Learning Unsupervised Depth Estimation, from Stereo to Monocular Images

Andrea Pilzer

ICT International Doctoral School,
Department of Information Engineering and Computer Science

The University of Trento

Advisors: Prof. Dr. Nicu Sebe and Prof. Dr. Elisa Ricci
A thesis submitted for the degree of

Doctor of Philosophy (PhD)

Commitee:

- Prof. Luigi Di Stefano,
  University of Bologna, Italy
- Prof. Vittorio Murino,
  University of Verona, Italy
- Prof. Nicola Conci,
  University of Trento, Italy

Day of the defense: 22/6/2020

Abstract

In order to interact with the real world, humans need to perform several tasks such as object detection, pose estimation, motion estimation and distance estimation. These tasks are all part of scene understanding and are fundamental tasks of computer vision. Depth estimation received unprecedented attention from the research community in recent years due to the growing interest in its practical applications (*i.e.* robotics, autonomous driving, etc.) and the performance improvements achieved with deep learning. In fact, the applications expanded from the more traditional tasks such as robotics to new fields such as autonomous driving, augmented reality devices and smartphones applications. This is due to several factors. First, with the increased availability of training data, bigger and bigger datasets were collected. Second, deep learning frameworks running on graphical cards exponentially increased the data processing capabilities allowing for higher precision deep convolutional networks, ConvNets, to be trained. Third, researchers applied unsupervised optimization objectives to ConvNets overcoming the hurdle of collecting expensive ground truth and fully exploiting the abundance of images available in datasets.

This thesis addresses several proposals and their benefits for unsupervised depth estimation, *i.e.* , (i) learning from resynthesized data, (ii) adversarial learning, (iii) coupling generator and discriminator losses for collaborative training, and (iv) self-improvement ability of the learned model. For the first two points, we developed a binocular stereo unsupervised depth estimation model that uses reconstructed data as an additional self-constraint during training. In addition to that, adversarial learning improves the quality of the reconstructions, further increasing the performance of the model. The third point is inspired by scene understanding as a structured task. A generator and a discriminator joining their efforts in a structured way improve the quality of the estimations. Our intuition may

sound counterintuitive when cast in the general framework of adversarial learning. However, in our experiments we demonstrate the effectiveness of the proposed approach. Finally, self-improvement is inspired by estimation refinement, a widespread practice in dense reconstruction tasks like depth estimation. We devise a monocular unsupervised depth estimation approach , which measures the reconstruction errors in an unsupervised way, to produce a refinement of the depth predictions. Furthermore, we apply knowledge distillation to improve the student ConvNet with the knowledge of the teacher ConvNet that has access to the errors.

To my loved Leysan that pushes me to undertake new adventures,
To my family that supported me during these years,
To Paolo that encouraged me to pursue this journey,
To all the people I worked with, in particular Nicu and Elisa,
Thank you, Andrea.

Contents

[1] "Unsupervised Adversarial Depth Estimation Using Cycled Generative Networks", A. Pilzer, D. Xu, M. Puscas, E. Ricci and N. Sebe; 2018 International Conference on 3D Vision (3DV), Verona, 2018, pp. 587-595

[2] "Progressive Fusion for Unsupervised Binocular Depth Estimation using Cycled Networks", A. Pilzer, S. Lathuilière, D. Xu, M. M. Puscas, E. Ricci and N. Sebe; IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2019.2942928

[3] "Structured Coupled Generative Adversarial Networks for Unsupervised Monocular Depth Estimation", M. M. Puscas, D. Xu, A. Pilzer and N. Sebe; 2019 International Conference on 3D Vision (3DV), Québec City, QC, Canada, 2019, pp. 18-26

[4] "Refine and Distill: Exploiting Cycle-Inconsistency and Knowledge Distillation for Unsupervised Monocular Depth Estimation", A. Pilzer, S. Lathuilière, N. Sebe, E. Ricci; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9768-9777

1

Introduction

1.1 Motivation

We are in the middle of a data driven revolution, given that the invention of writing around 3000 BC, 90% of the data produced by humans was generated in 2017 and 2018[5] alone. This vast amount of data is nowadays commonly defined as *big data*. The trend is still exponentially growing with the expansion of web-related services such as streaming platforms (*e.g.* YouTube, Netflix), social networks (*e.g.* Facebook, SnapChat, Instagram) and messaging applications (*e.g.* WhatsApp, Telegram) to cite some of them. Users around the world consume all these data, and there is still room for expanding services and market opportunities. The huge amount of data available to web-services corporations such as Amazon, Google and Facebook allowed them to become the most valuable companies by market capitalization in the world, and among the most profitable companies in the world. However, it is not only the data that made them valuable, but also the capability to use it.

The tool used by these big corporations is Artificial Intelligence (AI). It is employed to analyze the data and extract useful and remunerative information, *e.g.* , suggesting to a potential customer an item to purchase, a new brand they could like or a piece of news that may be of interest. We are fully immersed in a big hype created by the media around AI, where it would be more appropriate to talk about machine learning. Machine learning is, in reality, the tool used to conveniently extract statistics, patterns, and generate forecasts starting from big data. Recently, a branch of machine learning stood out and started to revolutionize many

[5] https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/

applications, *i.e.* , deep learning. Deep learning is a family of machine learning algorithms for representation learning. It is based on artificial neural networks, *i.e.* stacks of non-linear layers composed of units or neurons that perform simple operations. As a consequence, these networks can be expanded to have a very big capacity and consequently be powerful enough to represent complex data. The drawback is that until very recently, the computational power required to train them and the data did not exist. The processing power issue was resolved in the early 2010s with the development of deep learning frameworks running on graphic cards (GPUs). At the same time, the research community proposed more demanding challenges based on big datasets. In this dissertation, we study computer vision, and the *paradigm shift* towards deep learning for images came from the spectacular success of AlexNet [1] on the ImageNet [2] Challenge.

