Single View Image 3D Geometry Estimation Using Self-Supervised Machine Learning

Hang Zhou

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University of East Anglia.

School of Computing Sciences

University of East Anglia

June 11, 2024

I, Hang Zhou, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Recovering 3D information from 2D RGB images is an essential task for many applications such as autonomous driving, robotics, and augmented reality, etc. Specifically, estimating depth information, which is lost during image formation, is a vital step for downstream tasks. With the development of deep learning, especially supervised learning, more and more researchers exploit this technique to improve depth estimation. However, supervised learning based models' performance heavily relies on the quality of depth ground truth which is expensive to collect. In contrast to supervised learning methods, based on well-established Structure-from-Motion, self-supervised approaches only require sequential images to train depth estimation models, which transfer a depth regression task to an image reconstruction task

In this thesis, we focus on improving self-supervised monocular depth estimation. To this end, we propose several approaches: Firstly, we explore temporal geometry consistencies across consecutive frames and propose a depth loss and a pose loss. Secondly, we adopt HRNet and attention mechanism to build a novel representation network architecture DIFFNet, which significantly benefits from higher resolution input images. Thirdly, we propose a two-stage training scheme upon the existing one-stage framework by introducing a second-stage training when a self-distillation loss is optimized at the same time as the photometric loss. All of my works have been published at conferences.

Acknowledgements

I would to thank my supervisor, Dr. Michal Mackiewicz for his advice, help, and encouragement throughout my PhD. Also many thanks to Dr. Han Gong, Dr. Sarah Taylor, and Dr. David Greenwood who were my former supervisory team. And special thanks to Dr. Han Gong for his career suggestions and help during my internship hunting. I also would like to say thank you to my collaborators Dr. Mark Fisher and Mazvydas Gudelis who can always inspire me during discussion. I gratefully acknowledge the Research and Specialist Computing Support service which provides the high performance computing cluters where all of my research was carried out.

I would like to say thank you to my colleagues Artjoms Gorpincenko, Brandon Hobley, Jake McVey who helped a lot since I stepped into the office. I still remember that every Tuesday was our football day when I had the most relaxing time in my first year. Many thanks to Yuteng Zhu, Yimeng Li, and Xiaofei Yang who helped me adapt to work and life in the UK.

I would like to thank my friends at UEA: Zhongye Chen, Shunan Fu, Haoyu Jin, Siyu Meng, and Xiaoyu Zhu for the time we spent together playing badminton, board games, and cooking. They are a constant source of joy that supports me during my final year.

Finally, I would like to thank my parents for their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

3D scene understanding has many practical applications in autonomous navigation, Augmented Reality (AR), and structure reconstruction. As a vital part of inferring the 3D geometry of a scene, depth estimation techniques have been attracting more and more attention in the last decades.

Depth perception hardware such as LiDAR sensors have been widely deployed on vehicles and personal electronic consumer products (iPhone Pro and iPad Pro). However, LiDAR devices are expensive, and the quality of generated depth is sparse and material-sensitive. To overcome LiDAR devices' limitations, some commercial LiDAR based products have been integrated with monocular or stereo camera systems to generate high-quality and high-resolution depth. Besides, camera-based perception systems are generally lighter and smaller than LiDAR systems which makes non-LiDAR systems possible to deploy on wearable products such as AR smart glasses. Another advantage is that non-LiDAR depth estimation systems typically consume less power compared to LiDAR systems, which can be crucial for battery-powered devices such as drones, autonomous vehicles, and smartphones. As a more compatible and economical solution to infer scene geometry, depth from photographs is becoming popular and attractive in academic and industrial research communities.

Camera-based depth estimation methods can be divided into monocular, stereo and multi-view according to types of input images when inferring. A summary of their strengths and weaknesses is illustrated in Table 1.1. Despite the aforemen-

	Monocular	**Stereo**	**Multi-View**
Definition	Predicts depth from a single image	Uses two or more cameras to calculate depth	Uses multiple images from different viewpoints
Strength	- Simple and cost-effetive - Requires only one camera - Versatile in different environments	- Provide accurate depth information - Real-time processing	- Comprehensive 3D modelling - Flexible use of multiple viewpoints
Weakness	- Less accurate due to lack of direct depth information - Requires large training datasets	- Requires precise calibration - Issues with occlusion and texture dependency	- Computationally intensive - Needs accurate alignment and matching

Table 1.1: Comparison of the strengths and weaknesses of monocular, stereo, and multi-view depth estimation.

tioned advantages of camera-based depth perception, monocular depth estimation methods, which can be particularly useful in applications where only one camera is available, are the most versatile as they can be integrated with other scene perception tasks more simply and seamlessly. Specifically, we exploited self-supervised learning techniques to overcome the drawback of requiring large amounts of data. Within the scope of this thesis, we developed three self-supervised based approaches to improve the performance of monocular depth estimation methods

In the past decade, supervised learning using Convolutional Neural Networks (CNNs) has been a popular topic in our computer vision research community. CNN-based supervised learning has achieved tremendous results in image recognition [12], segmentation [13], and depth estimation [14].

However, supervised training requires a significant amount of data labelled by humans or other hardware. When applying this technique to depth estimation, we need hardware such as LiDAR and Kinect sensors which need to be calibrated with cameras. Such devices incur a major cost and also introduce significant noise to ground truth due to the characteristics of such hardware.

Therefore, it is more desirable to explore the development of unsupervised learning methods. Unsupervised learning of depth can be summarised, by type of

Figure 1.1: Monocular Depth Estimation: Left: the RGB input images; Right: the corresponding outputs of our depth estimation method.

input, into two categories: stereo input [14] or monocular input. Indeed, using multi-view input images for depth estimation [15] is also possible, however, the multi-view method is usually considered less popular and a general case of stereo vision. It is often specialized to reconstruct a single object of interest rather than a frontal scene. We thus do not discuss multi-view depth estimation in detail in this thesis.

Stereo depth estimation [16], or stereo vision, is one of the most common methods for vision-based depth estimation. However, its major limitation is the difficulty of matching features between two stereo views. In some common cases where the scene texture is weak, stereo vision algorithms could easily fail (due to correspondence ambiguity). In comparison, monocular depth estimation is a more popular alternative to stereo reconstruction due to its low requirements for hardware setup. In Figure 1.1, we show some exemplar inputs and outputs from our monocular depth estimation methods, which will be further discussed in this thesis.

This said, for the monocular settings, some important constraints such as

epipolar geometry [17] (e.g. baseline length) are missing. Instead, the monocular methods [18, 4] use geometry in consecutive image sequences to build the pixel correspondence among frames. To this end, a depth and a pose neural networks are trained simultaneously. In this thesis, we concentrate on improving the accuracy of self-supervised monocular depth estimation methods in three published works.

1.1 Aims

With the goal of improving self-supervised monocular depth estimation performance, the work presented in this thesis is aimed at:

- overcoming the limitations of the original photometric loss function by introducing geometry constraints (discussed in Chapter 4).
- designing a representation learning backbone specifically optimized for depth estimation (discussed in Chapter 5).
- building a new training pipeline (discussed in Chapter 6).

1.2 Thesis Outline and Contributions

In this chapter, we have introduced our research background, motivations and aims. In the following Chapter 2, we systematically revisit prior works published in the deep learning era. We firstly review supervised learning based monocular depth estimation methods and discuss their limitations due to the quality of depth ground truth and amount of data. Then, we mainly focus on self-supervised learning based approaches categorized into two classes: stereo based depth perception and monocular depth estimation which were beneficial to each other in terms of the development of loss functions in the last decade. As for the latter one, we summarize prior works in two categories in terms of contributions: network architectures and loss functions, since our works in this thesis are also concerned with these two aspects. After the literature review chapter, we continue with three research chapters as below:

In Chapter 3 (largely based on Zhou et al. [19]): we present a new method for self-supervised monocular depth estimation. Contemporary monocular depth

estimation methods use a triplet of consecutive video frames to estimate the central depth image. We make the assumption that the ego-centric view progresses linearly in the scene, based on the kinematic and physical properties of the camera. During the training phase, we can exploit this assumption to create a depth estimation for each image in the triplet. We then apply a new geometry constraint that supports novel synthetic views, thus providing a strong supervisory signal. Our contribution is simple to implement, requires no additional trainable parameters, and produced competitive results when compared with other state-of-the-art methods at the time of publication.

In Chapter 4 (largely based on Zhou et al. [20]): based on a well-developed semantic segmentation network HRNet [5], we propose a novel depth estimation network **DIFFNet**, which can make use of semantic information in down and up sampling procedures. By applying feature fusion and an attention mechanism, our proposed method outperforms the state-of-the-art monocular depth estimation methods on the KITTI benchmark. Our method also demonstrates greater potential on higher resolution training data. Moreover, we propose an additional extended evaluation strategy by establishing a test set of challenging cases, empirically derived from the standard benchmark.

In Chapter 5 (largely based on Zhou et al. [21]): Since multi-task learning has succeeded in the supervised learning domain, we also would like to utilize this technique in our works. Besides, recent works have introduced additional learning objectives, for example semantic segmentation, into the training pipeline and have demonstrated improved performance. However, such multi-task learning frameworks require extra ground truth labels, neutralizing the most significant advantage of self-supervision. In this work, we propose **SUB-Depth**, a two-stage training framework, to overcome these limitations. Our main contribution is that we design an auxiliary self-distillation scheme and incorporate it into the standard self-supervised depth estimation (SDE) framework, to take advantage of multi-task learning without labeling cost. Then, instead of using a simple weighted sum of the multiple objectives, we employ generative task-dependent uncertainty to weight

each objective in our proposed training framework. We present extensive evaluations on KITTI to demonstrate the improvements achieved by training a range of existing networks using the proposed framework, and we achieve state-of-the-art performance on depth estimation task.

In Chapter 6, we conclude with the findings of this thesis and discuss potential improvements for future work.

1.3 Publications

The Chapter 3, 4 and 5 are largely based on the following three papers published in the conference proceedings:

1. **Hang Zhou**, David Greenwood, Sarah Taylor, and Han Gong. Constant velocity constraints for self-supervised monocular depth estimation. In European Conference on Visual Media Production (CVMP), 2020.

2. **Hang Zhou**, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In British Machine Vision Conference (BMVC), 2021.

3. **Hang Zhou**, Sarah Taylor, David Greenwood, and Michal Mackiewicz. Subdepth: Self-distillation and uncertainty boosting self-supervised monocular depth estimation. In British Machine Vision Conference (BMVC), 2022.

Chapter 2

Literature Review

In this Chapter, before we review deep learning based methods for depth estimation, we start with a recap of the classic 3D reconstruction method Structure from Motion, as it is the foundation of self-supervised monocular depth estimation.

Self-supervised depth learning can be treated as an alternative approach to supervised learning, we start with supervised-based approaches first and then cover self-supervised related works.

For a high-level overview of this Chapter, please find a diagram of how the related works are organized, shown in Figure 2.1.

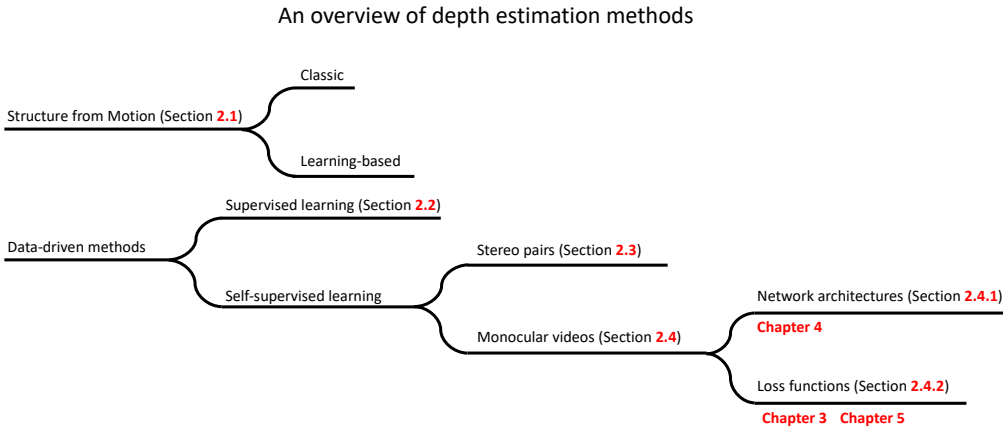


Figure 2.1: An overview of the methods discussed in this Chapter. Font indicates sections, and Chapter 3, Chapter 4 and Chapter 5 are associated with the related Sections.

2.1 Structure from Motion

Structure from Motion (SfM) is a branch of classic 3-D reconstruction algorithms which have been well exploited in computer vision. These methods simultaneously estimate the pose of the moving camera and structure (or shape) information from the views captured by a camera [16, 22, 23, 1]. However, their general framework is usually based on some strong assumptions which are often not met in the presence of occlusions, fine structures, moving objects, complex geometry, or weak texture. The majority of these methods can only estimate sparse reconstruction results, i.e. point clouds, for rigid scenes. Figure 2.2 shows an example of SfM 3-D reconstruction results.

These SfM methods, nevertheless, provide the theoretical foundation (e.g. geometric transforms and camera projection models) for us to develop deep CNN solutions which can handle more dynamic scenes and achieve dense 3-D reconstruction results. Since this PhD project focuses on learning-based approaches, we would omit the detailed discussions of traditional SfM in this thesis.

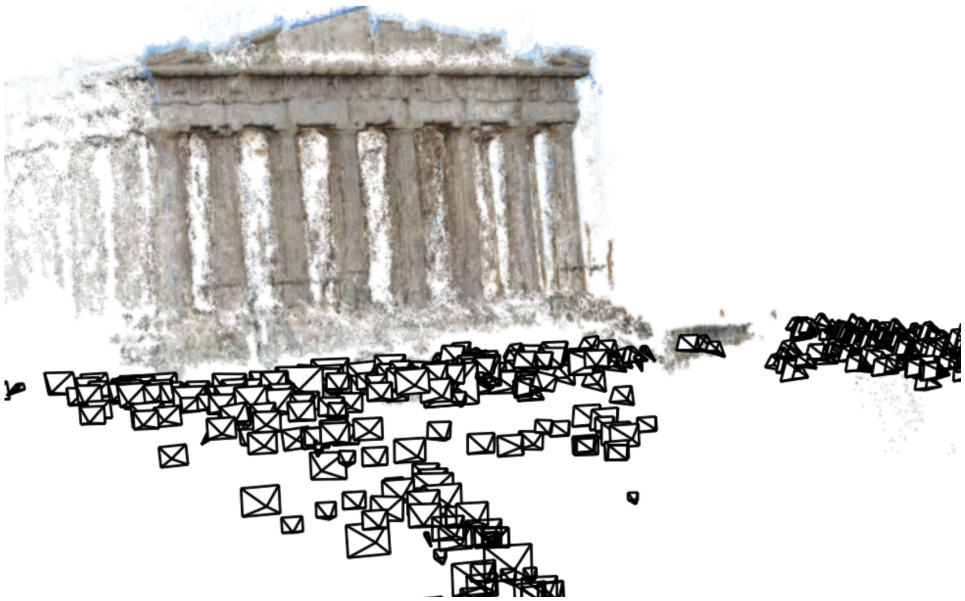


Figure 2.2: Structure from Motion (SfM): Multiple views of capturesMultiple images are taken (poses indicated by the black camera boxes) and the reconstruction SfM result is commonly a point cloud of a rigid object due to the limitation of sparse feature correspondences. This figure has been taken from [1].

More recently, learning-based SfM approaches such as Atlas [24], NeuralRecon [25] learn a neural implicit representation, Truncated Signed Distance Function (TSDF) volume, with the neural networks to reconstruct 3D scenes. As a result, such methods directly regress a form of 3D scene representation without an intermediate estimation of depth maps. So following works in this research line are out of the scope of this thesis.

2.2 Supervised depth estimation

Estimating dense depth information from only a single input image is an ill-posed problem as the input image can be projected to multiple depths – this is depth projection ambiguity. To overcome this issue, supervised learning is required to train the neural networks that map colour image input to depth in a statistically meaningful way. For instance, vanishing edges, lighting and scene context could provide important guidance for depth estimation. Supervised learning methods (esp. encoder-decoder) are expected to learn the cues of depth from a large amount of input and ground truth data. There have been a number of papers which have studied this research problem. For example, the method in [26] automatically generates plausible depth maps from videos using non-parametric depth sampling and use local motion cues to improve the inferred depth maps where optical flow is used to ensure temporal depth consistency. For training, they have adopted a Kinect-based system to collect a large dataset of stereoscopic videos with known depth. Some others [27, 28] have combined local predictions to improve depth estimation robustness. Saxena et al. [28] have adopted Markov Random Field (MRF) to infer a set of "plane parameters" that capture both the 3-D location and 3-D orientation of each small homogeneous patch called "Super-pixels" in the image. The MRF, trained by supervised learning, models both image depth cues and the relations between different regions of the image.

To tackle the inherent scale ambiguity of single-image depth estimation, Eigen et al. [29] proposed a model consisting of a global coarse-to-scale network and a local fine-scale network. The former network accounts for the overall depth

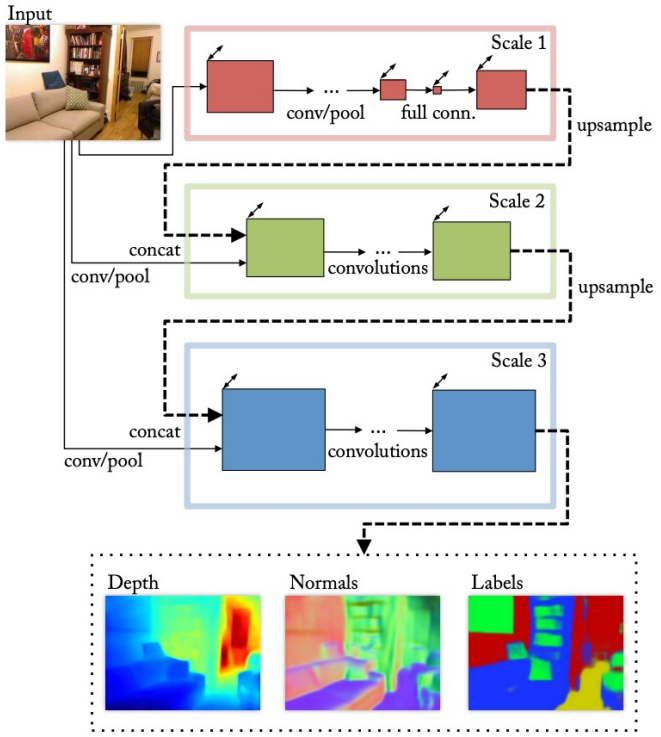


Figure 2.3: The multi-scale network architecture of Eigen and Fergus [2].

map structure prediction extracting global information from the image. The fine-scale network combines the outputs from the coarse-to-scale network and features from the original image to refine depth prediction locally. Besides the novel coarse-to-fine architecture design, the authors also proposed a scale-invariant mean squared error which measures the relationship between models' predictions and depth ground truth, regardless of the absolute global scale of ground truth. For a predicted depth map y and ground truth y^* each with n pixels indexed by i, the measurement also is used as a training loss shown in Equation 2.1.

$$L\left(y,y^*\right) = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}\left(\sum_i d_i\right)^2 \tag{2.1}$$

Where $d_i = \log y_i - \log y_i^*$ and $\lambda \in [0,1]$. When $\lambda = 1$, the training loss is exactly the scale-invariant error measurement.

Upon the two-scale design of [29], Eigen and Fergus [2] proposed a multi-scale network shown in Figure 2.3 which takes as input a sequence of three scales to gen-

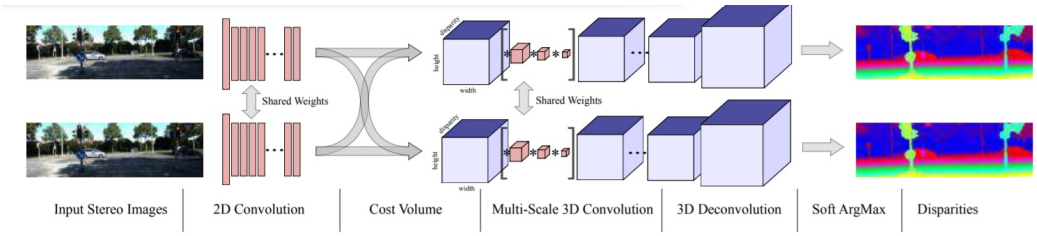


Figure 2.4: The deep stereo regression architecture.

erate features and refine predictions to higher resolution. This multi-scale network design also benefits other dense-prediction tasks like surface normal estimation and semantic segmentation.

