tuytelaars, T., & Gool, L. V. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.

[78] Faugeras, O. D., & Lustman, F. (1988). Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, *2*(03), 485-508.

[79] Olson, E., & Agarwal, P. (2013). Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, *32*(7), 826-840.

[80] Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis* (pp. 105-116). Springer, Berlin, Heidelberg.

[81] Endres, F., Hess, J., Sturm, J., Cremers, D., & Burgard, W. (2013). 3-D mapping with an RGB-D camera. *IEEE transactions on robotics*, *30*(1), 177-187.

[82] Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012, October). A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 573-580). IEEE.

[83] Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *13*(04), 376-380.

[84] Azuma, R. T. (1997). A survey of augmented reality. *Presence: teleoperators & virtual environments*, *6*(4), 355-385.

[85] Billinghurst, M., Clark, A., & Lee, G. (2015). A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, *8*(2-3), 73-272.

[86] Marchand, E., Uchiyama, H., & Spindler, F. (2015). Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, *22*(12), 2633-2651.