From that point on, deep learning started to outperform all other methods in terms of accuracy. AlexNet is a Convolutional Neural Network (ConvNet), composed of several convolutional filters that extract image representations, from low-level features to high-level representation of the whole image. Indeed, from then, the research has evolved rapidly, and better architectures have been proposed. However, we can still list some key advantages and key disadvantages of deep learning. The advantages are the big capacity of the models that can anchieve great performance and the ability to learn an efficient representation of the data without humans designing the feature extractor. The drawbacks are, to some extent, the low generalization ability, *i.e.* these models are very specialized on the data used for training, they require big amounts of labelled data to supervise the training and computational resources to perform it. Addressing the labelling problem, in this thesis, we will work towards improving depth estimation state-of-the-art with self-supervision. In other words, we aim to train good models by using only images and avoiding costly and tedious labelling. In fact, for depth estimation, every image depicting a scene requires a LiDaR synchronized with the camera to register the real distance of every point of the scene from the camera. Introducing errors is probable if the sensors set up is not arranged carefully with calibration and synchronization.

While the cost of annotations for thousands of images can be easy to grasp, let us introduce the reason behind the choice of depth estimation. Depth is a central task of scene understanding in computer vision, along with semantic segmentation, object recognition and motion understanding for example. These are all tasks that we, as humans, learned to perform very efficiently to find food, relate with other people and avoid dangers. Therefore, it is of great importance for us, in particular now that we are able to build any kind of machines, to

instruct them to relate with the real world. Depth has potentially infinite applications, surgery micro robots, autonomous cars, industrial robots, big machinery for natural resources extraction, remote sensing. All these applications do not rely only on depth but on an ensemble of sensors and scene understanding algorithms among which depth is an important component. We mentioned autonomous driving, and we will now provide a brief example of the multiple uses we may have for a depth estimation algorithm on a car. The main task of depth estimation would be probably, to provide redundancy for the LiDaR used to sense the distance from other objects. Nevertheless, it has the advantage to produce dense estimations and not point clouds. Moreover, it can be combined with other tasks such as semantic segmentation and optical flow; in fact an object has a flow in the scene and can probably be segmented with a single semantic label. When the object is identified, re-identification could be performed to track it. In addition to that, since the real world is in 3D, traditional object detection with a 2D bounding box is not sufficient, so depth estimation is used to the provide the 3D object detection coordinates for the points delimiting the object. The 3D bounding boxes are then exploited for the collision avoidance, for example, with pedestrians. This approach is only one of the examples of the possible application of depth in the real world, and we propose it as a motivation for our depth estimation studies.

1.2 Outline

In this thesis, we present the advances on unsupervised depth estimation achieved during the doctoral studies. Depth estimation methods learn to infer the depth of a scene from an image, the supervised ones need dense depth annotations of the pixels in the image to learn with in a supervised fashion. Whereas, unsupervised methods exploit scene geometry to devise effective optimization objectives to learn the depth of the scene. Therefore, we chose as our field of research unsupervised depth estimation. More in detail, we researched whether adversarial learning and data augmentation can improve unsupervised depth estimation, whether generator and discriminator in an cooperative adversarial learning setting can be more accurate for unsupervised depth estimation, and if a model can refine the predictions it inferred without any ground truth supervision.

We demonstrated that they positively impact unsupervised depth estimation methods, and the details will be addressed more extensively in the following chapters. In Chapter 4, "Stereo Adversarial Depth Estimation", we will address adversarial learning and use of resynthesized

data - starting from Section 4.1 - and then further improvement of our method with better integration of stereo features in a follow-up work presented in Section 4.2. In the following Chapter 5, "Structured Coupled Depth Estimation", we will continue our research investigating if we could improve our work with adversarial learning to obtain collaborative network structure between discriminator and generator. In this case, we will describe the design of a conditional random field that couples the losses of generator and discriminator and jointly backpropagates them. Finally, in Chapter 6 "Monocular Depth Refinement", we will present our research on model knowledge self-distillation and prediction refinement in the challenging setting of monocular depth estimation.

In the remaining of this Chapter, we will give an overview of the topics addressed in each Chapter (sub-sections 1.2.1, 1.2.2 and 1.2.3). After that, in sub-section 1.3, we will list our main contributions, then the last sub-section of the Introduction 1.4 will contain the papers published during the author's doctoral studies.

1.2.1 Stereo Adversarial Depth Estimation

Stereo matching is a well known problem in computer vision. The research community developed unsupervised or self-supervised depth estimation to solve it without requiring disparity or depth ground truth. Given two stereo views of the same scene, it is possible to calculate the dense correspondence map or disparity that aligns each corresponding pixel. Furthermore, in recent years, brilliant performance were shown from deep learning methods [3, 4, 5] based on Convolutional Neural Networks (ConvNets). Inspired by these successes, and the advances in image generation based on Generative Adversarial Networks (GANs) [6, 7, 8], in Chapter 4, we will explore if adversarial learning is beneficial for unsupervised depth estimation.