Apart from the end-to-end training methods above, for stereo depth estimation, Kendall et al. [30] have adopted 3-D convolutions to efficiently learn the context in the disparity cost volume. An example is shown in Figure 2.4. They modelled the problem as a regression optimisation. Due to sparsity of ground truth depth map, the network is trained by using the absolute error between the ground truth depth d_n and the estimated depth map d'_n for pixel n. The supervised regression loss $Loss_{\text{avg\_depth}}$ is defined in Equation 2.2.

$$Loss_{\text{avg\_depth}} = \frac{1}{N}\sum_{n=1}^{N}|d_n - d'_n| \tag{2.2}$$

where N indicates the number of pixels in the image.

Most of the above methods are fully supervised and they require full ground truth depth maps during training. However, it is time-consuming and often impractical to collect precise depth maps in real-world scenes. This motivated works on the development of weakly-supervised methods which require weakly labelled training data, e.g. unpaired synthetic depth data [31, 32, 33, 34]. Training a depth estimation model using ideal synthetic data is a solution to most of the issues but synthetic data can also cause domain bias. It is therefore infeasible to directly apply a model trained on synthetic data in real-world scenarios.

To address this, Atapour-Abarghouei et al. [32] adopted style transfer and adversarial training to predict dense per-pixel depth from a single real-world colour image by training on a large collection of synthetic environment data. They have

adopted a Generative Adversarial Network (GAN), as it is shown in Equation 2.3. The loss function consists of two components: 1) Photometric reconstruction loss (Equation 2.4), which "rewards" the generator to produce images that are structurally and contextually similar to the ground truth; 2) Adversarial loss (Equation 2.5).

$$Loss = \lambda Loss_{\text{rec}} + (1-\lambda) Loss_{\text{adv}} \tag{2.3}$$

$$Loss_{\text{rec}} = |G(x) - y| \tag{2.4}$$

$$Loss_{\text{adv}} = \min_{G} \max_{D} f[\log D(x,y)] + f[\log(1 - D(x, G(x)))] \tag{2.5}$$

where $Loss_{\text{rec}}$ is a photometric reconstruction loss, λ is a weighting parameter, $Loss_{\text{adv}}$ is an adversarial loss. G is a generative model that learns a mapping from the input x (RGB image) to the output y (depth map). The generator G attempts to produce fake samples $G(x)$ that cannot be distinguished from real ground truth y by the discriminator D. $f()$ denotes data distribution defined by $G(x)$.

To improve the robustness and generalization capabilities of models for single-view depth estimation, Ranftl et al. [35] proposed MiDaS, a strategy of optimally mixing diverse datasets for models' training. To effectively utilize the data from different sources, they first proposed a scale- and shift-invariant loss function to cope with the depth ranges and scales between datasets. When mixing datasets, instead of naively sampling images from each dataset equally, they adapted an approach [36] for Pareto-optimal multi-task learning to define learning on each dataset as an independent task, and the model parameters shared across datasets are optimized to find an approximate Pareto optimum via minimizing the multi-objective criterion 2.6:

$$\min_{\theta} \left(\mathscr{L}_1(\theta), \ldots, \mathscr{L}_L(\theta)\right)^{\top} \tag{2.6}$$

Where θ denotes trainable parameters and L denotes the number of datasets. $\mathscr{L}_i, i \in [1, L]$ present the proposed scale- and shift-invariant function optimized on each dataset.

Some other methods have also been proposed, e.g. supervised appearance

matching terms [37, 38] and sparse ordinal depths [39, 40]. These methods still require the collection of additional depth ground truth and sometimes the other types of annotations. Using synthetic data as training data is a temporary alternative [41] which can be limited by the generator's capacity to generate a rich set of synthetic data containing various real-world scenes and optical phenomena, e.g. motion blurs or lens glare.

Another way to generate supervisory signals is to adopt conventional structure from motion to generate sparse "ground truth" for both depth maps and camera poses [42, 43], where SfM is typically performed as a pre-processing step separated from training.

2.3 Self-supervised depth-from-stereo

While learning-based approaches have been developing for many years, the main bottlenecks are resulting from the cost of high-quality ground truth and the limited amount of data. As humans can infer depth from our binocular and motion to navigate environments without any direct depth clues, researchers intuitively explore self-supervised depth estimation with stereo pairs.

Inspired by ideas in stereo vision geometry, Xie et al. [44] firstly proposed Deep3D, a CNN trained directly on stereo pairs extracted from 3D movies. For each stereo pair, the model predicts a probabilistic disparity-like map from a 2D image on the left view as an intermediate output. Then, a differentiable selection layer combines the generated disparity map with the input left-view image to render a novel image for the right view. Finally, the model is trained end-to-end with an L1 loss between the ground truth right-view image and the rendered image. Garg et al. [14] transferred a depth prediction task to an image reconstruction task with calibrated stereo pairs and a known camera baseline. A depth map associated with a left view from a depth CNN was used to backward warp a corresponding right view. The supervisory signal was built on the pixel differences between the left view and the warped view. In addition to an L1 photometric loss, an L2 depth smoothness was used deal with the aperture problem. At that time, it achieved comparable per-

formance to supervised methods. As it suffered from artifacts, Monodepth [45] proposed by Godard et al. let a depth network generate disparities for both views (e.g. only feeding a left view but outputting two depth maps for left and right views respectively). Then, it imposed a left-right consistency regularization on depth maps from two views, which attempted to push depth maps to equal to its warped version from the other view depth map. For the photometric loss, instead of a naive L1 loss, it used a weighted combination of L1 and structural similarity loss (SSIM) [46], and an edge-aware depth smoothness regularization, which both not only significantly improve stereo depth performance, but also benefit monocular depth training approaches discussed in the next section. Given stereo depth estimation is mainly affected by artifacts in occlusion regions, it proposed a post-processing technique, which requires an additional forward of a horizontally-flipped input at testing time.

Further progress on this topic was achieved by Poggi et al. [47] who proposed Three-view Network to extend [45] using a novel trinocular assumption. With the poor availability of trinocular imagery datasets, this protocol was still trained with popular binocular stereo datasets. The depth network consists of a shared encoder and two separate decoders. When training, the shared encoder first takes as input either the right or the left view as the middle view and then feeds it to the corresponding decoder. In every iteration, each decoder generates a pair of depth maps for the 'middle' view and a side view, so the depth model can output two pairs of depth maps in total. What makes it outperform [45] is the introduction of additional geometry constraints via this assumption.

Pillai et al. [48] proposed SuperDepth to get high-fidelity depth maps by replacing the deconvolution layer [45] with sub-pixel convolutional layer [49] in the decoder. To better deal with cross-view occlusions, it incorporated a differentiable flip-augmentation layer and an occlusion regularization loss.

Gonzalez and Kim [50] experimented with several depth discretization techniques and proposed an exponential disparity discretization probability volume. Apart from this contribution, they also defined a two-stage training strategy, which first trains a model using a photometric loss and finetunes the model with an

occlusion-free photometric loss enabled by the proposed Mirrored Occlusion Module. An extended version [51] was proposed by the same team where positional encoding and a proxy depth regression loss were introduced.

Watson et al. [52] proposed DepthHint where an auxiliary supervision was introduced into the unsupervised stereo paradigm. They use an off-the-shelf traditional stereo matching algorithm, Semi-Global Matching [53], to generate depth hint maps on the fly. At training time, besides a photometric loss, an additional penalty minimizing the differences between depth hints and networks' own depth estimates is applied to pixels where hint depth maps provide a lower photometric loss than that of the latter. A concurrent work published by Tosi et al. [54] proposed a similar idea which also used SGM to generate proxy depth annotations and let networks' predictions regress using a reverse Huber loss [55] with $\alpha = 0.2$. In contrast to [52], it used a left-right disparity consistency and a manually set threshold to select reliable proxy labels.

As performance on objects' boundaries is a main effect factor on depth model evaluations, Zhu et al. [56] introduced an explicit constraint from semantic segmentation to depth estimation, which regularizes the depth border to be consistent with the border generated by a well-trained segmentation model. A recent work EPCDepth proposed by Peng et al . [57] proposed a self-distillation loss and enabled a novel full-scale depth network architecture. In contrast to prior encoder-decoder based networks, its encoder output depth maps as well as the decoder, which can enhance the encoder's geometry-specific representation learning ability. The self-distillation label was generated by selecting depth values between two same-scale depth maps from the encoder and the decoder according to the lower photometric loss. A log-based regression loss was used to minimize the discrepancy between the self-distillation annotation and the depth maps at each scale.

2.4 Self-supervised depth-from-mono

To circumvent the cost of pixel-wise annotation and stereo pairs, more and more researchers have been attracted to self-supervised monocular depth estimation. In-

spired by a traditional multi-view geometry algorithm, structure-from-motion.

Zhou et al. [18] proposed a paradigm for monocular self-supervised depth estimation, which all the following works are built upon. Given the assumptions of 'moving camera' and 'static scene', a depth network and a camera pose network are jointly trained for a novel view synthesis task with unstructured video frames. To match a target frame, a source view is warped by the target's depth map and a relative camera pose change from the depth network and pose network correspondingly, and then the differences, in a form of L1 loss, between the warped frame and the target view are used to optimize the two networks. The whole system is built on an insight that the view synthesis task can be performed well only when a depth network and a pose network can output accurate depth maps and ego-motion estimations. Since this training framework is built on two strong aforementioned assumptions, which are very likely to be violated, e.g. moving vehicles, pedestrians, and cyclists, it also models these limitations using a so-called explainability mask to softly filter out those violating pixels among each matching pair. Although this work did not archive the performance of the supervised methods, it showed promising results considering absence of ground truth depth supervision.

After this groundbreaking work, researchers have been continuously improving performance on two main aspects: 1) neural network architectures; 2) loss functions. Consequently, I will review related prior works from these two perspectives. For works containing contributions in more than one aspect, they will be multi-revisited in the following two subsections.

2.4.1 Network architectures

Every task in computer vision has been benefiting from the development of neural network architectures, for instance, from commonly used convolution networks VGGNet [58] and ResNet [59] to recently popular vision transformer architectures ViT [60] and Swin-transformer [61]. Inspired by networks originally designed for other tasks such as image recognition and semantic segmentation, many researchers have put efforts into novel architecture design for depth estimation task specifically. In this section, we review works to advance depth performance mainly in the aspect

of novel architectures.

Since latecomers choose as their baseline Godard et al. [4] Monodepth2, our review starts with this work. Monodepth2 achieved a significant improvement compared to Zhou et al. [18] by replacing the adapted DispNet [62] with a ResNet-18 and other contributions, e.g. auto-masking, full-scale loss.

Since standard convolution models use stride and pooling to increase perception field at the cost of irreversible information loss, which particularly harms depth estimation on objects' edges, Guizilini et al. [63] proposed PackNet where spatial details can be preserved during downsampling and upsampling via symmetrical packing and unpacking blocks. To account for downsampling, the packing block first folds representations at spatial dimensions to expand channel dimension using Space2Depth operation [49]. Then a 3D convolution layer is applied to the Space2Depth features aiming to expand the structured representation.

Finally, a 2D convolution layer maps the reshaped intermediate to a desired channel size. This block allows more parameters to extract spatial details benefiting feature upsampling during depth decoding. In contrast to commonly used bilinear feature upsampling, the unpacking block increases features' channel number via a 2D convolutional layer, and then decompresses packed spatial features through a 3D convolutional layer. Lastly, a feature map with appropriate dimensions is generated by a reshaping and Depth2Space operation [49].

With an assumption that the context for a pixel's depth estimation may not be at a contiguous local area, Johnston and Carneiro [64] introduced a self-attention context module [65]. It takes as input the lowest resolution features from a ResNet encoder and then generates an attention map for following depth decoding. In addition, discrete disparity volume blocks, a technique previously commonly used in depth-from-stereo and supervised depth estimation, are integrated into the depth decoder to generate multi-scale depth maps. These designs contribute to sharper results on thinner structures where the model normally infers depth from non-continuous regions.

To solve the problem that Monodepth2 [4] benefits marginally from higher res-

olution inputs (for quantitative improvement comparison, please refer to Table 4.2), Lyu et al. [6] proposed HR-Depth consisting of a redesigned skip-connection and an effective feature fusion module, inspired by Hu et al. [66], in the depth decoder.

Yan et al. [67] proposed CADepth-Net where two channel-wise attention modules were employed. To gain richer context representation, a structure perception module is placed after the last convolution layer of a depth encoder. Differing from Jhonston and Carneiro [64], this module computes global dependencies along the channel dimension. Then, a details emphasis module is integrated into the depth decoder to aggregate discriminative features via channel-wise reweighting.

To make the most use of semantic information, upon HRNet [5], we proposed DIFFNet in Chapter 4, which is enhanced by a principled strategy of attention-based internal feature fusion. It first replaced a ResNet-based encoder with a modified HRNet encoder which concatenates the same resolution features across all intermediate encoding stages. In the decoder, a channel-wise attention module from Hu et al. [66] was applied to the concatenated feature maps from different scales.

Inspired by DIFFNet [20], He et al. [68] proposed RA-Depth capable of aggregating multi-scale features with dense interactions via a naive HRNet [5] encoder and the proposed high-resolution decoder. Differing from depth decoders used in the aforementioned works, RA-Depth first introduced a multi-path feature fusion design into a depth decoder to form a dual HRNet.

Instead of fusing features across an encoder and a decoder with skip-connection. Hui [69] proposed a recurrent modulation unit to refine the fusion by adaptive modulating of the encoder features using the hidden state of the decoder. To break down the static scene assumption, this work integrated a 3D motion field estimation module in the camera pose network such that moving objects like moving vehicles motion can be modeled along with the camera motion.

Since Transformer-based and CNN-Transformer hybrid vision models are becoming dominant in other computer vision communities recently, more attention has been attracted to advancing self-supervised depth estimation by utilizing Transformer-based and CNN-Transformer backbones. In contrast to CNNs' limited

receptive fields, Transformers' inherent capability of encoding long-range relationships between pixels is a natural advantage, thanks to the self-attention mechanism.

Inspired by Lee et al. [70] MPViT, Zhao et al. [71] extended the Multi-Path Transformer Block in [70] to Joint CNN and Transformer Layer by introducing an additional CNN block. The proposed design can benefit from CNN's capability of local information modeling which is better than that of Transformers.

Bae et al. [72] compared CNNs and Transformers in terms of generalization abilities of depth estimation. The experiments illustrated that CNNs demonstrate a strong texture bias whereas Transformers show smaller texture bias which benefits depth estimation on unseen data.

With these observations, they proposed Monoformer, a CNN-Transformer hybrid network, by designing a module measuring the importance of global semantic representations and the local details. As a result, in the depth encoder, each stage generates not only a feature map but also a position attention map and a channel attention map respectively. In the depth decoder, a feature fusion module was proposed to automatically determine the importance between these two attentions using two learnable parameters. In the reported ablation study, the method outperformed the naive ViT model while it showed the best generalization.

As computation complexity is a well-known drawback of Transformers compared to CNNs, Zhang et al. [73] proposed Lite-Mono using channel-wise attention instead of spatial-wise attention which has a linear time complexity to input dimension. The proposed attention was adopted from Ali et al. [74].

Apart from the above single-frame depth estimation architectures, many works exploited spatial-temporal information to benefit monocular depth prediction via multi-frame inputs. Patil et al. [75] first introduced an RNN-based depth network to extract spatial-temporal information across consecutive frames. Wang et al. [76] proposed a module to connect the depth network and the pose network which takes two frames as input such that the depth model can extract implicit cues from nearby frames. Either applying RNNs to depth networks or sharing intermediate features from pose networks outperforms single-frame depth estimators. However, they both

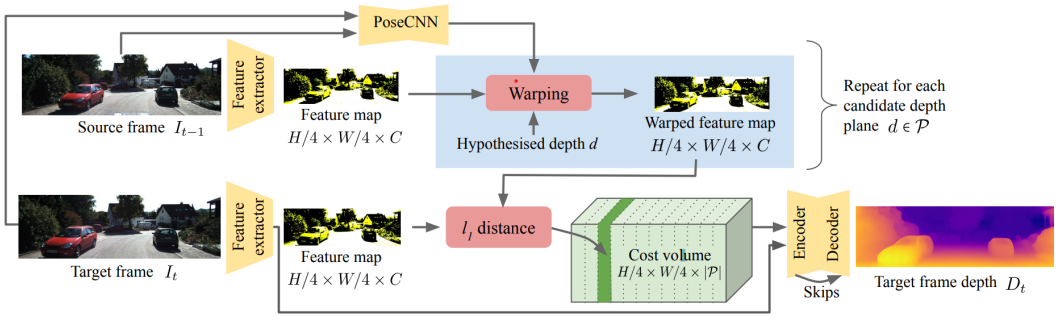


Figure 2.5: The cost volume based depth network from Watson [3].

rely on the depth networks' abilities for learning temporal-geometric features in latent spaces.

With the nature of the image synthesis task, warping a source frame to match a target frame, cost volume is a commonly used technique in multi-view stereo. As a result, a recently emerging branch of monocular depth in a self-supervised fashion is to explicitly utilize sequentially geometric information via feature matching across frames.

Based on the baseline [4], Watson et al. [3] proposed Manydepth consisting of a cost volume based encoder. When training and testing, a source frame and a target frame are fed into a shared feature extractor, then the generated source feature map is warped to match the generated target features using an estimated pose and preset depth candidates. For each depth candidate, a 2D cost map is computed by taking per point the sum of absolute differences between the target feature map and a depth-aware warped source feature map. Then such 2D cost maps are stacked to build a cost volume, as shown in Figure 2.5. The rest of the proposed network is the same as Godard et al. [4]. Thanks to multi-frame input, the depth encoder can learn cross-frame geometric features in addition to appearance based features, which is a vital advantage of multi-frame depth models to single-frame ones.

With the observation of hand-crafted matching metric, e.g. the sum of absolute differences in [3], leading to local minima, Guizilini et al. [77] proposed an attention based matching method to build the cost volume. It used an multi-head attention module proposed by Vaswani et al. [78] to develop a cross-attention matching module that projects the target features to queries and computes keys and values from the

matching features warped from source frames when generating attention maps. For each depth candidate, the attention values are obtained by taking the mean over all of the heads. This module can be executed many times, and each time the matching features are updated by the output values. Following each cross-attention module except the last layer, a self-attention module refines those three elements by computing queries from matching features instead. Finally, an attention based volume is generated to encode the similarity between features.

Ruhkamp et al. [79] developed a spatial-temporal attention architecture taking consecutive triplets as input and outputting three corresponding depth maps. For each frame's feature map, a spatial attention map is modeled explicitly by measuring the 3D distance between two points. A temporal attention layer iteratively selects a feature map as a query and the other two as keys and then calculates similarities for each key-query pair. By passing features through the proposed spatial attention and the temporal layer, the depth decoder can receive spatial-temporally aggregated features for temporal input frames.

2.4.2 Loss functions

Following Zhou et al. [18], Klodt and Vedaldi [80] introduced an off-the-shelf SfM approach to generate auxiliary supervisory signals. To deal with the noisy signal generated by traditional SfM, they proposed a probabilistic formulation that can let models learn where signals are reliable. For pixels violating assumptions, they also modeled the probability per pixel to let the network learn to down weight losses on the corresponding pixels.

To better model the photometric uncertainty, Yang et al. [81] proposed a learned brightness estimation module that aligns a source frame's lighting condition to that of the target frame. This approach let depth models get rid of the negative effect of lighting conditions changing when modeling photometric uncertainty.