In Section 4.1, we will present the first part of our studies focusing on adversarial learning and learning from resynthesized data. GANs were designed to generate realistic images from random noise and some adaptation to our problem was needed. Firstly, the generator for stereo depth estimation takes in two stereo views and outputs the scene disparity. Secondly, the disparity is used by a warping function to reconstruct one of the views and then learn from the reconstruction error. We proposed then to use the discriminator network on top of the reconstructed image to infer if the image was reconstructed or original. In this way, we applied the adversarial learning principle to unsupervised depth estimation.

To complement the proposed improvement in disparity estimation and thus image reconstruction quality, we introduce reconstructed images as input data. Our intuition is that better

quality of reconstructed images can be used to tune our model during learning further. Therefore, we propose to use a second network stacked after the first one to estimate depth from the reconstructed images. These two networks then form a *cycle* where the similarity of the final reconstructed images are evaluated against the original ones. The concept is similar to previous research [7, 8] where a source domain image is translated to a target domain (*e.g.* sketch to picture) and then back to the source domain. Although, with the difference that in our work, the source and target images are from the same domain and different stereo views.

In Section 4.2, we will present the continuation of the work in 4.1 where we focus on feature fusion for better depth estimations, simplification of the training process and significantly extend our ablation studies and state-of-the-art benchmarking. The research community introduced several methods to obtain better feature representation from the two stereo views and consequently, achieving performance gains. These techniques, such as cost volumes and 3D convolutions [9] are computationally costly. Instead, in our work, we offer a simple and effective feature warping and concatenation that improves performance at a lower computational cost.

1.2.2 Structured Coupled Depth Estimation

Our works demonstrated the benefit of introducing adversarial learning for unsupervised depth estimation. Therefore, we took our research one step forward, and we explored if generator and discriminator could cooperate to get better results. We introduce a monocular method, at inference time, for unsupervised depth estimation with generator and discriminator that couple the losses to actively learn from each other (Chapter 5). In detail, during training, the generator infers the disparity map and the synthesizes the stereo view, then the reconstruction error is computed. The discriminator outputs a per-pixel decision whether the image is real or synthetic. The coupling module takes all this information as input and processes it before backpropagation. In this way, a joint optimization process is performed for both generator and discriminator.

The proposed coupling is based on a Conditional Random Field (CRF). CRFs have been already employed successfully for supervised depth estimation [10, 11]. Unlike previous research, we propose to use CRFs for coupling generator and discriminator losses in an unsupervised depth estimation model. This choice is based on two primary goals. First, to exploit the structure of the images used to train the model with the CRF and second, to learn the best way to couple the losses without having to hand-craft any rule.

1.2.3 Monocular Depth Refinement

In the last Chapter 6, we will focus on monocular unsupervised depth estimation with prediction refinement. This scenario is more challenging because the model needs to infer from a single stereo view the depth of the scene. During training, the model is trained with self-supervised learning on both stereo views. However, during inference, one stereo view is enough to infer the pixel-wise depth map. Depth estimation literature, as in Eigen *et al.* [12] and Godard *et al.* [5], demonstrated the ability of the monocular setting to obtain high accuracy. In our work, building on previous findings, we proposed a self-refinement strategy to refine the model predictions [13, 14] and show with extensive experimental results that it dramatically improves performances. Finally, we propose self-distillation as a way to improve the original estimations with the additional knowledge available to the refinement network. Distillation is the practice of training a smaller *student* model with additional information from a bigger and more accurate *teacher* model, a more complete introduction about distillation can be found in Related Works Section 2.5. In our case, we call it self-distillation because the teacher and the student are part of the same model and contribute to the optimization process together.

More in detail, we devise a monocular cycled network capable of resynthesizing the input image and thus measure, in an unsupervised fashion, the reconstruction error. We call this first block *student* network. Later the error is employed by a refinement network, called *teacher*, for refining the predictions. The choice of the name *student* and *teacher* is due to the fact that the *teacher* infers the depth, not only from the RGB images as the *student*, but can exploit additional information contained in the reconstruction error aforementioned. On top of that, we propose to use the knowledge distillation paradigm to transfer knowledge from the better performing teacher to the student. While knowledge distillation is interesting from a research perspective, it is even more useful in practice, because in a real case scenario with finite resources it allows to achieve good performance by employing only the *student* network.

1.3 Contributions

Working in the context of 3D scene understanding, on one side, we contribute to the field of unsupervised depth estimation with novel models. On the other we explore alternative optimization objectives. We then perform an extensive qualitative and quantitative experimental evaluation of the proposed solutions, to quantify their effectiveness.

Specifically, we present

- *improved reconstruction loss via adversarial learning*, we devise adversarial learning, in the context of unsupervised depth estimation, by integrating it in the image recostruction optimization objective. Concretely, as our experiments demonstrated, a better image reconstruction translates in more accurate depth inference.

- *additional optimization constraint with resynthesized data*, we propose a cycled generative network. In this way, the model will learn from both original images and the resynthesized images used for the computation of the reconstruction loss. Moreover, the cycled network, not only exploits resynthetized images, but also enforces additional constraints resulting in better optimization of the network.