To compare different uncertainty modelling methods for self-supervised depth learning in a comprehensive way, Poggi et al. [10] first charted methods in literature ranging from empirical uncertainty estimation to predictive uncertainty modeling. Based on the evaluations of different approaches, they proposed a teacher-student

paradigm in which a student depth network is trained with the pseudo annotations generated from a well-trained teacher depth network while a generative uncertainty is learned within the student model. Inspired by the teacher-student training scheme, we proposed a two-stage training framework discussed later in Chapter 5. The first stage is the same as that of [4]. What makes it different from [10] is that in the second stage, a student network is trained with a multi-objective loss function consisting of the photometric loss and an L1 loss. Furthermore, we also introduce uncertainty modelling to automatically weight the two objectives during the second stage of training.

To solve the inherent problem of depth scale ambiguity, Bian et al. [82] imposed a loss on the inconsistency of depth estimates between consecutive views and a self-discovered mask to detect pixels belonging to moving objects. However, introducing such geometric consistency supervision harms the estimation accuracy on depth discontinuity regions.

To resolve this problem, Ruhkamp et al. [79] proposed a cycle-consistency masking scheme and an occlusion-aware geometric loss. To deliver more accurate relative depth structures, Wang et al. [83] designed a two-stream depth network to disentangle depth and scale predictions along with a scale-aware geometric loss which enforces depth consistency and provides supervision for scale learning.

For explicitly learning input-scale invariant depth, He et al. [68] designed a data augmentation technique that generates training samples by randomly cropping original images to arbitrary scales and a cross-scale depth consistency over depth maps from different scale inputs in a scene. This approach can also benefit models trained with fixed-resolution images to generalize well on a higher resolution input.

Since the accuracy of depth models heavily relies on the image synthesis quality measurement, Godard et al.[45] first introduced SSIM [46] into self-supervised stereo depth estimation. As Monodepth [45]'s successor, Monodepth2 [4] inherited it to improve the naive L1-based appearance matching function used by Zhou et al. [18], in a form of the weighted sum of SSIM and L1 loss. To alleviate the effects of contaminated regions for this image warping task, where moving objects, static

cameras and resulting occlusions occur, unlike Zhou et al. [18]'s learned explainability mask, Godard et al. [4] proposed a simple yet effective per-pixel minimum reprojection strategy and an auto-masking operation to reduce occlusions and filter out static pixels respectively. Furthermore, they proposed a full-resolution multi-scale method that upsampled different intermediate depth maps to the input size when calculating photometric loss at each scale. In Chapter 3, built upon Monodepth2, we propose a depth-pose consistency loss by explicitly imposing a linear motion hypothesis on the camera ego-motion. The main idea of our proposed method is that distance from the camera to any static scene instance (e.g. roads) varies linearly in a short time interval.

As pixels across frames in textureless regions (e.g. the sky, road) are less discriminative to the photometric loss, Shu et al. [84] proposed an auto-encoder network to learn discriminative representations for each frame. During training, the photometric loss takes the generated features as input in addition to the original target-source pair. With the observation that the largest depth maps are not always accurate over all pixels, Peng et al. [57] designed a self-selective supervision signal that distills the best depth value for each pixel by comparing the minimum photometric loss computed across depth maps in different scales. The depth map sampled from the outputs can be treated as 'ground truth' to build a regression loss.

In addition to the 2D based photometric loss, Mahjourian et al. [85] proposed a 3D point cloud alignment loss to enforce a geometric consistency of inferred point clouds and relative pose changes across a consecutive triplet. The proposed loss used a traditional rigid registration, Iterative Closest Point (ICP), to compute a transformation and a residual registration error that is treated as the negative gradient with respect to pose and depth estimates respectively.

Some works introduced other tasks which are highly correlated with depth estimation in order to utilize the underlying temporal-spatial geometric cues. Those vision tasks also need to establish dense mappings between pixels on nearby frames in a self-supervised manner such that depth networks can benefit from multi-task learning and additional geometry consistency without the additional cost of annota-

tions.

Intuitively, self-supervised optic flow estimation is an auxiliary task meeting the above requirements. Zou et al. [86] introduced a separate optic flow network to form a system, DF-net, that learns depth, camera motion and optic flow jointly. The main idea is that for rigid regions it synthesizes the rigid optic flow using estimated depth and pose and puts a cross-task consistency penalty on the discrepancy between the synthesized flow and the estimated flow from the optic flow model. To detect the rigid regions, it employed forward-backward consistency check [87] on the synthesized rigid flow.

At the same time, Yin and Shi [88] proposed Geonet which reasons pixel correspondence for static and dynamic components in the scenes separately. Differing from Zou et al. [86] who used a separate flow network, it introduced a residual flow model to capture the residual flow for non-rigid regions upon the rigid flow synthesized by depth and camera pose predictions. Chen et al. [89] proposed GLnet where an adaptive photometric loss is designed. This loss term took into account the per-pixel minimum error between a synthesized image warp by the depth-pose outputs and the image generated by estimated optic flow. Furthermore, in contrast to [88] and [86], it does not synthesize a flow from depth and pose to establish an optic flow consistency with the corresponding output of a flow estimator, but, it directly enforces a global epipolar constraint over the dense correspondences from optical flow prediction.

To improve depth-pose-flow consistency, Ranjan et al. [90] introduced a motion segmentation model to distinguish static backgrounds and moving objects in scenes. In addition to 2D optic flow, Hur and Roth [91] first incorporated 3D scene flow estimation with depth in a self-supervised fashion. To this end, they modified the decoder of Sun et al. [92] PWC-Net to output scene flow and depth simultaneously. When warping the feature map, they project the scene flow to the optic flow using the corresponding depth.

Apart from depth-flow consistency, Yang et al. [93] introduced a surface normal consistency constraint based on the idea that predicted depths should be com-

patible with the surface normal computed from depth estimates. To this end, they constructed a depth-to-normal layer to compute normal directions and a normal-to-depth layer to recover depth maps from synthesized surface normal. Built upon this depth-normal consistency, Yang et al. [94] introduced an edge estimation task by assuming that for those pixels without edges in-between, their reprojected point clouds should be on a planar surface.

2.5 Conclusion

In this Chapter, we have presented a literature review on data-driven based depth estimation methods. We have summarised prior works in two categories: supervised based and self-supervised based methods. Furthermore, for self-supervised monocular depth estimation, we discussed the current research trends on two branches: network architectures and loss functions. Within this framework, we will discuss our works in the following Chapter 3-5, where we proposed our approaches via novel loss functions and learning backbones.

Chapter 3

Temporal Geometry Consistencies for Self-Supervised Monocular Depth Estimation

In this chapter, with the observation on the KITTI [7] that most moving cars have similar velocities with the camera-mounted car capturing data, we proposed a depth-consistency constraint on generated depth maps from sequential images. Besides, we present a relative camera pose change consistency loss to exploit temporal geometry consistencies concerning the camera-mounted car motion.

Similar to the photometric loss, these loss terms are introduced with no additional annotation costs. Both losses are built on the assumption that the ego-centric view progresses linearly in the scene. Trained with the combination of proposed losses and photometric loss, our depth and pose models both show noticeable improvements. In summary, our contributions are:

- We propose the notion of velocity consistency for monocular depth estimation.
- We investigate a relative pose constraint across video frames captured in a short period.
- We describe an innovative training framework in which a depth CNN predicts the depth from three consecutive frames of input. We exploit relative depth

across these frames and, through a simple motion model, we construct a novel geometry constraint as a supplementary supervisory signal.

- Our method was published at CVMP2020 and yielded state-of-the-art monocular depth estimation and pose estimation results on the KITTI benchmarks at the time of publication Zhou et al. [19]. The code is available at `https://github.com/brandleyzhou/monocular_depth`

The proposed system's overview is shown in Figure 3.1.

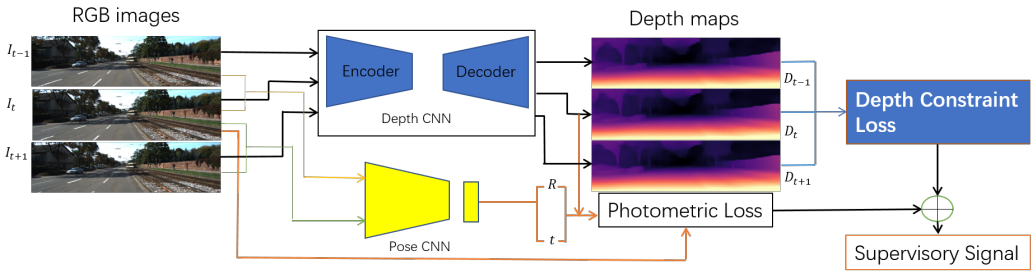


Figure 3.1: An overview of our method when training. A depth CNN and a pose CNN take a sequence of three consecutive video frames as input I_{t-1}, I_t, I_{t+1}. The depth CNN computes corresponding depth maps D_{t-1}, D_t, D_{t+1} and simultaneously the pose CNN outputs the rotation R and translation t of the camera. D_t, T and R are used to synthesise a new view and a photo-consistency loss is computed with the input image I_t (orange lines). Our main contribution is a velocity constraint loss which is computed over D_{t-1}, D_t, D_{t+1} (blue lines). To mentor training of the networks, a novel supervisory signal is constructed by combining the photo-consistency and depth-pose constraint loss.

3.1 Methods

In this section, we describe the framework of our model training and describe how we build the supervisory signals during the training of our models. Fundamentally, our method is a form of Structure from Motion (SfM), where the monocular camera is moving within a rigid environment to provide multiple views of that scene. Our framework is built upon Zhou et al. [18] and Monodepth2 [4] (see Section 2.3 for details).

Let $I_t \in R^{H \times W \times 3}$, $t \in \{-1, 0, 1\}$ be a frame in a monocular video sequence captured by a moving camera, where t is the frame time index. Similarly, let $D_t \in R^{H \times W}$ denote the depth map corresponding to image I_t. The camera pose changes

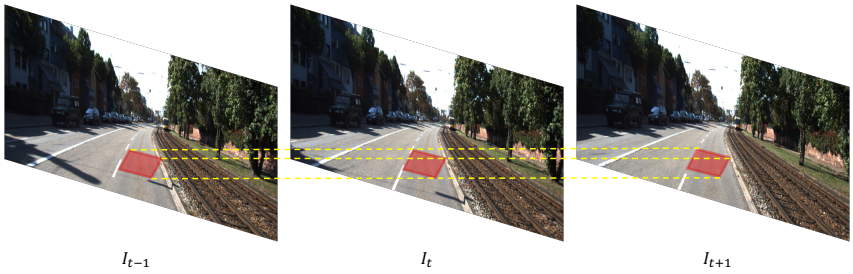


Figure 3.2: Constructing the depth constraint requires identifying the common pixels at identical regions with respect to different camera planes. These three frames I_{t-1}, I_t, I_{t+1} denote a consecutive training sample. Red boxes illustrate an example of identified common pixels. Yellow dotted lines denote spatial location mapping of the exemplar region in the three frames. In our assumption, our proposed depth constraint only takes the area of the red box into account.

from time t to time t' (t, $t' \in \{-1, 0, 1\}$, $t \neq t'$) is encoded by the 3×3 rotation matrix $\mathbf{R}_{t \to t'}$ and the 3-element translation vector $\mathbf{t}_{t \to t'}$. We obtain the 4×4 camera transformation matrix thus:

$$M_{t \to t'} = \begin{bmatrix} \mathbf{R}_{t \to t'} & \mathbf{t}_{t \to t'} \\ 0 & 1 \end{bmatrix} \tag{3.1}$$

Our aim is to train two CNN networks to simultaneously estimate the pose change of the camera motion, and the depth of the scene respectively.

$$M_{t \to t'} = \Theta_{\text{pose}}(I_t, I_{t'}) \tag{3.2}$$

$$D_t = \Theta_{\text{depth}}(I_t) \tag{3.3}$$

3.1.1 Photometric consistency as supervision

Self-supervised depth prediction reformulates the learning task as a novel view-synthesis problem [18, 4]. Specifically, during training, we let the coupled network synthesise the photo-consistency the appearance of a target frame from another viewpoint of the source frame. We treat the depth map as an intermediate variable to constrain the network to complete the image synthesis task, in which we set I_0 as a target frame and $I_{t'}$, $t^{'} \in \{-1, 1\}$ as source frames.

Let $(u,v) \in R^{H \times W}$ be the calibrated coordinates of a pixel in a target frame I_0. In this case, let the origin $(0,0)$ be the top-left of the image. In the process of imaging, a 3D point $(X,Y,Z) \in R^3$ projects onto the pixel plane at location (u,v) through a perspective projection operator. As Equation 3.4 shows:

$$\begin{aligned} proj(X,Y,Z,K) &= (f_x \frac{X}{Z} + c_x, f_y \frac{Y}{Z} + c_y) \\ &= (u,v) \end{aligned} \tag{3.4}$$

where K are known camera intrinsic containing (f_x, f_y, c_x, c_y) which denote focal lengths and the size of the photon sensor. Therefore, given a depth value $D(u,v)$, a 2D image pixel coordinate (u,v) can be reprojected to a 3D point coordinate (X,Y,Z) in the camera coordinate system through reprojection operator, Equation 3.5.

$$\begin{aligned} reproj(u,v,D(u,v),K) &= D(u,v)(\frac{u-c_x}{f_x}, \frac{v-c_y}{f_y}, 1) \\ &= (X,Y,Z) \end{aligned} \tag{3.5}$$

Suppose that the transformation matrix $M_{0 \to t'}$ correctly encodes the pose change of the camera from time 0 to time t', we can project a pixel on the target frame I_0 onto a corresponding location on source frame $I_{t'}$. e.g. given a pixel coordinate (u,v), the corresponding pixel's coordinate (u',v') can be computed as:

$$(u',v') = proj(M_{0 \to t'} reproj(u,v,D_0(u,v)),K) \tag{3.6}$$

Where $proj$ is the projection operator defined in Equation 3.4. Given such correspondence between pixels across consecutive frames, we can warp a source frame I'_t to match the corresponding target frame I_t and therefore construct a photometric-consistency supervision signal using the discrepancies between the warped image and the target frame.

3.1.2 Differential samplers

As Equation 3.6 is an ideal mathematical model outputting continual values for (u', v'), for sampling pixel values from the source frame I'_t, we include a differentiable bilinear sampling mechanism, as proposed in spatial transformer networks [95]. We can now linearly interpolate the values of the 4-pixel neighbours (top-left, top-right, bottom-left, bottom-right) of $I_{t'}(u', v')$ to give the RGB intensities for the synthesised frame $I_{t'\to 0}$ as follows:

$$I_{t'\to 0}(u, v) = \sum_u \sum_v w^{uv} I_{t'}(u', v') \tag{3.7}$$

where w^{uv} is linearly proportional to the spatial proximity between (u, v) and (u', v'), and $\sum_{u,v} w^{uv} = 1$. We use the official *grid_sample* function in PyTorch [96] for sampling pixel values from images.

3.1.3 Photo-consistency losses

Classic depth estimation using SfM relies on a number of assumptions which can fail in the presence of occlusions, fine structures, moving objects, complex geometry, weak texture (e.g. road, sky) and non-Lambertian surfaces. To mitigate these problems our method builds a strong supervisory signal by combining a number of individual loss functions.

For monocular depth estimation, an important supervisory signal to learn geometry from unlabelled video sequences is brightness constancy, which has been adopted as an invariant constraint [18]. The constraint is based on the assumption that pixels in different video frames that correspond to the same scene point must have the same intensity in general. Existing methods [18, 4, 6] have shown that a brightness constancy constraint is sufficient (at least in common cases) to guide the learning of the depth regression network and the camera pose estimation network.

Due to brightness constancy, the RGB intensities of the two corresponding pixels, in two different frames $I_0(u, v)$ and $I_{t'\to 0}(u, v)$, should match. Therefore, we

can write the fundamental photo-consistency loss as in Equation 3.8:

$$\ell_{brightness} = \sum_{(u,v)\in\Omega} |I_{t'\to 0}(u,v) - I_0(u,v)| \tag{3.8}$$

where Ω indicates the set of all pixel coordinates in a frame with respect to the defined coordinate origin. Note, we mask the brightness loss $\ell_{brightness}$ with a stationary mask, described in Section 3.1.4. All quantities in Equation 3.8 are known except for the synthesizes frame $I_{t'\to 0}$ which is sampled from the $I_t^{'}$ through estimating D_t, $M_{t\to t'}$ by the two CNN networks as Equations 3.3 and 3.2 have shown. This basic photo-consistency loss only compares pixel intensity values. An additional constraint, Structural Similarity Index Measure has been shown to improve robustness for reconstructed images' quality measurement [46]. Given a pair of images a and b, their Structural Similarity $SSIM(a,b) \in [0,1]$ is given by:

$$SSIM(a,b) = \frac{(2\mu_a\mu_b)(\sigma_a b + \varepsilon)}{(\mu_a{}^2 + \mu_b{}^2)(\sigma_a{}^2 + \sigma_b{}^2) + \varepsilon} \tag{3.9}$$

where ε is a small constant to avoid division by zero, $\mu_a = \frac{1}{n}\sum_{i=1}^{n} a_i$ is the mean intensity of image a, $\sigma_a{}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_a)^2$ is its variance, and $\sigma_{ab} = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_a)(b_i - \mu_b)$ is the intensity correlation of the two images. Finally, our combined structural similarity and brightness loss becomes:

$$\ell_p(I_0, I_{t'\to 0}) = \alpha(1 - SSIM(I_0, I_{t'\to 0})) + (1-\alpha)\ell_{brightness}(I_0, I_{t'\to 0}) \tag{3.10}$$

where the weighting parameter α is set as 0.85 empirically [4].

3.1.4 Stationary pixel masking

Important assumptions for training are that the scene is captured by a moving camera, and the scene is static with respect to a world origin point. If any of these conditions is violated, the training performance can be detrimentally affected. Using a simple auto-masking method proposed by Godard et al. [4], we can filter the pixels that do not change appearance from one frame to the next in the video sequence. This mask allows the depth estimation network to ignore objects that move

at the same velocity as the camera and even ignore whole frames in a monocular sequence when the camera is still.

A pixel is defined as moving when the photo-consistency loss between the target view I_0 and the synthetic view $I_{t'\to 0}$ through warping the source view $I_{t'}$, is lower than the same error between the target view and source view $I_{t'}$. More formally:

$$mask^s = |I_0 - I_{t'\to 0}| < |I_{t'} - I_0| \tag{3.11}$$

The mask is binary, and no additional hyperparameter is required, as the mask can be computed in the forward pass of the network training. The pixels with almost unchanged intensities between consecutive frames often indicate no relative camera movement, an object that is relatively static to the camera, or a low texture region such as sky and roads. As such, our training method uses stationary pixel masking to only consider the photo-consistency loss contribution from the "moving" pixels.

3.1.5 Photometric loss with an edge-aware smoothness

To regularize the depth in low gradient regions, we utilize edge-aware smoothness [45]:

$$\ell_s(D_0) = |\frac{\nabla D_0}{\partial x}|e^{-|\frac{\nabla I_0}{\partial x}|} + |\frac{\nabla D_0}{\partial y}|e^{-|\frac{\nabla I_0}{\partial y}|} \tag{3.12}$$

We also employ the minimum photometric error, auto-masking and multi-scale depth loss techniques which were introduced in [4]. The final photometric loss function is defined:

$$\ell_{photometric} = min(\ell_p(I_0, I_{t'\to 0})) + \beta\ell_s(D_0), t' \in \{-1, 1\} \tag{3.13}$$

Where β is a weighting coefficient between the photometric loss $\ell_{photometric}$ and depth smoothness ℓ_s. The objective loss is averaged per pixel, pyramid scale and image batch.