- *a stereo feature fusion network, Progressive Fusion Network (PFN)*, in the context of binocular stereo depth estimation we developed a general multi-scale refinement network that combines information from both stereo views. In principle, it can be applied to any multi-scale application for feature sharing, independently of the supervised or unsupervised scenario. We proved in our experimental section its effectiveness for depth estimation.

- *a coupled Generative Adversarial Network to better structure the loss before backpropagation*, we propose to couple the adversarial learning with the image reconstruction task in a structured way, such that, both tasks can benefit each other. Which, is obtained during training, by fusing generator and discriminator predictions and errors with a convolutional Conditional Random Field (CRF).

- *a model capable of self-refining the predictions and self-improving itself*, we revisit the cycled network for monocular depth estimation and devise a student-teacher model. By exploiting student model errors, the teacher model refines the predictions to obtain better accuracy. Furthermore, we apply knowledge distillation from the teacher model to the student model to improve the latter one.

1.4 Publications

The following list gives an overview of the publications included in this thesis in chronological order, note that a few of these works (marked with *) are not included in this thesis

- "Viraliency: Pooling Local Virality", X. Alameda-Pineda, A. Pilzer, D. Xu, N. Sebe, E. Ricci; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6080-6088.*
- "Unsupervised Adversarial Depth Estimation Using Cycled Generative Networks", A. Pilzer, D. Xu, M. Puscas, E. Ricci and N. Sebe; 2018 International Conference on 3D Vision (3DV), Verona, 2018, pp. 587-595.
- "Refine and Distill: Exploiting Cycle-Inconsistency and Knowledge Distillation for Unsupervised Monocular Depth Estimation", A. Pilzer, S. Lathuilière, N. Sebe, E. Ricci; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9768-9777.
- "Online Adaptation through Meta-Learning for Stereo Depth Estimation", Z. Zhang, S. Lathuilière, A. Pilzer, N. Sebe, E. Ricci, J. Yang; arXiv 1904.08462.*
- "Structured Coupled Generative Adversarial Networks for Unsupervised Monocular Depth Estimation", M. M. Puscas, D. Xu, A. Pilzer and N. Sebe; 2019 International Conference on 3D Vision (3DV), Québec City, QC, Canada, 2019, pp. 18-26.
- "Progressive Fusion for Unsupervised Binocular Depth Estimation using Cycled Networks", A. Pilzer, S. Lathuilière, D. Xu, M. M. Puscas, E. Ricci and N. Sebe; IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI: 10.1109 / TPAMI .2019 .2942928.

2

Related Work

In this Chapter we focus on related works, mainly about depth estimation. More in detail, we will discuss those that were important for our research. In Section 2.1, we discuss the benchmark datasets for depth estimation, we will then move to describe relevant depth estimation works with supervised learning in Section 2.2. In Section 2.3, we introduce unsupervised depth estimation. Finally, in Section 2.4 and in Section 2.5 we give an overview of adversarial learning and knowledge distillation, respectively.

2.1 Datasets

Supported by multiple public datasets, depth estimation made major advancements in recent years. Some of them are monocular (*i.e.* each image is a view of a different scene), as [15, 16], while others are stereo datasets (*e.g.* a stereo binocular camera setting with two cameras is used, two views of the same scene are available). NYUD [15] is a monocular dataset recorded indoors and depicts several different rooms (*e.g.* kitchen, office, etc.) while Make3D [16] is composed of outdoor scenes. These two datasets are relatively small having a total number of images in the order of few hundreds. Binocular stereo examples are KITTI [17], CityScapes [18] and ApolloScape [19], these datasets are recorded from cars driving around several cities, the first two in Germany and the third in China. These dataset have much larger scale and are composed by thousands of images with annotations. Annotations are also more extensive covering several different tasks. For example CitiScapes has fine grained annotations for semantic segmentation, while KITTI contains more annotation for depth estimation and stereo matching acquired with a LIDAR. ApolloScape has annotations for the aforementioned tasks and in addition lane

segmentation, trajectory and tracking. With the same spirit of generalizing to several computer vision problems NuScenes [20] dataset was released in 2019. With more than 1.4 million images it is much bigger than the previous and has annotations for depth, semantic segmentation, 3D object detection. Another difference is the stereo camera setting with 6 cameras that guarantee a 360 degree view around the car.

2.2 Supervised Depth Estimation

Literature in the field of computer vision and more recently of deep learning based computer vision demonstrated that is possible to achieve outstanding results with supervised learning. Nowadays, supervised learning requires large amounts of labelled data to feed to deep neural networks. More precisely, in the case of depth estimation, the deep neural network will regress from the input image pixels their distance from the camera.

Several monocular depth estimation methods have been proposed [3, 10, 21, 22, 23] and they proved to be very effective given enough training data. Eigen *et al.* [3] proposed a convolutional neural network (ConvNet) demonstrating the benefit of exploiting local and global information. Laina *et al.* [23] build on the previous findings to propose a deeper ConvNet architecture. Fu *et al.* [24] greatly improve performance by casting the continuous depth estimation problem as a discretized depth classification problem. Probabilistic graphical models, to better exploit structural information of the scene, were integrated into deep end-to-end ConvNets by [10, 21, 25, 26]. Wang *et al.* [25] proposed hierarchical CRFs for joint depth and semantic segmentation estimation. Xu *et al.* [26] combined multi-scale features with Conditional Random Fields (CRFs). Further improving it in [10] with the introduction of an attention mechanism for the feature fusion.

Other works use ConvNets to learn supervised stereo matching. Chang *et al.* [27] proposed a pyramid pooling model to learn global context aware features. [28] propose residual prediction refinement, while [29] a MAP criterion with sub-pixel precision to address ambiguous stereo matching. More recently pyramid models have been revisited in Yin *et al.* [30] for probabilistic based hierarchical stereo matching.

Tackling the growing variety of datasets and applications, Tonioni *et al.* [31] proposed a method for real-time depth estimation with adaptation to novel environments. Another approach is [32] where a model is initially pretrained on synthetic data and successively trained on the target domain.

2.3 Unsupervised Depth Estimation

Large annotated dataset are the bottleneck for supervised learning. For this reason self-supervised or unsupervised depth estimation emerged and quickly reached comparable results as supervised methods. Unsupervised methods exploit the geometry of the scene in binocular stereo images [4, 5, 33, 34] or in consecutive video frames [35, 36, 37]. The first, compute the dense correspondence map (*i.e.* disparity) between the stereo frames, whereas the latter reproject successive temporal frames by estimating camera parameters and relative pose to infer the depth. The depth map D can be calculated from the disparity map d through $D = fB/d$, where f is the cameras focal length and B is the baseline distance between two stereo cameras.

Garg *et al.* [4] proposed to take in input the left stereo view and estimate the disparity with a CNN. Then through a warping operation reconstruct the left stereo view from the right stereo view using the disparity. They minimize the reconstruction error on the left stereo view, thus not requiring explicit depth ground truth but just rectified stereo image pairs. Shortly after Godard *et al.* [5] building on top of [4] proposed ot estimate both the right-to-left disparity, aligned with the left input stereo view, and the left-to-right disparity, aligned with the right stereo view. They also propose an additional structured loss and a disparity consistency objective to further constrain the model. In [9] instead they focus on better feature representation, with 3D convolutions. Their method is effective but has the drawback of increased computational cost due to the 3D operations. In [35] they propose to use monocular video sequences and, based on the reconstruction loss among successive frames together with an occlusion aware criterion they are able to optimize a model for depth and pose estimation. Zhan *et al.* [34] add the temporal constraint further improving depth estimation by joint visual odometry and depth learning. A similar intuition was followed by Godard *et al.* [38] where they jointly estimate pose and depth of the scene, propose to use an occlusion aware loss to obtain better convergence and estimations. An orthogonal approach was proposed by Lai *et al.* [39] where they successfully use an optical flow model for guaranteeing temporal consistency, alongside an unsupervised depth estimation model between binocular stereo views.

Domain adaptation for unsupervised depth estimation, exploiting synthetic data for training, following the idea of [32] was proposed by Kundu *et al.* [40]. They propose an adversarial unsupervised adaptation setup for high dimensional features. Successively in [41] they implement joint training from real and synthetic data, while Zhao *et al.* [42] take advantage of the CycleGAN [7] approach to augment training data for their self-supervised model.

2.4 Adversarial Learning

Generative Adversarial Networks (GAN) were first proposed by Goodfellow *et al.* [6] as a novel approach for image generation. They propose to learn an image generator from a noise vector to fool a discriminator model that has to discriminate if the image is real or generated. These two models are trained to compete with each other, on one side the generator should improve the quality of generated images while on the other the discriminator has to become better in discriminating fakes, thus the name of adversarial learning. Since then, the research community worked hard to improve Goodfellow's approach generation quality and training stability. Some works concentrate on improving the model architecture, as in Radford *et al.* [43] where they propose a deep convolutional architecture. While others introduce class supervision or conditioning as a further constrain [44, 45]. Another research direction is to improve the optimization objective, with Wasserstein distance in [46], later with Cramer distance in [47] or more simply with least squares distance as proposed by [48]. Several computer vision applications employ GANs, some examples are image-to-image translation [7, 8] where an image is transformed into a desired domain image, generation of images in different poses [49] and image animation [50].

2.5 Distillation

Since the breakthrough of deep learning with AlexNet [1] on the image classification benchmark ImageNet [2] deep ConvNets grew larger and larger to increase performance. Model distillation aims to train a small *student* model, by small lower capacity (*i.e.* number of parameters) and computations requirements has to be intended, imitating a large *teacher* network. This idea, applied to deep learning, was first proposed by Hinton *et al.* [51] by adding to the traditional discriminative training objective of the student model an additional term. This term aims to make the logit prediction of the student similar to those of the teacher. In [52] they exploit distillation to obtain faster training, whereas in [53] they improve the optimization objective by using attention. The distillation principle found several applications in computer vision as domain adaptation [54], learning from noisy labels [55], face recognition [56] and lane detection [57].

3

Datasets and Evaluation

We will now detail the evaluation metrics used throughout this thesis and the training and evaluation dataset splits used in the different chapters.