3.1.6 Constant velocity constraints

In this section, we describe our main contribution, a novel loss term for training. We allow ourselves the assumption that most training frames have been captured

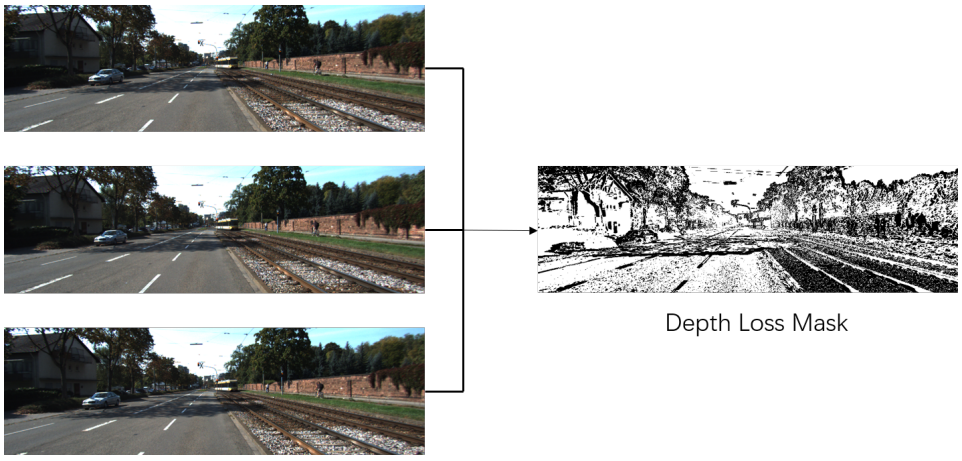


Figure 3.3: Illustration of Depth Constraint Mask: the brighter regions indicate pixels that are common to objects in all three frames.

in a short time interval, during which the velocity of the moving camera can be considered constant. Maintaining that assumption, in a set of consecutive video frames, the distance from the camera to any static scene instances in front of the camera, varies only linearly.

Suppose that we denote D_t as the depth map at some time step, we have the following equation hold for the major areas in the depth maps:

$$D_{t+1} - D_t \approx D_t - D_{t-1} \tag{3.14}$$

Our idea models relative depth changes of pixels that belong to the same instance (e.g. road) at the same locations on all three frames. We illustrate the concept in Figure 3.3. We introduce a new mask to constrain the depth loss to ensure we only consider the pixels of an instance common to all frames:

$$mask^d = [|I^g_t - I^g_{t+1}| < \beta] \cap [|I^g_t - I^g_{t-1}| < \beta] \tag{3.15}$$

where I^g is the mean luma image and β is a threshold value empirically set as 10 for 8-bit intensity values. A visualisation of an exemplar mask is shown in Figure 3.3. We apply the mask to form an additional depth loss term as follows:

$$\ell_{depth} = \lambda \mu mask^d \odot |(|D_{t+1} - D_t| - |D_t - D_{t-1}|)| \tag{3.16}$$

where the weighting parameter λ is set empirically at 0.001 from $\lambda = \{0.1, 0.01, 0.001, 0.0001\}$, μ refers to the function that computes the mean of all matrix elements, and $\odot$ is the Hadamard Product. As a result of this geometry constraint, which models the depth relation of corresponding pixels on different frames, this penalty term makes it possible for the network to estimate depth from frames that contain a lot of moving objects in the scene or even are captured by a static camera and therefore violate the photo-consistency assumptions.

Furthermore, based on this strong assumption that the camera-mounted vehicle moves at a constant speed, we also propose a constraint on the outcomes from Θ_{pose}, which puts a relative pose change consistency on transformations $\mathbf{M}$ and translations $\mathbf{t}$ among three consecutive frames as Equation 3.17 showing.

$$\ell_{pose} = \|\mathbf{M}_{t-1\to t+1} - \mathbf{M}_{t-1\to t}\mathbf{M}_{t\to t+1}\| + \|\mathbf{t}_{t-1\to t} - \mathbf{t}_{t\to t+1}\| \tag{3.17}$$

We empirically impose an additional regularization on relative translation $\mathbf{t}$. Finally, we combine the masked photo-consistency loss, depth constraint loss and pose consistency loss:

$$\ell_{total} = \ell_{photometric} + \ell_{depth} + \ell_{pose} \tag{3.18}$$

3.1.7 Model topology

Our model trains weights for two discreet networks, a depth estimation network, and a pose network. We use the depth network and the pose network of Monodepth2 [4] as our backbones for depth and pose estimation respectively. The network diagram is shown in Figure 3.4. The depth network takes as input an RGB image, and outputs the corresponding depth estimation map; the pose network takes two RGB images as input to predict the 6–DoF relative pose.

The depth network shown in (a) of Figure 3.4 follows the well-known U-Net architecture [62], It is a symmetric *encoder* and *decoder* with skip connections on every layer but the input and output. The range of spatial resolution allows mod-

elling both deep abstract features and local information. The encoder is ResNet-18 [97] with a total of 11m trainable parameters, initialized with weights trained on ImageNet [98]. Pretraining has been shown to improve accuracy compared to training from randomly initialized weights [4]. The depth decoder follows [45], with a sigmoid nonlinearality on the output, and ReLU on the internal layers. However, the convolution layers use reflection, rather than zero padding, which gives a better estimate of source image pixel values when sampling from outside the border.

The pose network shown in (b) of Figure 3.4 follows a similar design as the depth network encoder, however, it requires two frames to infer the relative camera pose change. Again, like the depth encoder, we use weights pretrained on ImageNet [98] to initialize the pose encoder. The output of the pose network is a 6–DoF relative pose in an axis-angle and translation representation.

3.1.8 Training

For monocular self-supervised training we use a sequence length of three images. To increase training data, we flip each input image horizontally and also augment brightness, contrast, saturation and hue ± 0.2 randomly. The same augmentation is applied to all three images in the input. We have implemented the networks using

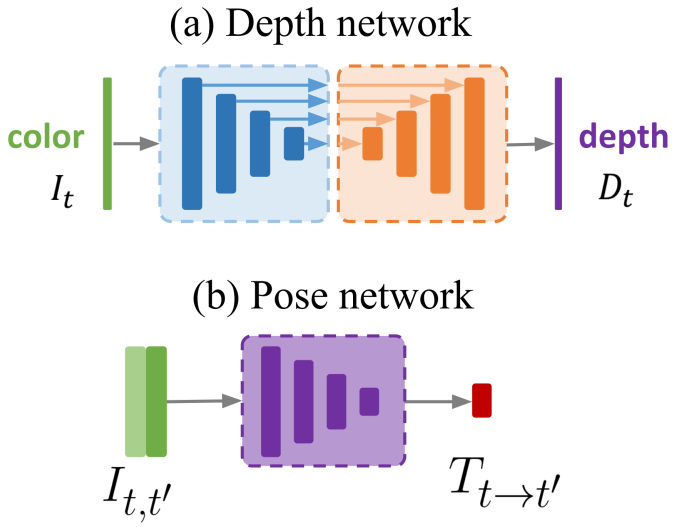


Figure 3.4: Network diagram of Monodepth2 [4], and this figure is directly from the paper. The depth network takes an RGB image I_t as input to estimate a depth map D_t, and the pose network takes two adjacent frames $I_t, I_{t'} (t' \in [t-1, t+1])$ as input to generate a relative pose change between the two frames.

Table 3.1: Definitions of Evaluation Metrics. D_p is a pixel in the ground-truth depth map d, d'_p is a pixel in the estimated depth map d', and n is the total number of pixels for each depth image.

Mean Relative Error (Abs Rel)	$\frac{1}{n}\sum_p^n \frac{\lvert D_p - d'_p \rvert}{D_p}$
Mean Relative Squared Error (Sq Rel)	$\frac{1}{n}\sum_p^n \frac{(D_p - d'_p)^2}{D_p}$
Root Mean Squared Error (RMSE)	$\sqrt{\frac{1}{n}\sum_p^n (D_p - d'_p)^2}$
Root Mean Squared Log Error (RMSE log)	$\sqrt{\frac{1}{n}\sum_p^n (log(D_p) - log(d'_p))^2}$
Threshold Accuracy (δ_i)	% of D_p, s.t. $max(\frac{D_p}{d'_p}, \frac{d'_p}{D_p}) = \delta_i < threshold_i$ $threshold_i = 1.25^i, i \in 1,2,3$

PyTorch, and they were trained using an NVIDIA Quadro P5000 GPU with 16GB memory. During training all model weights are updated simultaneously, by minimising the combined loss. The model was trained for 20 epochs, 1105 iterations every epoch using Adam [99], with a batch size of 12 and an input and output resolution of 640×192. We set the initial learning rate as 10^{-4} for the first 15 epochs and then decremented to 10^{-5} for fine-tuning the remainder. When evaluating, we only report the performance of models trained in the last epoch.

3.2 Experiments

In this section, we describe the dataset, show the evaluation metrics we use from [29] in Table 3.1, and our evaluation results in comparison with the state-of-the-art methods at the time of publication.

3.2.1 Dataset

KITTI [7] is a dataset that contains stereo images and corresponding 3-D laser scans of outdoor scenes captured by imaging equipment mounted on a moving vehicle [99]. The RGB images have a resolution of about 1241×376 and the corresponding depth maps are very sparse with a large amount of missing data. For training, we adopted the same dataset split used by [29]. After removing the static

frames by a pre-processing step suggested by [18], this results in 39,810 monocular frame triplets for training and 4,424 frame triplets for validation. To simplify the training processing, the camera intrinsic matrix are assumed identical for all the frames in different scenes. To obtain this "universal" intrinsic matrix, we offset the principal point of the camera to the image centre and reset the focal length as the average of all the focal lengths in KITTI. This assumption is only valid when the capturing cameras are similar. Indeed, a more precise solution would be required to also estimate the individual intrinsic matrices for different videos sequences.

3.2.2 Results

Table 3.2: Quantitative results on KITTI Benchmark using the Eigen split: ↑ represents the higher the better, and ↓, lower is better. The best scores in the table are underlined.

Method	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta_1 < 1.25 \uparrow$	$\delta_2 < 1.25^2 \uparrow$	$\delta_3 < 1.25^3 \uparrow$
SfMlearner [18]	0.183	1.595	6.709	0.27	0.734	0.902	0.959
Yang [93]	0.182	1.481	6.501	0.267	0.725	0.906	0.963
GeoNet [100]	0.149	1.060	5.567	0.226	0.796	0.935	0.975
Wang [101]	0.151	1.257	5.583	0.228	0.81	0.936	0.974
DF-Net [86]	0.150	1.124	5.507	0.223	0.806	0.933	0.973
LEGO [94]	0.162	1.352	6.276	0.252	-	-	-
EPC++ [102]	0.141	1.029	5.35	0.216	0.816	0.941	0.976
Struct2depth [103]	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Monodepth2 [4]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
PackNet-SfM [104]	<u>0.111</u>	<u>0.785</u>	<u>4.601</u>	<u>0.189</u>	0.878	<u>0.960</u>	<u>0.982</u>
Our method	0.112	0.816	4.715	0.190	<u>0.880</u>	<u>0.960</u>	<u>0.982</u>

In this section, we perform a quantitative evaluation to compare our proposed method with the other representative algorithms by using the common metrics discussed above.

Table 3.2 shows that our method outperforms all other methods on the KITTI 2015 dataset [7]. The exception to this is PackNet-SfM [104] which achieves marginally better performance on relative and RMSE errors, and equal or worse performance on threshold accuracy.

One of the reasons that our method produces more robust results given the same training data is that it uses a triplet of frames to supervise the training process while other approaches, such as Struct2Depth [103], rely on a pair of source and target images. Of course, this could also mean that the computational cost of training using our method would also be increased.

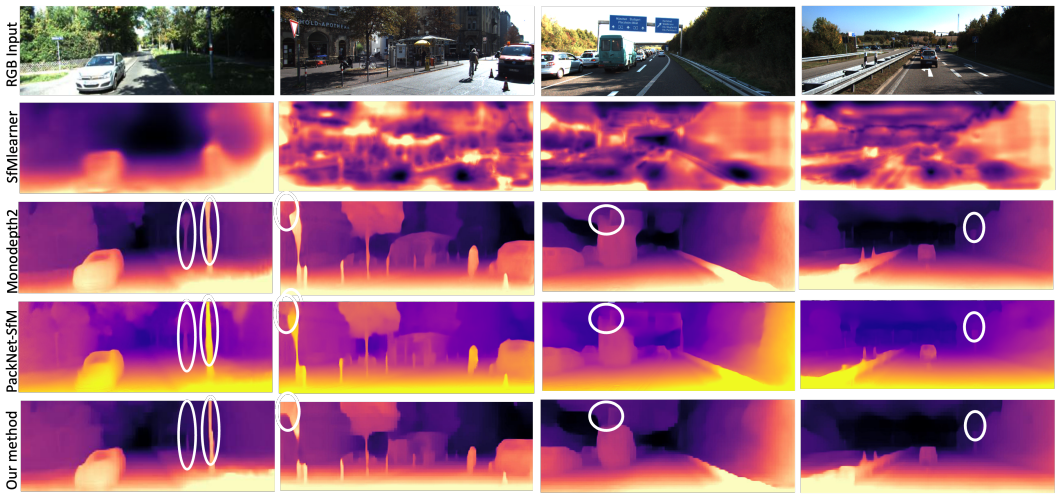


Figure 3.5: Visualisation of depth estimation results. The top row contains the input images. The remaining rows show the depth estimation results from contemporary methods, visualised by false colours. Hotter colours indicate closer objects.

Another reason is that in the KITTI dataset [7], there are many frames that are captured by a static camera that contain moving objects. These problematic frames are filtered out by the other existing methods as their training methods cannot make use of these frames. However, with our novel depth constraint loss, those frames are made useful for training.

It should be noted that our model architecture is the same as that in Monodepth2 [4]. However, training with our proposed depth constraints has resulted in improved performance overall evaluation metrics — a clear indication that our constant velocity assumptions are valid. Figure 3.5 shows the depth maps generated by SfMlearner [18], Monodepth2 [4], PackNet [104] and our method for some target frames. We observe that our method predicts fewer artefacts affected by the shadows in the scene, and more robustly identifies the contours of objects. For example, in the first column our method more accurately segments the post in the foreground, and correctly identifies that the furthest post is obscured by a tree. In the third column it is clear that our method better captures depth details around the vehicle's contour.

To better understand the behaviour of our system, we visualized the per-pixel errors of the depth map, as shown in Figure 3.6. We observe that objects that are

far from the camera have lower accuracy than those that are closer. Therefore our approach is very well suited to applications that require precise near-field depth information. As common with all contemporary works, our method suffers occasional

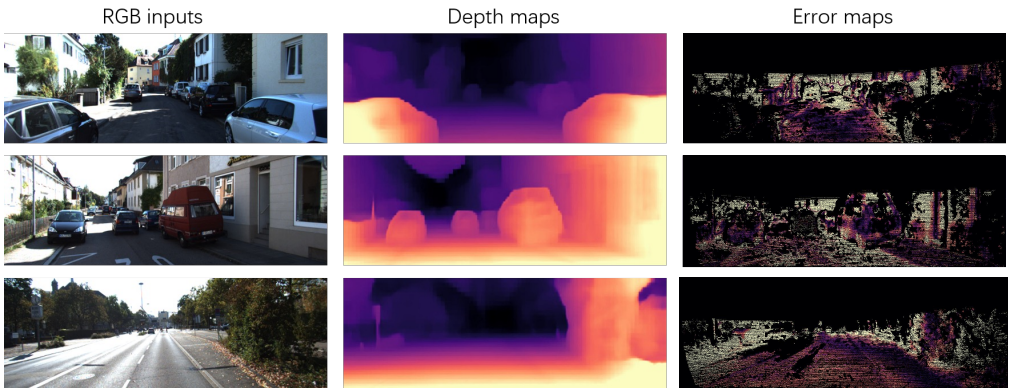


Figure 3.6: Visualisation of depth error maps. Here we show the error from our predicted depth maps compared to the improved ground truth from the KITTI test set. The first column contains the input images, the middle column shows the depth estimation and the right column shows the per-pixel depth error at pixels which have valid depth ground truth. Hotter colours indicate greater error.

failures in difficult scenes. Figure 3.7 provides some examples. We remain highly motivated to tackle these problematic areas in future work.

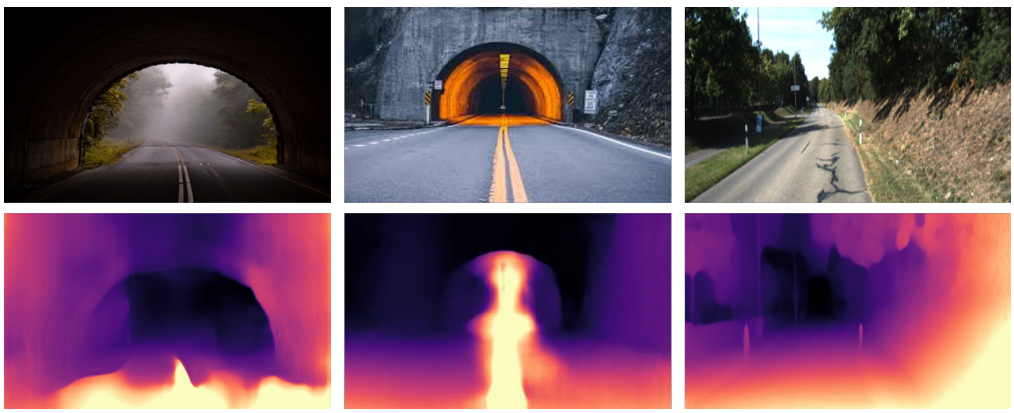


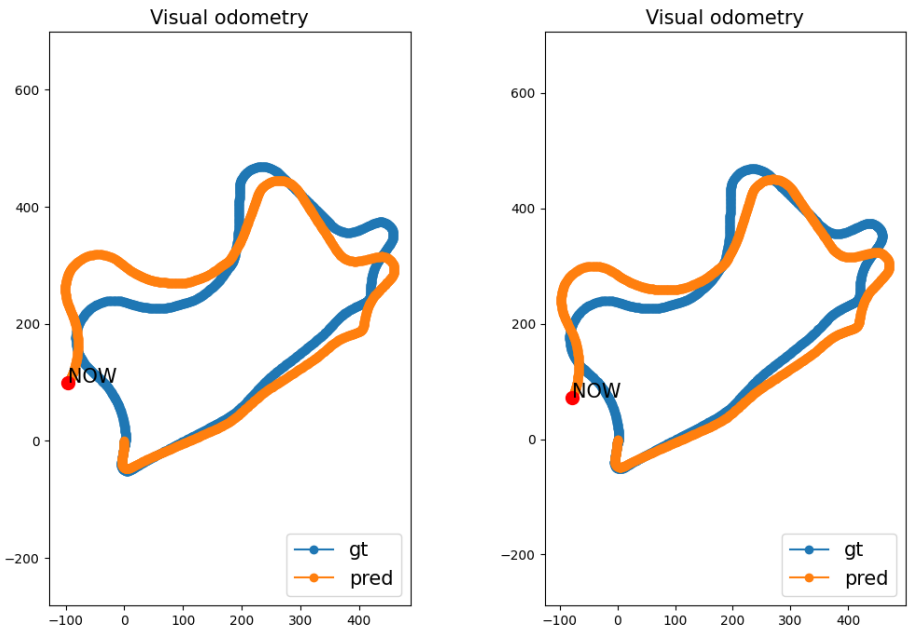
Figure 3.7: Common failure cases. Road marks have been incorrectly recognised as closer objects in the left and middle figures. The tunnel structure has been recognised as infinity (i.e. similar to Sky) in the middle figure. The sky in the right figure has been recognised as an object not at infinity. These failures exist in all contemporary methods, and motivate future work that can handle these difficult examples.

Table 3.3: Quantitative results on the KITTI odometry benchmark using average absolute trajectory error, and standard deviation, the lower the better.