3.1 Evaluation Metrics

To quantitatively evaluate the proposed approaches, we follow several standard evaluation metrics used in previous works [3, 5, 25]. Let P be the total number of pixels in the test set and $\hat{d}_i$, d_i the estimated depth and ground truth depth values for pixel i. We compute the following metrics:

- Mean relative error (abs rel): $\frac{1}{P}\sum_{i=1}^{P}\frac{\|\hat{d}_i - d_i\|}{d_i}$,
- Squared relative error (sq rel): $\frac{1}{P}\sum_{i=1}^{P}\frac{\|\hat{d}_i - d_i\|^2}{d_i}$,
- Root mean squared error (rmse): $\sqrt{\frac{1}{P}\sum_{i=1}^{P}(\hat{d}_i - d_i)^2}$,
- Mean $\log 10$ error (rmse log): $\sqrt{\frac{1}{P}\sum_{i=1}^{P} \parallel \log \hat{d}_i - \log d_i \parallel^2}$
- Accuracy with threshold τ, *i.e.* the percentage of $\hat{d}_i$ such that $\delta = \max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) < \alpha^{\tau}$. We employ $\alpha = 1.25$ and $\tau \in [1, 2, 3]$ following [3].

Even though, in Chapters 4 and partially 5, the models perform inference from stereo images, we chose to align our evaluation with the depth from monocular images metrics. This is motivated, on one hand, by our goal to devise models to perform inference from monocular, as in Chapter 6 and as outlined in the proposed future works Section 7.1. On the other hand, it allows for a better comparative performance among all the methods presented in this thesis.

3.2 Evaluation Datasets

In Section 2.1 we introduced the main datasets used nowadays for depth estimation. Now we will detail the training and testing splits used in the following chapters. All the proposed methods will be evaluated on two large stereo images dataset KITTI and CityScapes. Both datasets are recorded from driving vehicles through several German cities, during different times of the day and seasons.

For the **KITTI** dataset, we use the Eigen split [3] for training and testing. This split contains 22,600 training image pairs, and 697 test pairs. We do data augmentation with online random horizontal flipping of the images during training. Note that when flipped, the images are also swapped, to guarantee they are in correct position relative to each other. We added also random colour augmentations, namely gamma, brightness and color distributions. The input images are down-sampled to a resolution of 512×256 from 1226×370.

The **Cityscapes** dataset presents higher resolution images and is annotated mainly for semantic segmentation. To train our model we combine the densely and coarse annotated splits to obtain 22,973 image-pairs. For testing we use the 1,525 image-pairs of the densely annotated split. The test set also has pre-computed disparity maps for the evaluation. During training, the bottom one fifth of the image is cropped following [5] and then is resized to 512×256. Similarly to KITTI we perform data augmentation on-the-fly with the same settings during training.

Furthermore, in Section 4.2, we add additional experiments on **ApolloScape** dataset. It has been collected from a stereo camera attached to a car driving in different Chinese cities. To the best of our knowledge, we are the first to benchmark depth estimation methods on the ApolloScape dataset. We employ two sequences from the *Scene Parsing* data split, scene *road02* and *road03*, obtaining 9156 training image pairs and 2186 testing image pairs. Note that, the other sequences use varying setting stereo camera settings and, as a consequence, cannot be used easily for depth estimation. The dataset provides dense depth ground-truth for all the images. At training time, we applied the same online random augmentations that we applied to KITTI dataset.

4

Stereo Adversarial Depth Estimation

In this Chapter we will present our research on learning from resynthesized data and adversarial learning applied to stereo unsupervised depth estimation. In Section 4.1, we devise a adversarial optimization objective for a dense prediction task as depth estimation. After that, we propose a cycled network for data augmentation. The cycled structure allows us to learn from both original and resynthesized images. We show in experimental results that the proposed ideas benefit model performance. The extension of this work is presented in Section 4.2, where we devise a multiscale feature refinement strategy, called Progressive Fusion Network, with the goal of reducing training times and model parameters. Moreover, we extensively benchmark this approach against state-of-the-art on large publicly available datasets.

4.1 Unsupervised Adversarial Depth Estimation using Cycled Generative Network [1]

While recent deep monocular depth estimation approaches based on supervised regression have achieved remarkable performance, costly ground truth annotations are required during training. To cope with this issue, in this Section we present a novel unsupervised deep learning approach for predicting depth maps and show that the depth estimation task can be effectively tackled within an adversarial learning framework. Specifically, we propose a deep generative network that learns to predict the correspondence field (*i.e.* the disparity map) between two image

[1] "Unsupervised Adversarial Depth Estimation Using Cycled Generative Networks", A. Pilzer, D. Xu, M. Puscas, E. Ricci and N. Sebe; 2018 International Conference on 3D Vision (3DV), Verona, 2018, pp. 587-595