	Sequence 09	Sequence 10
Zhou [18]	0.021±0.017	0.020±0.015
GeoNet [100]	0.012±0.007	0.012±0.009
DDVO [101]	0.045±0.108	0.033±0.074
Monodepth2 [4]	0.021±0.009	0.014±0.010
Ours with the pose consistency	0.020±0.009	0.012±0.010

3.2.3 Odometry

To investigate how the proposed pose consistency loss (Equation 3.17) impacts the quality of pose estimation from the pose network Θ_{pose}, we directly evaluate it on Sequences 09 and 10 from KITTI odometry split following Zhou et al. [18] evaluation method. We use Absolute Trajectory Error (ATE) to measure the performance. For comparison, we also report results from other methods. Table 3.3 shows that our proposed pose consistency loss improves the performance of the pose network. We also visualize the trajectories generated by our pose network trained with and without the pose consistency respectively on two testing sequences in Figure 3.8 and Figure 3.9.



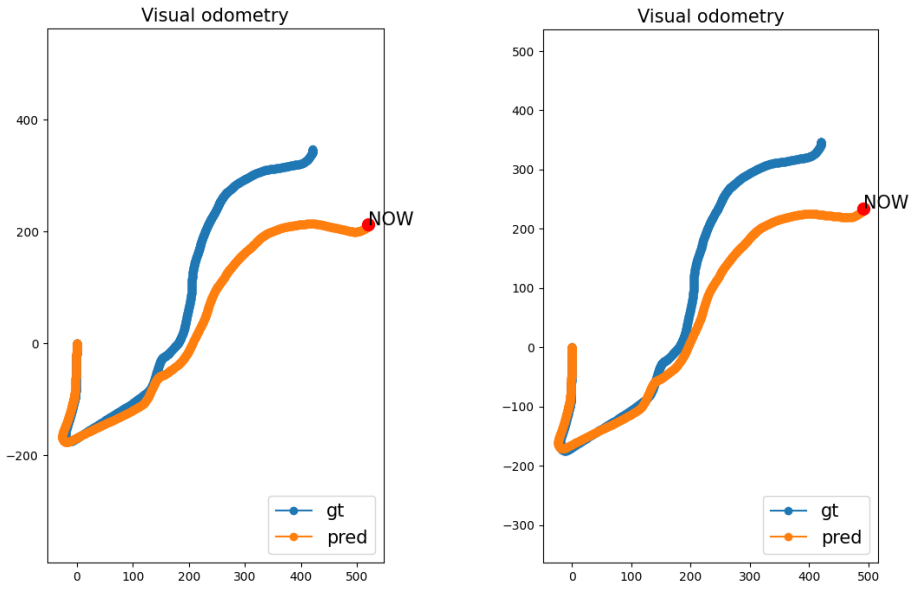
(a) Trajectories on KITTI Sequence 09, without pose consistency loss

(b) Trajectories on KITTI Sequence 09, with pose consistency loss

Figure 3.8: Visualisation of odometry results on Sequence 09.

Table 3.4: Ablation study. The first row represents the baseline, and ✓denotes an implementation option. The best scores in the table are **bold**. ✓ identify our final system.

Method	Depth loss	Pose loss	The lower the better				The higher the better		
			Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
Monodepth2 [4]			0.115	0.980	5.142	0.210	0.877	0.960	0.981
Ours	✓		0.114	0.936	5.010	0.203	0.876	0.960	0.980
		✓	0.115	0.894	4.814	0.192	0.876	0.959	0.981
	✓	✓	**0.112**	**0.816**	**4.715**	**0.190**	**0.880**	**0.960**	**0.982**



(a) Trajectories on KITTI Sequence 10, without pose consistency loss

(b) Trajectories on KITTI Sequence 10, with pose consistency loss

Figure 3.9: Visualisation of odometry results on Sequence 10.

3.2.4 Ablation Study

To understand how the components of our models incrementally contribute to the overall performance in monocular depth learning, we perform an ablation study by changing variables of our components as shown in Table 3.4.

We choose Monodepth2 [4] as our baseline shown in the first row of Table 3.4. In the second row, a marginal improvement is made by introducing our proposed depth constraint loss (Equation 3.16) to the photometric loss. In the third row, We only introduce the pose loss (Equation 3.17) to the baseline showing that the baseline model is not able to benefit from the proposed pose loss solely. In the final row, we show the biggest improvement by introducing our novel depth constraint

loss and the pose change consistency loss simultaneously.

3.3 Conclusions

In this Chapter, we have presented a novel temporal-geometry loss for monocular depth estimation and achieved state-of-the-art results on a popular benchmark. As far as we know, no prior work exploited the relationship between depth maps and relative pose changes from consecutive video frames. From a simple real-world conception, we introduce and develop additional loss items as a supplementary supervisory signal to photo-consistency loss. Our novel depth loss and pose loss are based on the assumption that the velocity of the camera moving through the scene in consecutive video frames is constant. We validate this assumption by comparing against similar approaches objectively and show depth visualisations of the competing methods. Our idea is simple to understand and implement and introduces no additional learnable parameters.

In the next Chapter, we will present a novel representation backbone optimized for depth estimation.

Chapter 4

Self-Supervised Monocular Depth Estimation with Internal Feature Fusion

In the previous Chapter, we improved the accuracy of depth models by integrating temporal-geometric constraints with the original photometric loss. In this Chapter, we will propose a new network architecture to improve the performance further.

Like many computer vision tasks, depth network performance is determined by the capability to learn accurate spatial and semantic representations from images. Most of depth estimation approaches [4, 18, 104] including our proposed method in Chapter 3 use naive U-Net [62] based architectures. However, in such an encoder-decoder architecture, low-level feature maps containing more spatial information are only able to be aggregated with high-level and semantically richer feature maps in depth decoders, which results in a huge semantic gap between the encoder and decoder feature maps. Due to semantic gaps, those methods are not able to gain significant improvements when higher resolution inputs are available [6]. To bridge semantic gaps between encoded and decoded representations, works [105, 106, 5] have designed multi-path encoders which aggregate multi-scale feature maps with dense interaction for pixel-to-pixel prediction tasks such as semantic segmentation.

In this chapter, based on a well-developed semantic segmentation network HR-Net [5], we propose a novel depth estimation network DIFFNet, which can make use

of semantic information in down and up sampling procedures. By applying feature fusion and an attention mechanism, our proposed method outperforms the state-of-the-art monocular depth estimation methods on the KITTI benchmark. Our method also demonstrates greater potential for higher resolution training data. We propose an additional extended evaluation strategy by establishing a test set of challenging cases, empirically derived from the standard benchmark.

Our contributions are:

- We apply a novel internal feature fusion mechanism to a semantic network for depth estimation, to bridge the semantic gap between encoder and decoder feature maps.

- We propose an effective attention module in the decoder to process skip connections.

- Our proposed method advances the state-of-the-art on the KITTI benchmark and outperforms other methods on a customised benchmark at the time of publication (Zhou et al. [20]). The code is available at `https://github.com/brandleyzhou/DIFFNet`

- We propose an extended evaluation strategy where methods can be further tested using difficult cases in the benchmark data, formed in a self-established manner.

4.1 Self-supervised monocular depth estimation framework

Our general framework is based on the SfM paradigm that is followed by all other self-supervised monocular depth estimation approaches e.g. [18, 4, 19]. And we use the photometric loss defined in Equation 3.13 as our objective function ℓ_{final}. Please refer to Section 3.1.5 for a detailed description.

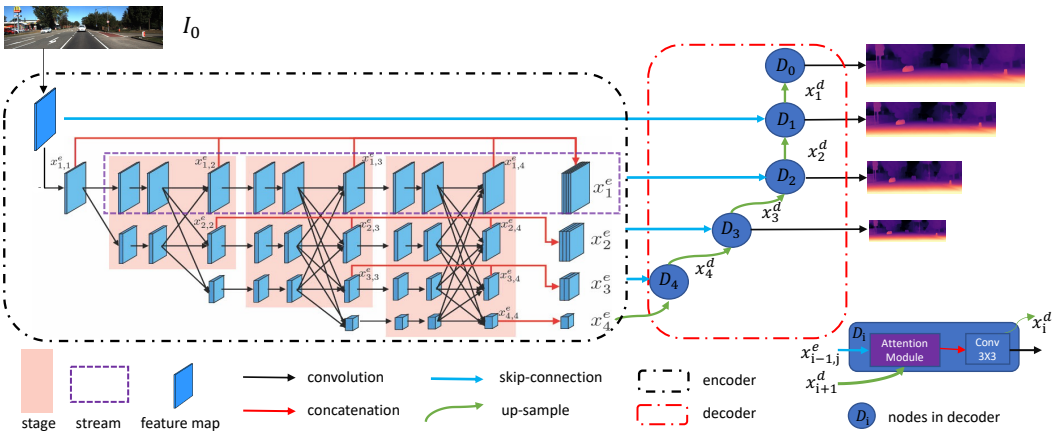


Figure 4.1: An overview of the DIFFNet depth network. The encoder uses feature fusion to generate stacks of multi-stage feature maps. For visual simplicity, we only highlight one stream in depth encoder with a purple dotted box. The decoder uses an attention module and a 3×3 convolution layer to restore compressed feature maps at different scales.

4.2 DIFFNet

DIFFNet introduces a novel depth network which combines multiple resolution feature fusion and a spatial attention mechanism. In this section we provide details on our proposed network, which is built on an encoder-decoder architecture and is illustrated in Figure 4.1.

4.2.1 High-resolution depth encoder

Low level but high resolution features are spatially precise, and, conversely, high level but low resolution features are not spatially precise but are semantically rich. Many existing depth estimation approaches [4] are built on ResNet which encodes the input image as a low-resolution feature map. Instead, we investigate an effective architecture that is capable of fusing semantically-rich and spatially-precise features.

High-Resolution Network (HRNet) [5] maintains high resolution representations by the feature extraction process, with two key design characteristics: multiple streams with every feature map in the stream having the same resolution, and multiple stages having different resolution exchanging information in each stage. HRNet is illustrated in Figure 4.3(a) showing each stage as a red box and each stream as a row. Let $x_{r,s}^e$ denote the feature map from an HRNet encoder node located in the rth

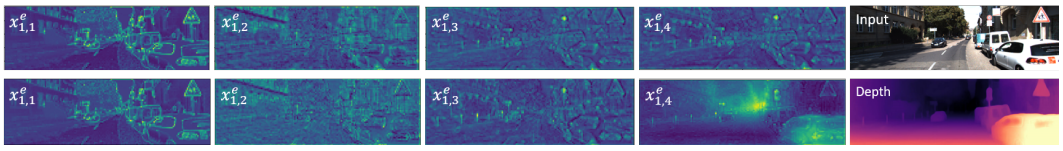


Figure 4.2: Visualisation of intermediate feature maps. We show four intermediate feature maps from stream $r = 1$ and stages $s = 1,2,3,4$ in the HRNet [5] (top) and DIFFNet (bottom) encoders. The final column shows the RGB input and DIFFNet predicted depth map.

sub-stream and at the sth stage. The resolution of sub-stream r is $\frac{1}{2^{r-1}}$ of the resolution of the first stream. As r increments, the number of channels in the feature maps doubles.

When we use an HRNet as the encoder for our depth network, we observe significant improvements over other approaches that use ResNet as the encoder. An HRNet has four streams and four stages, and outputs five feature maps at different scales from the final stage, $x^e_{0,0}$ and $x^e_{r,4}, r = 1,2,3,4$. Information from features in previous stages is ignored. We augment this module with internal feature fusion to further exploit the potential of the HRNet architecture:

Multi-stage Internal Feature Fusion Based on the relationship between feature resolution and spatial information, we assume that feature maps with more channels contain more semantic information and vice versa. To get a semantically-rich intermediate feature map without changing the scale we could increase the number of convolution kernels. However, this would dramatically increase the computational complexity. For example, given a C_{in} dimensional feature and a kernel with a size 3×3 to output a C_{out} dimensional feature, the number of trainable parameters is $C_{in} \times C_{out} \times 3 \times 3$. If we need double C_{out}, the number of parameters also doubles. HRNet contains a multi-stage convolution strategy (Figure 4.3a), and so increasing the convolution kernels leads to a large increase in parameters. However, DIFFNet forces feature maps from different stages to contain different semantic information but fuses outputs from all intermediate stages using a concatenation strategy before decoding. Without additional parameters, this strategy is capable of extracting richer feature maps – see column four in Figure 4.2, which shows a smaller semantic gap between DIFFNet encoded features and decoded outputs.

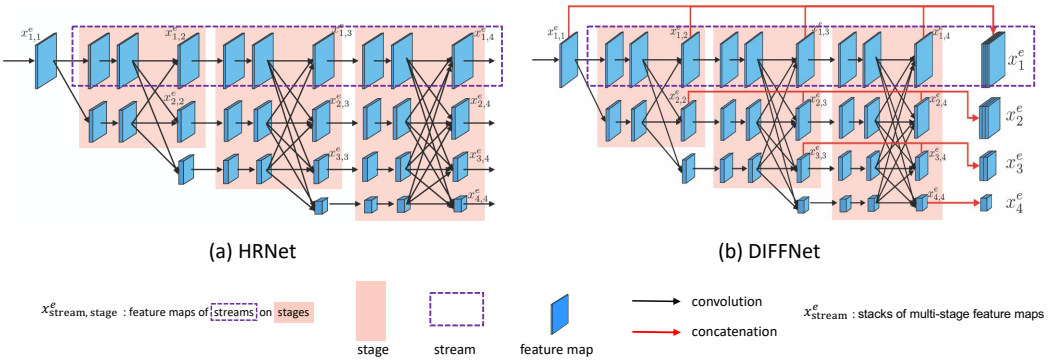


Figure 4.3: (a) Original HRNet [5] and (b) DIFFNet architecture with internal feature fusion which concatenates feature maps from multiple stages for each stream.

The stack of feature maps for stream r is computed as:

$$x_r^e = [x_{r,s}^e], \qquad s = r, \cdots, 4 \tag{4.1}$$

where $[\cdot]$ is the concatenation layer. The modified architecture is illustrated in Figure 4.3b in which the red arrows denote a concatenation of feature maps. The advantages of giving low level feature maps more semantic information (stacking multi-stage features) is explored in Section 4.3.4.

4.2.2 Attention-based depth decoder

Our decoder is based on a U-Net architecture with further inspiration taken from [6, 66, 107]. Specifically, we introduce an attention mechanism to process the skip-connections from the encoder. An illustration of the decoder can be seen in Figure 4.1 with an outline of each decoder node, D_i, shown bottom right. Let x_i^d denote the output of decoder node D_i, calculated as:

$$\begin{cases} x_4^d = \mathscr{D}(\sigma([\mu(x_4^e), x_3^e])), \\ x_i^d = \mathscr{D}(\sigma([\mu(x_{i+1}^d), x_{i-1}^e])), & i = 1,2,3 \\ x_0^d = \mathscr{D}(\sigma(\mu(x_1^d))) \end{cases} \tag{4.2}$$

where $\mu(\cdot)$ is an upsampling operator, $\sigma(\cdot)$ is an attention module, $[\cdot]$ is concatenation layer and $\mathscr{D}(\cdot)$ is a 3×3 convolution layer.

Attention Module. We explore three strategies for incorporating attention into

the decoder: channel-wise attention, spatial attention and channel-spatial attention. Given a feature map $\mathscr{F} \in \mathbb{R}^{C\times H\times W}$, the attention aggregated maps $\mathscr{F}'_{c,s,cs} \in \mathbb{R}^{C\times H\times W}$ are computed as:

$$\begin{aligned} \mathscr{F}'_c &= M_c(\mathscr{F}) \bigodot \mathscr{F}, \\ \mathscr{F}'_s &= M_s(\mathscr{F}) \bigodot \mathscr{F}, \\ \mathscr{F}'_{cs} &= M_s(\mathscr{F}'_c) \bigodot \mathscr{F}'_c. \end{aligned} \tag{4.3}$$

where $M_c(\cdot)$ and $M_s(\cdot)$ are attention map generators which output a 1D channel attention map $m_c \in \mathbb{R}^{C\times 1\times 1}$ and a 2D spatial attention map $m_s \in \mathbb{R}^{1\times H\times W}$ respectively, and $\bigodot$ denotes element-wise multiplication. During multiplication, the attention values are copied accordingly with channel attention values being broadcast along the spatial dimension, and vice versa (see [107] for details). We compare these three attention strategies in Section 4.3.4 and identify that channel-wise attention gives the best performance.

4.3 Experiments

In this section, we validate that our proposed network can output semantically-rich and spatially-precise depth maps, and our contributions improve the representation learning ability of HRNet while outperforming other published methods on the KITTI benchmark [7]. Furthermore, we analyse the characteristics of the more challenging scenes from the test partition of the KITTI dataset, and publish identifying information for the high error images.

4.3.1 Dataset

KITTI [7] is a dataset that contains stereo images and corresponding 3D lidar scans of outdoor scenes captured by imaging equipment mounted on a moving vehicle [99]. The RGB images have a resolution of $\approx 1241 \times 376$ and the corresponding depth maps are sparse with a large amount of missing data. For training, we adopt the dataset split proposed by [29]. After removing the static frames by a preprocessing step suggested by [18], this results in 39,810 monocular frame triplets for training and 4,424 frame triplets for validation. To simplify the training process,

the camera intrinsic matrices are assumed identical for all the frames in different scenes. To obtain this "universal" intrinsic matrix, we offset the principal point of the camera to the image centre and reset the focal length as the average of all the focal lengths in KITTI. This assumption is only valid when the capturing cameras are similar.

4.3.2 Implementation details

Our models are trained and tested on a single NVidia RTX 6000 GPU using Pytorch [96]. A depth network and a pose network are trained for 20 epochs using the Adam optimizer [99] with the default betas 0.9 and 0.999. They were trained with a batch size of 16 and an input and output resolution of 640×192. We set the initial learning rate as 10^{-4} for the first 14 epochs and then 10^{-5} for fine-tuning the remainder. In the objective function ℓ_{final} (Equation 3.13), we let the SSIM weight $\alpha = 0.85$ and the edge-aware smoothness weight $\beta = 1 \times 10^{-3}$.

Depth Network. We implement our proposed DIFFNet as described in Section 4.2 as our backbone. We use HRNet pre-trained only on ImageNet [98] to initialize DIFFNet (the effect of pre-training is shown in Table 4.3). At training, losses from four scaled depth maps are averaged. When testing, only the maximum resolution depth map is output by the model.

Pose Network. We implement the architecture proposed in [4] for pose estimation, which is built on ResNet-18. The pose network takes the two adjacent frames as input and outputs the relative pose which is parameterized with a 6-DOF vector. We experimented with replacing the pose encoder with HRNet, but did not achieve the same performance gains that we observe with the depth network.

4.3.3 Evaluation on KITTI

Using metrics described in Chapter 3, we evaluate the performance of DIFFNet on KITTI. The quantitative results are summarized in Table 4.1. Our method outperforms state-of-the-art approaches in terms of Absolute Relative Error and RMSE. When trained on the stereo examples in KITTI, our method achieves best results on all metrics. Given a higher image resolution of 1024×320, the accuracy of

Table 4.1: Results on KITTI Benchmark using the Eigen split grouped by training methodology. M: trained on monocular videos, MS: trained on binocular videos. Se: trained with semantic labels. The best scores are **b**old and the second are underlined.