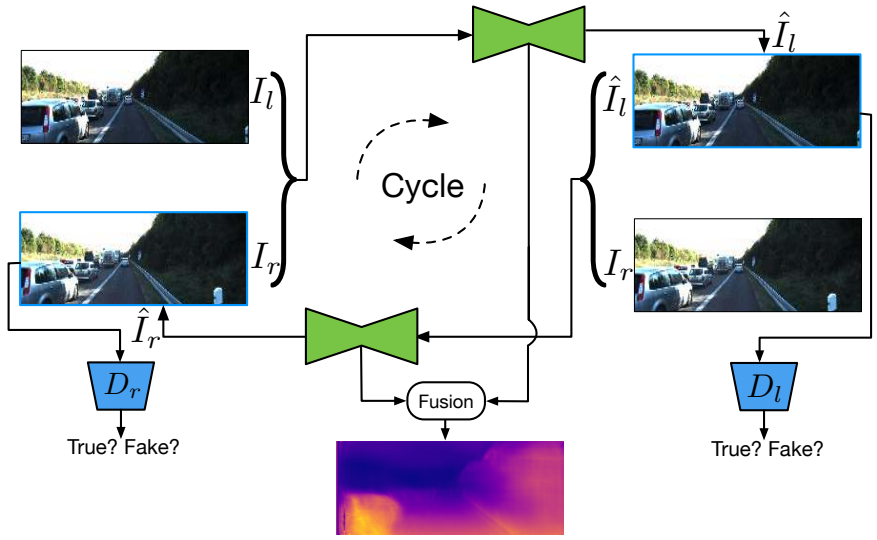


Figure 4.1: Motivation of the proposed unsupervised depth estimation approach using cycled generative networks optimized with adversarial learning. The generator model takes two stereo image views as input and resynthesizes $\hat{\mathbf{I}}_l$. Then, as in a cycle, the model resynthesizes $\hat{\mathbf{I}}_r$. The left and right image synthesis in a cycle provides strong constraint and supervision to better optimize both generators. Final depth estimation is obtained by fusing the output from both generators.

views in a calibrated stereo camera setting. The proposed architecture consists of two generative sub-networks jointly trained with adversarial learning for reconstructing the disparity map and organized in a cycle such as to provide mutual constraints and supervision to each other. Extensive experiments on the publicly available datasets KITTI and Cityscapes demonstrate the effectiveness of the proposed model and competitive results with state of the art methods.

4.1.1 Introduction

As one of the fundamental problems in computer vision, depth estimation has received a substantial interest in the past, also motivated by its importance in various application scenarios, such as robotics navigation, 3D reconstruction, virtual reality and autonomous driving. Over the last few years the performances of depth estimation methods have been significantly improved thanks to advanced deep learning techniques.

Most previous works considering deep architectures for predicting depth maps operate in a supervised learning setting [3, 10, 21, 58] and, specifically, devise powerful deep regression models with Convolutional Neural Networks (CNN). These models are used for monocular depth estimation, *i.e.* they are trained to learn the transformation from the RGB image domain

to the depth domain in a pixel-to-pixel fashion. In this context, multi-scale CNN models have shown to be especially effective for estimating depth maps [3]. Upon these, probabilistic graphical models, such as Conditional Random Fields (CRFs), implemented as neural networks for end-to-end optimization, have proved to be beneficial, boosting the performance of deep regression models [10, 21]. However, supervised learning models require ground-truth depth data which are usually costly to acquire. This problem is especially relevant with deep learning architectures, as large amount of data are typically required to produce satisfactory performance. Furthermore, supervised monocular depth estimation can be regarded as an ill-posed problem due to the scale ambiguity issue [59].

To tackle these problems, recently unsupervised learning-based approaches for depth estimation have been introduced [60, 61]. These methods operate by learning the correspondence field (*i.e.* the disparity map) between the two different image views of a calibrated stereo camera using only the rectified left and right images. Then, given several camera parameters, the depth maps can be calculated using the predicted disparity maps. Significant progresses have been made along this research line [4, 5, 62]. In particular, Godard *et al.* [5] proposed to estimate both the direct and the reverse disparity maps using a single generative network and utilized the consistency between left and right disparity maps to constrain on the model learning. Other works proposed to facilitate the depth estimation by jointly learning the camera pose [35, 36]. These works optimized their models relying on the supervision from the image synthesis of an expected view, whose quality plays a direct influence on the performance of the estimated disparity map. However, all of these works only considered a reconstruction loss and none of them have explored using adversarial learning to improve the generation of the synthesized images.

We follow the unsupervised learning setting and propose a novel end-to-end trainable deep network model for adversarial learning-based depth estimation given stereo image pairs. The proposed approach consists of two generative sub-networks which predict the disparity map from the left to the right view and viceversa. The two sub-networks are organized in a cycle (Fig. 4.1), such as to perform the image synthesis of different views in a closed loop. This new network design provides strong constraint and supervision for each image view, facilitating the optimization of both generators from the two sub-networks which are jointly learned with an adversarial learning strategy. The final disparity map is produced by combining the output from the two generators.

In summary, the main contributions of this Section are threefolds:

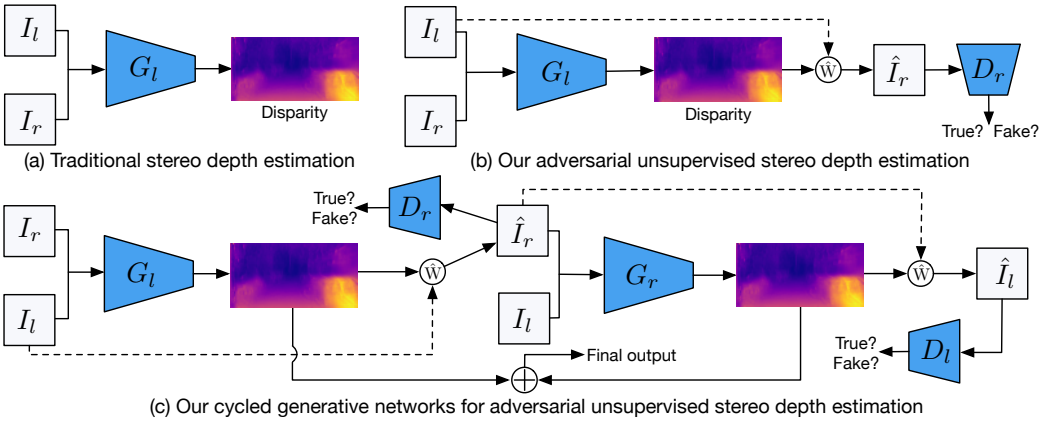


Figure 4.2: An illustrative comparison of different methods for unsupervised stereo depth estimation: (a) traditional supervised stereo-matching-based depth estimation, (b) the proposed unsupervised adversarial depth estimation and (c) the proposed cycled generative networks for unsupervised adversarial depth estimation. The symbols D_l, D_r denote discriminators, and G_l, G_r denote generators. The symbol $\hat{\mathrm{W}}$ denotes a warping operation.