Method	Train	WxH	lower is better				higher is better		
			Abs rel	Sq rel	RMSE	RMSE log	δ_1	δ_2	δ_3
SfMlearner [18]	M	640x192	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Li [108]	M	416x128	0.130	0.950	5.138	0.209	0.843	0.948	0.978
Chen [109]	M+Se	512x256	0.118	0.905	5.096	0.211	0.839	0.945	0.977
Monodepth2 [4]	M	640x192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
SGDepth [110]	M+Se	640x192	0.113	0.835	4.693	0.191	0.879	0.961	0.981
SAFENet [111]	M+Se	640x192	0.112	0.788	4.582	0.187	0.878	0.963	**0.983**
VC-Depth [19]	M	640x192	0.112	0.816	4.715	0.190	0.880	0.960	0.982
PackNet [63]	M	640x192	0.111	0.785	4.601	0.189	0.878	0.960	0.982
Mono-Uncertainty[10]	M	640x192	0.111	0.863	4.756	0.188	0.881	0.961	0.982
Fang [112]	M	640x192	0.111	-	4.660	0.186	0.884	0.962	0.982
HR-depth [6]	M	640x192	0.109	0.792	4.632	0.185	0.884	0.962	**0.983**
Johnston [64]	M	640x192	0.106	0.861	4.699	0.185	0.889	0.962	0.982
DIFFNet	M	640x192	**0.102**	**0.764**	**4.483**	**0.180**	**0.896**	**0.965**	**0.983**
Monodepth2 [4]	MS	640x192	0.106	0.818	4.750	0.196	0.874	0.957	0.979
HR-depth [6]	MS	640x192	0.107	0.785	4.612	0.185	0.887	0.962	0.982
Fang [112]	MS	640x192	**0.101**	-	4.512	0.188	0.881	0.961	0.981
DIFFNet	MS	640x192	**0.101**	**0.749**	**4.445**	**0.179**	**0.898**	**0.965**	**0.983**
Monodepth2 [4]	M	1024x320	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Fang [112]	M	1024x320	0.109	-	4.581	0.185	0.890	0.964	0.983
PackNet [63]	M	1280x384	0.107	0.802	4.538	0.186	0.889	0.962	0.981
SGDepth [110]	M+Se	1280x384	0.107	0.768	4.468	0.186	0.891	0.963	0.982
SAFENet [111]	M+Se	1024x320	0.106	0.743	4.489	0.181	0.884	0.965	**0.984**
HR-depth [6]	M	1024x320	0.106	0.755	4.472	0.181	0.892	0.966	**0.984**
Feat-Depth [84]	M	1024x320	0.104	0.729	4.481	0.179	0.893	0.965	**0.984**
Guizilini [113]	M+Se	1280x384	0.100	0.761	**4.270**	0.175	0.902	0.965	0.982
DIFFNet	M	1024x320	**0.097**	**0.722**	4.345	**0.174**	**0.907**	**0.967**	**0.984**

DIFFNet further increases while continuing to outperform competing methods (see in Table 4.2 for more details).

In Figure 4.4 we illustrate the qualitative performance of DIFFNet against PackNet [63], HR-depth [6] and Monodepth2 [4]. DIFFNet outperforms all self-supervised approaches and even those which use semantic labels as an external supervision resource. We draw attention to the second row that shows our method, where we have used a dashed outline to illustrate the benefits of our semantic backbone when compared with other methods. We achieve greater detail in a number of roadside items, while holding the advantage of fewer trainable parameters than the other techniques (see Table 4.4).

4.3.4 Ablation Study

To validate the performance improvements that our contributions provide, we conduct an ablative analysis. We establish a baseline by replacing the original ResNet-

Table 4.2: Quantitative results from different resolution setting training and test: ↑ represents the higher the better, and ↓, lower is better. Abs Imp means absolute improvement. The best scores in the table are **bold**.

Method	WxH	Abs Rel ↓	Abs Imp	$\delta_1 < 1.25$ ↑	Abs Imp
Monodepth2 [4]	640x192	0.115	0	0.877	0.002
	1024x320	0.115		0.879	
Fang [112]	640x192	0.111	0.002	0.884	0.006
	1024x320	0.109		0.890	
HR-depth [6]	640x192	0.109	0.003	0.884	0.008
	1024x320	0.106		0.892	
SAFENet [111]	640x192	0.112	**0.006**	0.878	0.006
	1024x320	0.106		0.884	
UnRectDepth [114]	640x192	0.107	0.004	0.894	0.003
	1024x320	0.103		0.897	
PackNet [63]	640x192	0.111	0.004	0.878	0.011
	1280x384	0.107		0.889	
SGDepth [110]	640x192	0.113	**0.006**	0.879	**0.012**
	1280x384	0.107		0.891	
DIFFNet	640x192	0.103	**0.006**	0.893	**0.012**
	1024x320	0.097		0.905	

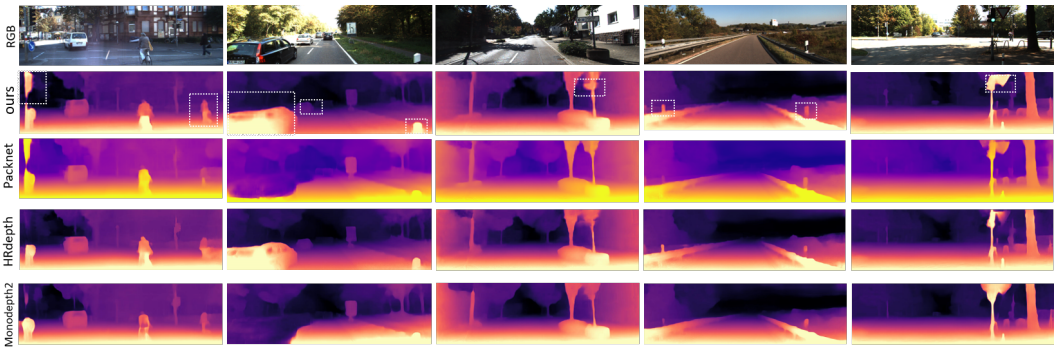


Figure 4.4: Visualisation of depth estimation results. The top row contains the input images. The second row shows the result from DIFFNet, and the remaining rows are from other contemporary methods. Note the improvement in detail for many roadside items, that our semantic backbone provides. Hotter colours indicate closer objects.

based depth encoder in Monodepth2 [4] with HRNet-18. Table 4.3 shows the results of the analysis, with the progressive addition of pre-training the encoder on ImageNet, multi-stage fusion (MF), channel-wise attention (CA) and space-wise attention (SA). The largest performance gain is achieved by pre-training the encoder rather than training from scratch. We observe that channel-wise attention yields increased accuracy compared with spatial attention. Furthermore, feature fusion improves baseline performance for all attention configurations with the exception

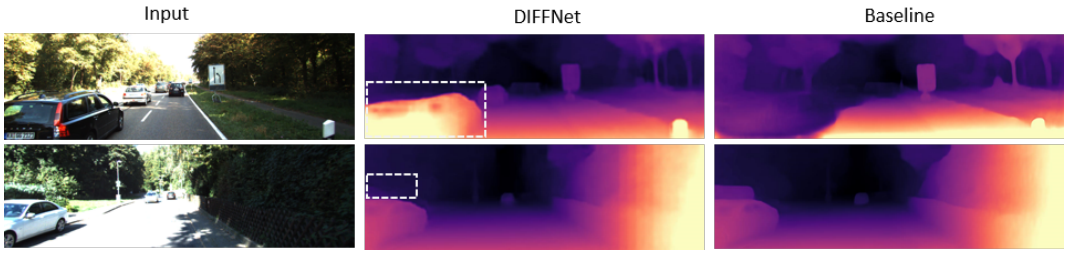


Figure 4.5: Visualisation of the ablation study. Row one shows that with more semantic information fed into depth decoder, the predicted depth map will more precise. Row two shows that DIFFNet produces a depth map with fewer artefacts than the baseline method.

of channel-spatial (CA + SA) in the last row of Table 4.3. A qualitative comparison of DIFFNet and the baseline model is shown in Figure 4.5.

Table 4.3: Ablation Studies. MF: Multi-stage Fusion. CA: Channel-wise Attention. SA: Space-wise Attention. ✓ identify our final system.

Method	Pre-train	Encoder	Decoder		The lower the better				The higher the better		
		MF	CA	SA	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
Baseline					0.124	0.990	5.158	0.202	0.858	0.952	0.974
	✓				0.108	0.799	4.609	0.186	0.888	0.963	0.982
DIFFNet		✓	✓		0.119	0.937	4.905	0.198	0.867	0.955	0.979
	✓	✓			0.105	0.817	4.593	0.183	0.893	0.964	0.982
	✓	✓	✓		**0.102**	**0.764**	**4.483**	**0.180**	**0.896**	**0.965**	**0.983**
	✓	✓		✓	0.107	0.822	4.637	0.183	0.890	0.963	0.983
	✓	✓	✓	✓	0.103	0.769	4.530	0.180	0.892	0.964	0.983

4.3.5 Extended Evaluation

Table 4.1 reveals the relative performance gap between contemporary methods on KITTI is diminishing. From empirical testing, we observe that the 10 images that give the highest error from each of these methods represents $\approx 1.4\%$ of the KITTI test set, but contributes $> 3\%$ of error when evaluating. Hence, error is not uniformly distributed throughout the test set, but certain images are more challenging than others. A model's performance on its own top 10 hard cases is a key factor in measuring its robustness and stability. For a fair comparison, we propose that the difficult cases from competing methods form a single challenge set. It is our hope that future authors will accept this strategy when they evaluate their models and compare against others.

In our case, we create a challenging test set that is the union of the 10 images with highest error from the four approaches shown in Table 4.4, including a baseline

Figure 4.6: The union set of 23 images that have the highest error from the models tested.

method discussed in Section 4.3.4. The union set comprises 23 images in KITTI benchmark: 58, **68**, **73**, **106**, 164, 173, 183, **260**, **330**, **374**, 377, 385, 386, **388**, **394**, **395**, 477, 504, 518, 548, **549**, 559, 683. Those from ours are **bold** and common hard cases are red. The corresponding images are shown in Figure 4.6, and 3 images are common to all sets of Monodepth2 [4], HR-depth [6] and our DIFFNet.

In Table 4.4 it is clear that our method performs competitively under this most difficult test, resulting in the lowest Absolute Relative Error. We can hypothesise these are the most challenging images due to the large regions of foliage in combination with difficult lighting.

Table 4.4: Quantitative results on the challenging KITTI examples. The baseline method is described in our ablation study, discussed in Section 4.3.4.

Method	Parameters	Run-time FPS	lower is better				higher is better		
			Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
Monodepth2 [4]	14.84M	99	0.213	2.197	6.468	0.295	0.741	0.906	0.950
HR-depth [6]	14.62M	116	0.205	**1.591**	**5.726**	0.282	0.738	0.902	0.957
DIFFNet	10.8M	87	**0.197**	1.803	5.988	**0.282**	**0.763**	**0.912**	**0.957**

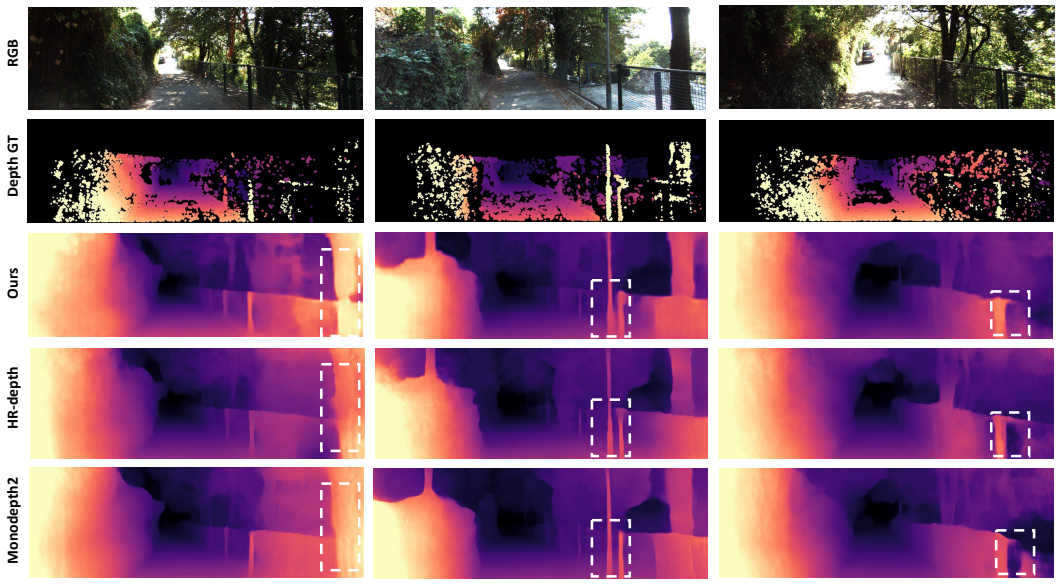


Figure 4.7: We create a hard test set of 23 images shown in Figure 4.6 that is the union set of the ten highest error images from recent well known works (Table 4.4). Here we illustrate the *intersection* set of 3 images with the corresponding depth ground truth and qualitative results from ours, HR-depth [6] and Monodepth2 [4] respectively. It shows that depth estimation on thin structures, such as continuous separation nets on the roadside, is still challenging.

4.4 Conclusion

In this chapter, we have proposed DIFFNet for self-supervised monocular depth estimation. Based on HRNet, which is designed for other computer vision tasks, we adopt it and improve it with two simple but effective strategies. Specifically, we incorporate multiple resolution feature fusion and a channel attention mechanism. With fewer parameters to learn, DIFFNet outperforms other state-of-the-art self-supervised methods, especially when high resolution input is available. We have shown that the DIFFNet encoder computes semantically rich feature maps, and our ablation study demonstrates the performance gain from each proposed modification. Finally, we introduced a creative strategy for evaluating models by investigating difficult test cases, and we invite authors to adopt the same approach going forward.

In the next Chapter, we will exploit a new training framework to boost the depth models' performance.

Chapter 5

SUB-Depth: Self-distillation and Uncertainty Boosting Self-supervised Monocular Depth Estimation

In the previous Chapter, we have shown the significant importance of a representation learning backbone. Then, in this Chapter, we present a novel training framework SUB-Depth.

Our main contribution is that we design a two-stage training framework by proposing an auxiliary self-distillation loss and incorporating it into the standard self-supervised monocular depth estimation (SDE) framework. In the first stage, a depth network is trained using the standard training framework [4]. In the second stage, given the trained network as a teacher, in addition to the photometric loss the proposed self-distillation loss is introduced to regularize a student depth network's training. When training a student network, instead of using a simple weighted sum of the photometric loss and the self-distillation loss, we employ generative task-dependent uncertainty to weight each objective in our proposed training framework. We present extensive evaluations on KITTI to demonstrate the improvements achieved by training a range of existing networks using the proposed framework, and we achieve state-of-the-art performance on monocular depth estimation.

We call our system SUB-depth, and summarise its following key contributions:

- We propose a novel two-stage training framework for self-supervised monoc-

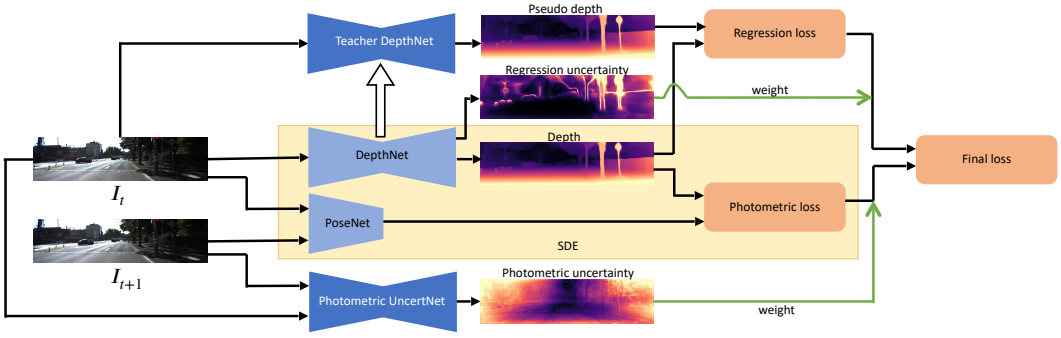


Figure 5.1: An overview of the SUB-Depth framework. SUB-Depth extends the standard existing self-supervised monocular depth estimation framework (SDE) (highlighted) using self-distillation and uncertainty modelling. The teacher DepthNet outputs a supervisory signal for training the DepthNet, and enables computation of a regression loss. Both regression and photometric uncertainty maps are learned and used to weight the respective losses. The teacher DepthNet is pretrained with the highlighted SDE framework by optimising the photometric loss.

ular depth estimation.

- Instead of manually tuning loss terms' weights, we utilize the task-dependent uncertainty idea, and experiment with several ways of uncertainty modeling.
- We conduct exhaustive experiments to show that the proposed training framework is able to boost existing models' performance significantly at the time of publication (Zhou et al. [21]). The code is available at `https://github.com/brandleyzhou/SUB-Depth`

5.1 SUB-Depth training framework

In this section, we first introduce the standard SDE framework, then the proposed self-distillation, and two task-dependent homoscedastic uncertainty formulations. The final system overview is shown in Figure 5.1.

5.1.1 Self-supervised monocular depth estimation

An SDE framework (highlighted by the yellow box in Figure 5.1) trains a DepthNet Θ_{depth} and a PoseNet Θ_{pose} simultaneously for an image reconstruction task with a triplet of sequential RGB frames $I_t \in \mathbb{R}^{H\times W\times 3}, t \in \{-1,0,1\}$. During training, a Θ_{depth} and a Θ_{pose} are optimized simultaneously using the photometric loss

$\ell_{photometric}$ defined in Equation 3.13. Please refer to Section 3.1.5 for a detailed discussion.

5.1.2 Self-distillation loss

Most related works focus on integrating other supervised learning based tasks into an SDE framework. Typically, when introducing a segmentation task, the segmentation network shares the encoder in the SDE depth network, and all components are trained jointly with the sum of the photometric loss and the segmentation loss. Although depth models trained with such a multi-task system can improve their performance, this neutralises the advantage of SDE framework, which only requires sequential images to train depth models.

Unlike existing multi-task strategies, self-distillation avoids introducing extra manual annotations. Instead, we use an SDE trained teacher depth network Θ_{teacher} to output pseudo depth ground truth $d_{pseudo} = \Theta_{\text{teacher}}(I_0)$. We then let the depth map from the DepthNet $d = \Theta_{\text{depth}}(I_0)$ regress the d_{pseudo}. The objective can be formulated as an L1 **regression loss**:

$$\ell_{regression} = |\Theta_{\text{depth}}(I_0) - \Theta_{\text{teacher}}(I_0)| \tag{5.1}$$

As Θ_{teacher} and Θ_{depth} have the same network architecture, we name this objective **self-distillation loss**.

By simply introducing a Θ_{teacher}, we retrain depth networks using following weighted loss function:

$$\ell = \omega_{pho} \times \ell_{photometric} + \omega_{reg} \times \ell_{regression} \tag{5.2}$$

Where ω_{pho} and ω_{reg} are weights for $\ell_{photometric}$ and $\ell_{regression}$ respectively. We train and evaluate models using different weighting settings, shown in Table 5.1. From the table, we observe that this naive multi-objective learning framework can output a Θ_{depth} which outperforms the Θ_{teacher} trained with standard SDE framework, no matter what the ratio of the two weights is. However, when we set $\omega_{pho} = 0.2$ and $\omega_{reg} = 0.8$, models gain the best performance for Rel Abs, while they are im-

Table 5.1: Comparison between manually tuned objective weights, evaluated on the KITTI [7] Eigen split. We experiment with several combinations of ω_{pho} and ω_{reg}. The best weighting pairs are in red. The best (Rel Abs and δ_1) scores are **bold and underlined**. Error and accuracy metrics' definitions are given in 5.3.1.

Objective weights		Error metrics				Accuracy metrics		
ω_{pho}	ω_{reg}	Rel Abs	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
0	1	0.112	0.884	4.740	0.189	0.881	0.961	0.982
0.2	0.8	**0.110**	0.855	4.724	0.188	0.881	0.961	0.982
0.4	0.6	0.112	0.866	4.736	0.189	0.881	0.961	0.982
0.5	0.5	0.112	0.888	4.766	0.189	0.882	0.961	0.981
0.6	0.4	0.113	0.876	4.774	0.189	**0.884**	0.962	0.983
0.8	0.2	0.113	0.885	4.799	0.190	0.882	0.961	0.981
1	0	0.115	0.903	4.863	0.193	0.877	0.959	0.981

proved significantly for δ_1 when $\omega_{pho} = 0.6$ and $\omega_{reg} = 0.4$. As it is hard to get an optimal weight setting, we utilize uncertainty based methods to balance loss terms automatically.

From the first row and the last row from the Table 5.1, we also observe that even if let a student network regress the outputs from a teacher network with Equation 5.1 as loss function directly, the student can outperform the teacher network trained with the photometric loss. The reason for such improvements is that the teacher-student training decouples depth network and pose network which are trained simultaneously when optimizing the photometric loss [10]. The improvement on the other rows against the last row also shows the necessity of introducing the self-distillation loss.

5.1.3 Task-dependent uncertainty formulation

Following [115], given a dataset (x, y), we let the network output the mean $\hat{y}$ and the variance σ of a posterior probability distribution $p(y|\hat{y}, \sigma)$ over ground truth y, which can be modelled as Laplacian or Gaussian. If Laplace's distribution:

$$p(y|\hat{y}, \sigma) = \frac{1}{2\sigma} \exp \frac{-|\hat{y} - y|}{\sigma} \tag{5.3}$$

is used, then the network can be trained by minimising the loss [80]:

$$loss = \frac{|\hat{y} - y|}{\sigma} + \log(\sigma) \tag{5.4}$$

where the variance σ increases when the ground truth y is unreliable. As a result, we can treat σ as task-dependent uncertainty, and the penalty term $\log(\sigma)$, avoids the degenerate solution $\sigma = +\infty$. To avoid σ being negative, we use sigmoid as the activation function for the last layer.

We introduce uncertainty modelling for each objective in the framework:

Uncertainty for image reconstruction. Intuitively, as photometric loss is a measurement of the difference between two images, it is natural to estimate its uncertainty with a model that takes two images as input. While prior works [10, 81] use the DepthNet Θ_{depth} for modelling the photometric uncertainty, we propose a separate Photometric UncertNet Θ_{pho} to estimate the uncertainty. As for the input of the proposed uncertainty network, we experiment with different settings: 1). feeding the target frame I_t, 2). feeding the target I_t and aligned $I_{t'}$ (see in Table 5.2 for more details). Finally, we let UncertNet take the target frame and the source frame as inputs and output the photometric uncertainty map σ_{pho}, as shown in Figure 5.1. Then the uncertainty weighted photometric loss, with the penalty term $\log(\sigma_{pho})$, for the image reconstruction objective is given by:

$$\ell_{reconstruction} = \frac{\ell_{photometric}}{\sigma_{pho}} + \log(\sigma_{pho}) \tag{5.5}$$

In Table 5.2, in addition to our final uncertainty modelling scheme (last row), we experiment with two input settings for the proposed Photometric UncertNet (first two rows).

Uncertainty for self-distillation. We let the DepthNet Θ_{depth} encode and output depth regression uncertainty σ_{reg}. Besides, we explore using a standalone regression uncertainty network to estimate depth uncertainty (see the 3rd row in Table 5.2). Then the uncertainty weighted regression loss with the penalty term $\log(\sigma_{reg})$ for

Table 5.2: SUB-Depth experiments with different uncertainty inputs for the Photoimetric UncertNet. First row: feeding I_t. Second row: feeding I_t and I_{t+1}. Third row: feeding I_t and warped I_{t+1}.

Input	Abs Rel	Sq Rel	RMSE log	δ_1	δ_2	δ_3
I_t	0.113	0.905	0.189	0.882	0.961	0.982
I_t and warped I_{t+1}	0.111	0.875	0.188	0.882	0.960	0.982
I_t and I_{t+1}	**0.110**	**0.821**	**0.185**	**0.884**	**0.962**	**0.983**

the self-distillation objective can be computed as:

$$\ell_{distillation} = \frac{\ell_{regression}}{\sigma_{reg}} + \log(\sigma_{reg}) \tag{5.6}$$

The self-distillation loss has been proposed by Poggi et al. [10] in the context of modelling depth estimation uncertainty. Their main purpose is to estimate predictive depth uncertainty without depth ground truth. When modelling uncertainty they train a new instance of the teacher network [4] to mimic the outputs of the teacher model, which also simultaneously generates depth uncertainty. Note that the new networks of [47] are trained only using Equation 5.6 or Equation 5.5 each time as a single objective. In contrast to theirs, our proposed student network is simultaneously trained with two objectives Equation 5.5 and Equation 5.6 with the uncertainties weighting the two losses respectively. We conduct a quantitative comparison between ours and the corresponding methods of Poggi et al. [10] in terms of depth estimation and uncertainty modelling shown in Table 5.5 and Table 5.6. .

5.1.4 Multi-objective learning with uncertainty

Finally we combine the uncertainty weighted photometric loss ($\ell_{reconstruction}$) and regression loss ($\ell_{distillation}$) to build **SUB-Depth**:

$$\ell_{final} = \ell_{reconstruction} + \ell_{distillation} \tag{5.7}$$

The result is a multi-objective learning system, which trains Θ_{depth} for an image reconstruction objective and a self-distillation objective using the sum of task-dependent uncertainty weighted losses.

The difference during training between the naive unweighted sum of losses

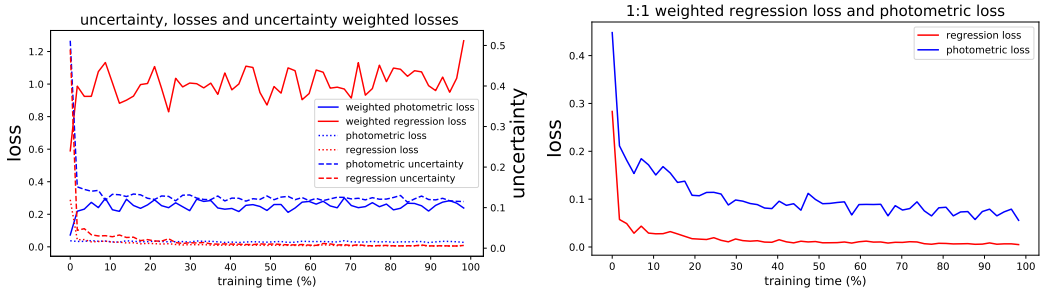


Figure 5.2: Left: The task-dependent losses, uncertainty weighted losses and uncertainty estimates during SUB-Depth training. Right: The corresponding task-dependent losses of the same system trained with no uncertainty modelling. Uncertainty modelling increases the contribution of the regression loss, and down-weights photometric loss.

and uncertainty weighted losses is shown in Figure 5.2. On the left plot, Θ_{depth} is trained with self-distillation as a prime objective. In this graph, we observe that, although the unweighted regression loss is lower than the unweighted photometric loss throughout most of the training, after applying the task-dependent uncertainty weighting the self-distillation loss contributes more to the ℓ_{final} than the reconstruction loss. This change is due to the regression uncertainty σ_{reg} being lower than the photometric uncertainty σ_{pho}, and indicates that pseudo-labels from the teacher DepthNet provide a more reliable supervisory signal than the pixel-level metrics used in the photometric loss. For comparison, the right plot in Figure 5.2 shows the naive 1:1 weighted multi-objective training framework without uncertainty modelling. In this case, the photometric loss dominates the loss throughout training following similar curves to the respective unweighted losses on the left plot.

5.2 Implementation

Our models are trained and tested on a single NVIDIA RTX 6000 GPU using PyTorch [96]. A depth network and a pose network are trained for 20 epochs using the Adam optimiser [99] with the default betas 0.9 and 0.999. They were trained with a batch size of 8 and an input and output resolution of 640×192. We set the initial learning rate as 10^{-4} for the first 14 epochs and then 10^{-5} for fine-tuning the remainder. In the objective function ℓ_{final} (Equation 5.7), we set the SSIM weight $\alpha = 0.85$ and the edge-aware smoothness weight $\beta = 1 \times 10^{-3}$.

Training Protocol. SUB-Depth is a two-stage training framework. In the first stage, a depth network and a pose network are trained with the photometric loss in Equation 3.13. In the second stage, we fix the depth network's weights and treat it as a teacher network to generate pseudo depth ground. Then we initialize a student depth network, a pose network and a photometric uncertainty network, which are trained simultaneously using the proposed loss function in Equation 5.7.

DepthNet and Teacher DepthNet. To verify the generalisation capability of SUB-Depth, we train three different architectures: Monodepth2 [4], HR-depth [6] and DIFFNet [20], which represent baseline-level, mid-level and state-of-the-art methods when trained with the standard self-supervised depth estimation framework (e.g. Monodepth2 [4]. DepthNet models are initialised on the Imagenet [98] pretrained weights. The teacher DepthNets are fixed models that are pretrained with the SDE framework. To generate the associated regression uncertainty, we modify output layers which originally produce one-channel depth maps to two-channel output.

PoseNet and Photometric UncertNet. For all training settings, we implement the architecture proposed in [4] for pose estimation, which is built on ResNet-18. The pose network takes the two adjacent frames as input and outputs the relative pose which is parameterised with a 6-DOF vector. The photometric uncertainty network uses an encoder-decoder with skip-connections. The encoder is based on the ResNet-18 architecture and the decoder follows the design of the Monodepth2 depthnet decoder [4]. The photometric uncertainty network takes adjacent frames as input and outputs photometric uncertainty maps.

5.3 Experiments and results

In this section we describe and evaluate our framework on the KITTI dataset. We explore the observed improvements in performance, and perform an ablation study to determine the contribution of each component of the SUB-Depth training framework.

5.3.1 Dataset and metrics

KITTI [7] is a dataset that contains stereo images and corresponding 3D laser scans of outdoor scenes captured by imaging equipment mounted on a moving vehicle [99]. The RGB images have a resolution of $\approx 1241 \times 376$ and the corresponding depth maps are sparse with a large amount of missing data. For training, we adopt the dataset split proposed by [29] and resize images to 640×192. After removing the static frames by a pre-processing step suggested by [18], this results in 39,810 monocular frame triplets for training and 4,424 frame triplets for validation. To simplify the training process, the camera intrinsic matrices are assumed identical for all the frames in different scenes. To obtain this "universal" intrinsic matrix, we offset the principal point of the camera to the image centre and reset the focal length as the average of all the focal lengths in KITTI. **Depth metrics** described by Eigen [29] are the most common used metrics for evaluating depth estimation accuracy. They include four error metrics: the Absolute Relative Error (Abs Rel), Squared Relative Error (Sq Rel), Root Mean Squared Error (RMSE), and the log of RMSE; accuracy metric: δ_1, δ_2, δ_3. We report each of these measures for each setting in our evaluation.

Uncertainty metric. Although uncertainty modelling is not our main contribution, we validate and compare the uncertainty outputs with two selected methods from Poggi et al. [10]. When evaluating uncertainty, we treat the depth regression uncertainty from DepthNet Θ_{depth} as depth uncertainty. From Ilg et al. [116], we use the Area Under the Sparsification Error (AUSE), the lower the better, and the Area Under the Random Gain (AURG), the higher the better, to quantify the uncertainty modelling performance of three depth metrics: Abs Rel, RMSE and δ_1, respectively in Table 5.4.

5.3.2 Evaluation on KITTI

To evaluate the performance of SUB-Depth, we select and retrain three model architectures from prior work using our training framework: Monodepth2 [4], HR-depth [6] and DIFFNet [20]. In each case, when compared to the original model (teacher DepthNet), we see significant improvements in all metrics. Table 5.3

Table 5.3: Quantitative comparison of SUB-Depth to existing SDE framework trained models on KITTI [7] Eigen split. The best results in each subsection are in **bold**. Models trained with SUB-Depth outperform the same models trained with SDE in every case.

Method	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
Monodepth2 [4]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
+ SUB-Depth	**0.110**	**0.821**	**4.648**	**0.185**	**0.884**	**0.962**	**0.983**
Improvement	0.005	0.082	0.115	0.008	0.007	0.003	0.002
HR-depth [6]	0.109	0.792	4.632	0.185	0.884	0.962	0.983
+ SUB-Depth	**0.106**	**0.770**	**4.545**	**0.182**	**0.888**	**0.963**	**0.983**
Improvement	0.003	0.022	0.087	0.003	0.004	0.001	0
DIFFNet [20]	0.102	0.764	4.483	0.180	0.896	0.965	0.983
+ SUB-Depth	**0.099**	**0.695**	**4.326**	**0.175**	**0.900**	**0.966**	**0.984**
Improvement	0.003	0.059	0.157	0.005	0.004	0.001	0.001

Table 5.4: Quantitative comparison of uncertainty modelling. We evaluate two uncertainty metrics for each selected depth metric and compare with two uncertainty modelling methods (Log and Self) in [10]. AUSE is lower the better, and AURG is higher the better.

	Abs Rel		RMSE		δ_1	
Method	AUSE	AURG	AUSE	AURG	AUSE	AURG
Poggi-Log [10]	0.051	0.027	3.097	1.188	0.060	0.056
Poggi-Self [10]	0.036	0.038	2.292	1.779	0.037	0.072
SUB-Depth	0.035	0.037	2.196	1.770	0.034	0.072

displays this quantitative comparison for all standard metrics for KITTI. We particularly draw attention to the improvement for DIFFNet, a recent state-of-the-art model, that still exhibits substantial improvement. DIFFNet trained using SUB-Depth establishes a new level of performance on the KITTI corpus. In Table 5.4, We evaluate the uncertainty modelling performance on three different depth metrics. With respect to AUSE, our proposed method outperforms other competitors from Poggi et al. [10], while, for AURG, there is a marginal gap between ours and the Self method.

To validate the performance improvements gained by SUB-Depth and evaluate the contribution of each design, we conduct an ablation study as shown in Table 5.6. Monodepth2 [4] is used as the underlying architecture for all results reported in this table. The first row $\ell_{photometric}$ is the result from the standard SDE framework, and

Table 5.5: Quantitative comparison of uncertainty modelling on improved ground truth [11].

	Abs Rel		RMSE		δ_1	
Method	AUSE	AURG	AUSE	AURG	AUSE	AURG
Poggi-Log [10]	0.039	0.020	2.562	0.916	0.044	0.038
Poggi-Self [10]	0.030	0.026	2.009	1.266	0.030	0.045
SUB-Depth	0.029	0.026	1.950	1.245	0.028	0.045

Table 5.6: Ablation Studies. We observe increased performance as self-distillation is introduced, and further improvements with the addition of uncertainty modelling. We also include results of methods Poggi-Log and Poggi-Self from Poggi et al. [10] as our counterparts. The best results in each subsection are in **bold**.

Objective	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
$\ell_{photometric}$(Baseline)	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Poggi-Log [10]	0.113	0.928	4.919	0.192	0.876	0.958	0.981
$\ell_{regression}$	0.112	0.884	4.740	0.189	0.881	0.961	0.982
Poggi-Self [10]	0.111	0.863	4.756	0.188	0.881	0.961	0.982
Ours(1:1 weighted)	0.112	0.888	4.766	0.189	0.882	0.961	0.981
Ours(uncertainty weighted)	**0.110**	**0.821**	**4.648**	**0.185**	**0.884**	**0.962**	**0.983**

performs the worst of all settings. In second row $\ell_{regression}$, by simply using the trained DepthNet as a teacher DepthNet we achieve improved performance across all measures. In last two rows, performance improves further as $\ell_{photometric}$ and $\ell_{regression}$ are combined and weighted by corresponding uncertainty estimation.

In Table 5.7, we extended our quantitative evaluation by selecting the top 10 most challenging images for each model, following the method described in Section 4.3.5 of Chapter 4. The top 10 hardest images show areas of deep shadow, poor lighting, foliage and other photographically indistinct regions. Our method deals with this uncertainty and improves on the results of all prior methods for this subset of the benchmark test set.

Qualitative evaluations are provided in Figure 5.3 for randomly selected examples. For each example, we show input RGB and output depth and regression uncertainty maps. The uncertainty map correctly marks object boundaries with high values where the transition from near to far distance is more difficult to predict. To show generalisation performance, in Figure 5.4, we additionally qualitatively evaluate the same depth network on the Cityscapes dataset [8]. Although trained

Table 5.7: Quantitative comparison of SUB-Depth to existing SDE framework trained models on top-10 selected subset of KITTI [7] benchmark. The best results in each subsection are in **bold**. Models trained with SUB-Depth outperform the same models trained with SDE in every case.

Method	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
Monodepth2 [4]	0.250	3.008	7.515	0.353	0.683	0.870	0.924
+ SUB-Depth	**0.229**	**2.451**	**6.885**	**0.330**	**0.713**	**0.876**	**0.931**
Improvement	0.021	0.557	0.63	0.023	0.030	0.006	0.007
HR-depth [6]	0.240	1.687	5.433	0.320	0.669	0.871	0.947
+ SUB-Depth	**0.222**	**1.566**	**5.176**	**0.304**	**0.710**	**0.891**	**0.949**
Improvement	0.018	0.121	0.257	0.016	0.041	0.020	0.002
DIFFNet [20]	0.225	2.160	6.357	0.312	0.712	0.899	0.951
+ SUB-Depth	**0.209**	**1.672**	**5.783**	**0.294**	**0.723**	**0.907**	**0.957**
Improvement	0.016	0.488	0.574	0.018	0.011	0.008	0.006

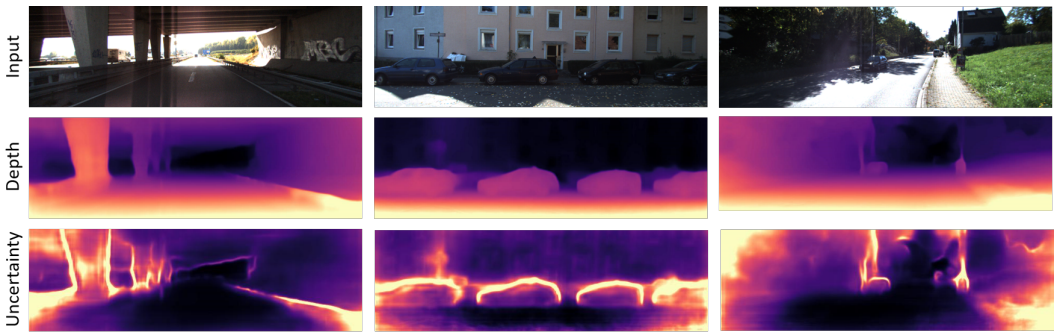


Figure 5.3: Qualitative results on KITTI [7]. We visualise the depth and the uncertainty maps from SUB-Depth trained Monodepth2. The uncertainty maps capture high uncertainty at object boundaries with a hotter color.

only on KITTI, the model appears to generalise well for both depth estimation and uncertainty modelling.

As KITTI does not have dense ground truth depth maps, we use Virtual KITTI [9] to compute depth error maps in Figure 5.5. In this qualitative evaluation we show, from top to bottom, the input RGB image, the depth error maps from the baseline Monodepth2 model and the error maps from Monodepth2 trained with SUB-Depth. For each randomly selected example, we highlight regions of the depth maps that show compelling improvements over prior work. The images are provided at high resolution to allow the reader to zoom in.

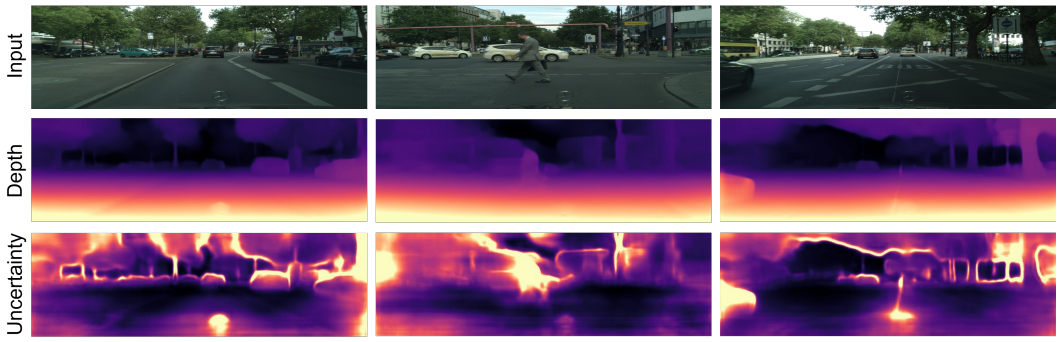


Figure 5.4: Generalisation results on Cityscapes [8]. We visualise the depth and the uncertainty maps from SUB-Depth trained only with KITTI. The uncertainty maps show higher uncertainty with a hotter color, and illustrate greater uncertainty at object boundaries and for moving objects.

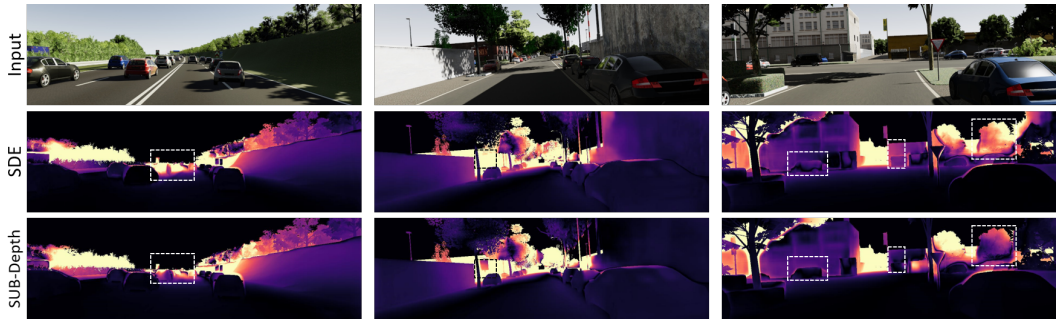


Figure 5.5: Visualisation of error map on Virtual KITTI [9]. The top row contains the synthetic input images. The second row shows the Abs rel error maps from SDE trained Monodepth2. The bottom row shows the error maps from SUB-Depth trained Monodepth2. The differences are highlighted by white dotted boxes.

5.4 Conclusion

In this Chapter, we presented a two-stage training framework for self-supervised monocular depth estimation, SUB-Depth. SUB-Depth extends the existing standard depth estimation framework with the introduction of self-distillation and uncertainty modelling. We introduce a teacher network and let the depth network be trained, not only for an image reconstruction objective but also for a self-distillation objective. To find the optimal objective weights, we utilize task-dependent uncertainty to weight losses for each objective. Through analysing losses and uncertainty during training, we discovered that, initially the image reconstruction loss contributes more than the self-distillation loss, but then self-distillation quickly becomes the primary objective since the estimated regression uncertainty is much lower than photomet-

ric uncertainty. We retrained three representative approaches using SUB-Depth to validate the generalisation capability of our proposed framework, and all outperform their counterparts. Our SUB-Depth training framework exhibits substantial improvements over the current state-of-the-art model on the KITTI benchmark for all depth metrics at the time of publication.

In the following Chapter, we will conclude the findings of this thesis.

Chapter 6

Conclusion and Future work

Single-view depth estimation, which predicts depth from a single image, has a wide range of potential applications across various fields. In autonomous vehicles and robotics, it enhances navigation safety through obstacle detection and improves object detection and tracking by providing spatial context. In Augmented Reality (AR) and Virtual Reality (VR), depth estimation facilitates scene structure understanding, enabling accurate placement of virtual objects, and enhances the realism of virtual environments. In photography and videography, depth estimation enhances autofocus systems, enables post-processing depth-of-field effects, and assists in 3D photography and image segmentation. Healthcare applications include 3D reconstruction of anatomical structures from 2D images for diagnostics and surgical planning.

To benefit the above-mentioned applications, the main goal of this thesis was to continuously improve the performance of self-supervised monocular depth estimation approaches. In this Chapter, we conclude with our novel contributions to solving this challenging problem and discuss potential improving directions for future work.

6.1 Contibutions

In Chapter 3, we first achieved our research goal by designing novel loss functions and building the associating data flow in the training phase. The motivation for our proposed loss function was that the original appearance-based photometric loss is sensitive to moving objects in the KITTI dataset. We initially assumed that the

camera-mounted vehicle was moving at a constant velocity when capturing data in a short time window. Based on this assumption, we proposed a depth consistency constraint among consecutive frames on pixels that belong to moving objects. To detect those pixels where our proposed depth consistency loss is valid, we designed an associating mask schema, inspired by Godard et al. [4]. Furthermore, we exploited relative pose change consistency between frames as a loss term to supplement the photometric loss. As far as we know, we were the first to explore such geometry consistencies without introducing additional data annotations. As a result, our depth model outperformed the baseline method on the KITTI benchmark [7], and surprisingly the pose estimation model was improved on visual odometry task as well.

To further improve the accuracy of depth estimation models, many prior works have made efforts to construct representation learning network architectures. So in Chapter 4, we proposed a novel learning backbone **DIFFNet**. DIFFNet consists of a high-resolution encoder and an attention-based depth decoder. The encoder is based on HRNet proposed by Wang et al. [5] that makes the most use of maintaining high resolution information and exchanging information between feature maps in different resolutions during the down-sampling process. Since such a design costs more trainable parameters when we want to increase the dimensions of extracted feature maps, we introduced a multi-stage internal feature fusion mechanism that enables the extraction of more semantic information by simply concatenating feature maps in different sampling stages. Furthermore, we explored different attention designs for feature decoding. After conducting the comparison experiments among spatial-wise attention, channel-wise attention and a combination of them, we proposed a channel-wise attention based depth decoder. To better compare with other methods, we proposed an extended evaluation method which enables researchers to better compare their approaches to others. As a result, the DIFFNet achieved state-of-the-art performance on both KITTI benchmark and our proposed evaluation method.

As explored for other computer vision tasks, multi-task learning has been a technique to improve one task by introducing another task during training. Previous

works [111, 117, 118, 119] have shown that training a single depth model benefits from multiple regression or classification objectives. In Chapter 5, inspired by prior works that train a student depth network with a trained teacher network [6, 10], we extended the single-task self-supervised depth estimation framework to a two-stage setting by introducing a self-distillation scheme associated with a regression objective. Compared with other multi-task settings which introduce supervised tasks such as semantic segmentation, one of the advantages of self-distillation is that the framework remains a self-supervised regime. The performance of multi-objective systems is dependent on the relative loss weighting for each task. Instead of manually tuning weights of loss terms, inspired by Kendall and Gal [120], we propose two uncertainty modelling strategies to calculate uncertainty for the self-distillation task and the image reconstruction task respectively. Specifically, the self-distillation uncertainty down-weights the regression loss when a teacher network outputs noisy depth values, and the photometric uncertainty outputs higher confidence where input frames satisfy the image reconstruction tasks' assumptions, that is, static world and ego-motion. As a result, our proposed training framework is able to further improve the performance of methods including **DIFFNet** discussed in Chapter 4.

6.2 Future work

While through the three chapters discussed our depth estimation model has shown improved performance in terms of higher accuracy and lower error rates on the KITTI benchmark, there are many problems unsolved, e.g. geometric consistency with other scene structure information, for instance, surface normals.

Given a depth map, we can calculate its corresponding surface normal. If we assume the optic axis of the camera is parallel to the ground, we can detect ground pixels according to their surface normal values as shown in the second column in Figure 6.1. However, the ground masks generated by depth maps are much different from the ground semantic maps, especially in shadows and lane marks on carriageways as the red points shown in the last column in Figure 6.1. It means that depth results in those areas are inaccurate, while the model achieves a higher overall

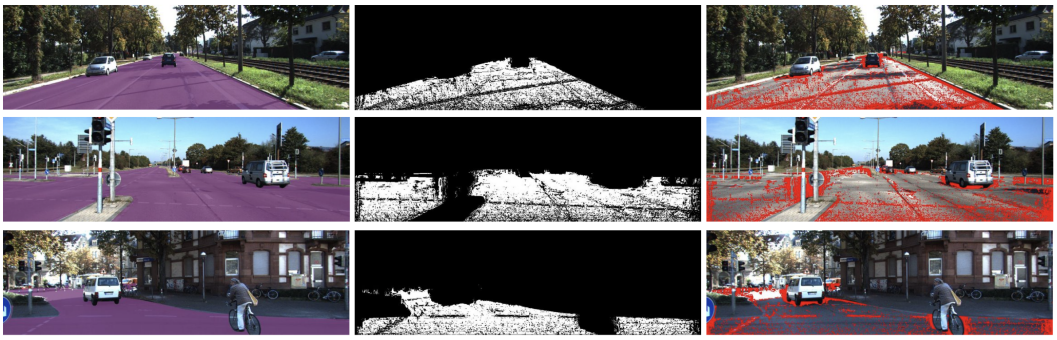


Figure 6.1: Geometric consistency on road (ground) pixels. The first column shows input images with their ground semantic maps in purple. The second column shows ground masks calculated from depth maps generated by the method in Chapter 4, in which white points represent ground pixels. Note that those ground masks from depth have been masked by the corresponding ground semantic maps in the first column. The last column shows the differences between ground semantic maps and ground masks from depth using red points.

accuracy.

We have tested other depth estimation approaches, but the problem still exists. In future, we would like to solve this problem by exploring the possibility of imposing a geometric constraint on the depth map and surface normals.

Apart from the unexplored surface normal constraint, there are yet well-solved problems for future work. The biggest challenge of self-supervised monocular depth estimation is dynamic objects or stationary cameras, which contaminate the photometric loss and lead to degradation of models' performance. Although some works have proposed approaches (e.g. explainability mask [18], auto-masking [4]) partially alleviating such issues by filtering out those regions in loss calculation, it is worth exploring multi-frame based architectures to solve this problem by modelling geometry information across temporal frames and disentangling object motions from scene changes.

Another significant challenge is depth estimation on non-Lambertian surfaces. Non-Lambertian surfaces exhibit specular reflection, where light is reflected in a specific direction rather than diffused evenly. This causes highlights or shiny spots, as seen on glossy or metallic surfaces. In the context of monocular depth estimation, Bright highlights from specular reflections can be mistaken for object features, lead-

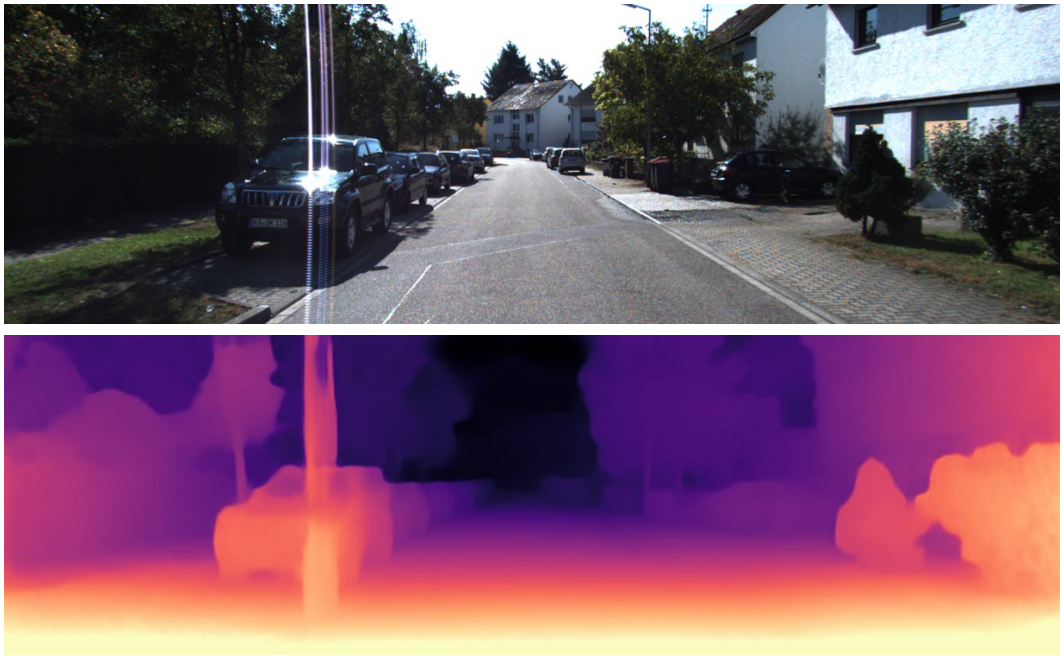


Figure 6.2: Top image: an example contains a non-Lambertian surface on a vehicle. Bottom image: a corresponding depth map containing artefacts due to the highlight caused by specular reflection.

ing to errors in depth maps as Figure 6.2 shown. The ability to accurately estimate depth in the presence of non-Lambertian surfaces can be achieved by incorporating more realistic reflectance models. Another solution is to utilise multiple viewpoints or images which can help disambiguate the effects of non-Lambertian reflections by providing additional information about the scene structure.

Bibliography

[1] David J Crandall, Andrew Owens, Noah Snavely, and Daniel P Huttenlocher. Sfm with mrfs: Discrete-continuous optimization for large-scale structure from motion. *Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2841–2853, 2012.

[2] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Computer Vision (ICCV)*, 2015.

[3] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[4] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *International Conference on Computer Vision (ICCV)*, 2019.

[5] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[6] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised

monocular depth estimation. *In AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 2013.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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

[12] S. Kumawat, M. Verma, and S. Raman. Lbvcnn: Local binary volume convolutional neural network for facial expression recognition from image sequences. In *Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[13] H. Lyu, H. Fu, X. Hu, and L. Liu. Esnet: Edge-based segmentation network for real-time semantic segmentation in traffic scenes. In *International Conference on Image Processing (ICIP)*, 2019.

[14] Ravi Garg, Vijay Kumar BG, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision (ECCV)*, 2016.

[15] Takanori Senoh, Koki Wakunami, Hisayuki Sasaki, Ryutaro Oi, and Kenji Yamamoto. Fast depth estimation using non-iterative local optimization for super multi-view images. In *Global Conference on Signal and Information Processing*, 2015.

[16] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[17] Qijie Li, Tianqing Chang, and Xuejun Jiao. A new targets matching method based on epipolar geometry. In *International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems*, pages 135–139, Tianjin, China, 2012. IEEE.

[18] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] Hang Zhou, David Greenwood, Sarah Taylor, and Han Gong. Constant velocity constraints for self-supervised monocular depth estimation. In *European Conference on Visual Media Production (CVMP)*, 2020.

[20] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *British Machine Vision Conference (BMVC)*, 2021.

[21] Hang Zhou, Sarah Taylor, David Greenwood, and Michal Mackiewicz. Subdepth: Self-distillation and uncertainty boosting self-supervised monocular depth estimation. In *British Machine Vision Conference (BMVC)*, 2022.

[22] C. Wu. Visualsfm: A visual structure from motion system. *http://ccwu.me/vsfm*, 2020,MAY,01.

[23] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *International Conference on Computer Vision (ICCV)*, 2011.

[24] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *European Conference on Computer Vision (ECCV)*, 2020.

[25] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[26] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2144–2158, 2014.

[27] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. *Transactions on Graphics*, 2005.

[28] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009.

[29] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Conference on Neural Information Processing Systems (NIPS)*, 2014.

[30] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[31] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[32] A. Atapour-Abarghouei and T. P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[33] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *European Conference on Computer Vision (ECCV)*, 2018.

[34] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision (ECCV)*, 2018.

[35] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2020.

[36] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[37] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 2016.

[38] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *International Conference on Computer Vision (ICCV)*, 2015.

[40] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single image depth perception in the wild. In *Conference on Neural Information Processing Systems*, page 730–738. ACM, 2016.

[41] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, page 942–960, 2018.

[42] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[43] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *European Conference on Computer Vision (ECCV)*, 2018.

[44] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[45] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[46] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Transactions on Image Processing (TIP)*, 2004.

[47] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *International Conference on 3D Vision (3DV)*, 2018.

[48] Sudeep Pillai, Rareş Ambruş, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *International Conference on Robotics and Automation (ICRA)*, 2019.

[49] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[50] Juan Luis Gonzalez and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *In Advances in Neural Information Processing Systems (NIPS)*, 2020.

[51] Juan Luis Gonzalez and Munchurl Kim. Plade-net: Towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[52] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *International Conference on Computer Vision (ICCV)*, 2019.

[53] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[54] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[55] Art B Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7):59–72, 2007.

[56] Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[57] Rui Peng, Ronggang Wang, Yawen Lai, Luyang Tang, and Yangang Cai. Excavating the potential capacity of self-supervised monocular depth estimation. In *International Conference on Computer Vision (ICCV)*, 2021.

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

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[60] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2020.

[61] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision (ICCV)*, 2021.

[62] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[63] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[64] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[65] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[66] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[67] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised monocular depth estimation. In *International Conference on 3D vision (3DV)*, 2021.

[68] Mu He, Le Hui, Yikai Bian, Jian Ren, Jin Xie, and Jian Yang. Ra-depth: Resolution adaptive self-supervised monocular depth estimation. In *European Conference on Computer Vision (ECCV)*, 2022.

[69] Tak-Wai Hui. Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[70] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[71] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. *arXiv preprint arXiv:2208.03543*, 2022.

[72] Jinwoo Bae, Sungho Moon, and Sunghoon Im. Deep digging into the generalization of self-supervised monocular depth estimation. *arXiv preprint arXiv:2205.11083*, 2022.

[73] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. *arXiv preprint arXiv:2211.13202*, 2022.

[74] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in Neural Information Processing Systems (NIPS)*, 2021.

[75] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 5(4):6813–6820, 2020.

[76] Jianrong Wang, Ge Zhang, Zhenyu Wu, XueWei Li, and Li Liu. Self-supervised joint learning framework of depth estimation via implicit cues. *arXiv preprint arXiv:2006.09876*, 2020.

[77] Vitor Guizilini, Rareș Ambruș, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[78] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Neural Information Processing Systems (NIPS)*, 2017.

[79] Patrick Ruhkamp, Daoyi Gao, Hanzhi Chen, Nassir Navab, and Beniamin Busam. Attention meets geometry: Geometry guided spatial-temporal attention for consistent self-supervised monocular depth estimation. In *International Conference on 3D Vision (3DV)*, 2021.

[80] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *European Conference on Computer Vision (ECCV)*, 2018.

[81] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[82] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth learning from video. *International Journal of Computer Vision*, 129(9):2548–2564, 2021.

[83] Lijun Wang, Yifan Wang, Linzhao Wang, Yunlong Zhan, Ying Wang, and Huchuan Lu. Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? In *International Conference on Computer Vision (ICCV)*, 2021.

[84] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *European Conference on Computer Vision (ECCV)*, 2020.

[85] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[86] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision (ECCV)*, 2018.

[87] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI conference on artificial intelligence (AAAI)*, 2018.

[88] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[89] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *International Conference on Computer Vision (ICCV)*, 2019.

[90] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[91] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[92] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[93] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *arXiv preprint arXiv:1711.03665*, 2017.

[94] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[95] Jaderberg Max, Simonyan Karen, Zisserman Andrew, et al. Spatial transformer networks. In *Advances in neural information processing systems (NIPS)*, 2015.

[96] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[97] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[98] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[99] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[100] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[101] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[102] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *In International Conference on Computer Vision (ICCV)*, 2011.

[103] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[104] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[105] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense convolutional networks for efficient prediction. *arXiv preprint arXiv:1703.09844*, 2(2), 2017.

[106] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Tremeau, and Christian Wolf. Residual conv-deconv grid network for semantic segmentation. In *British Machine Vision Conference (BMVC)*, 2017.

[107] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision (ECCV)*, 2018.

[108] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning (CoRL)*, 2020.

[109] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[110] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *European Conference on Computer Vision (ECCV)*, 2020.

[111] JaeHoon Choi, Dongki Jung, DongHwan Lee, and Changick Kim. Safenet: Self-supervised monocular depth estimation with semantic-aware feature extraction. In *Conference on Neural Information Processing Systems (NIPS)*, 2020.

[112] Zhicheng Fang, Xiaoran Chen, Yuhua Chen, and Luc Van Gool. Towards good practice for cnn-based monocular depth estimation. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[113] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *International Conference on Learning Representations (ICLR)*, 2020.

[114] Varun Ravi Kumar, Senthil Yogamani, Markus Bach, Christian Witt, Stefan Milz, and Patrick Mader. Unrectdepthnet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[115] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Conference on Neural Information Processing Systems (NIPS)*, 2017.

[116] Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *European Conference on Computer Vision (ECCV)*, September 2018.

[117] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[118] Lukas Liebel and Marco Körner. Multidepth: Single-image depth estimation via multi-task regression and classification. In *Intelligent Transportation Systems Conference (ITSC)*, 2019.

[119] Yawen Lu, Michel Sarkis, and Guoyu Lu. Multi-task learning for single image depth estimation and segmentation based on unsupervised network. In *International Conference on Robotics and Automation (ICRA)*, 2020.

[120] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.