- To the best of our knowledge, we are the first to explore using adversarial learning to facilitate the image synthesis of different views in a unified deep network for improving the unsupervised depth estimation;
- We present a new cycled generative network structure for unsupervised depth estimation which can learn both the forward and the reverse disparity maps, and can synthesize the different image views in a closed loop. Compared with the existing generative network structures, the proposed cycled generative network is able to enforce stronger constraints from each image view and better optimize the network generators.
- Extensive experiments on two large publicly available datasets (*i.e.* KITTI and Cityscapes) demonstrate the effectiveness of both the adversarial image synthesis and the cycled generative network structure.

4.1.2 Proposed Approach

We propose a novel approach for unsupervised adversarial depth estimation using cycled generative networks. An illustrative comparison of different unsupervised depth estimation models is shown in Fig. 4.2. Fig. 4.2a shows traditional stereo matching based depth estimation approaches, which basically learn a stereo matching network for directly predicting the disparity [60]. Different from the traditional stereo approaches, we estimate the disparity in an

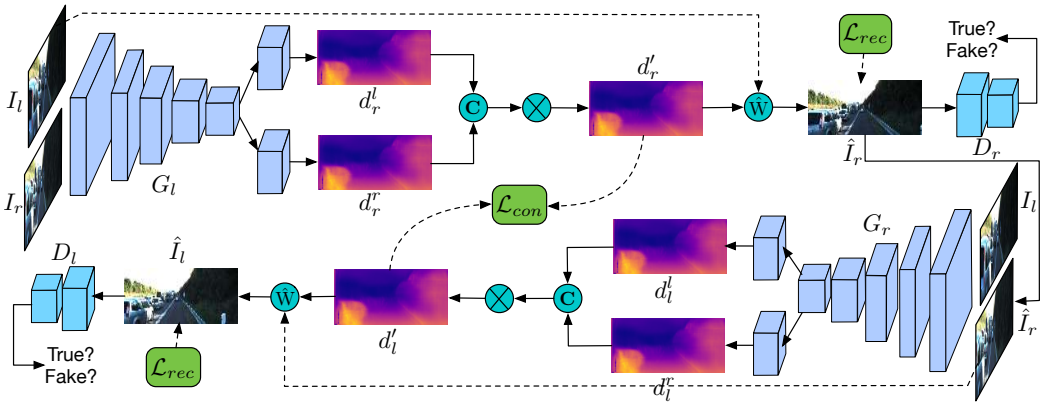


Figure 4.3: Illustration of the detailed framework of the proposed cycled generative networks for unsupervised adversarial depth estimation. It is based on an encoder-decoder structure that processes separately the two stereo views. The symbol © denotes a concatenation operation; $\times$ denotes a learned weighted average, implemented as a 1×1 convolution; $\mathcal{L}_{rec}$ represents the reconstruction loss for different generators; $\mathcal{L}_{con}$ denotes a consistency loss between the disparity maps generated from the two generators.

indirect means through image synthesis from different views with the adversarial learning strategy as shown in Fig. 4.2b. Fig. 4.2c shows our full model using the proposed cycled generative networks for the task. In this section we first give the problem statement, and then present the proposed adversarial learning-based unsupervised stereo depth estimation, and finally we illustrate the proposed full model and introduce the overall end-to-end optimization objective and the testing process.

4.1.2.1 Problem Statement

We target at estimating a disparity map given a pair of images from a calibrated stereo camera. The problem can be formally defined as follows: given a left image $\mathbf{I}_l$ and a right image $\mathbf{I}_r$ from the camera, we are interested in predicting a disparity map $\mathbf{d}$ in which each pixel value represents an offset of the corresponding pixel between the left and the right image. If given the baseline distance b_d between the left and the right camera and the camera focal length f_l, a depth map $\mathbf{D}$ can be calculated with the formula of $\mathbf{D} = (b_d * f_l)/\mathbf{d}$. We indirectly learn the disparity through the image synthesis. Specifically, assume that a left-to-right disparity $\mathbf{d}_r^{(l)}$ is produced from a generative network G_l with the left-view image $\mathbf{I}_l$ as input, and then a warping function $f_w(\cdot)$ is used to perform the synthesis of the right image view by sampling

from $\mathbf{I}_l$, *i.e.* $\hat{\mathbf{I}}_r = f_w(\mathbf{d}_r^{(l)}, \mathbf{I}_l)$. A reconstruction loss between $\hat{\mathbf{I}}_r$ and $\mathbf{I}_r$ is thus utilized to provide supervision in optimizing the network G_l.

4.1.2.2 Unsupervised Adversarial Depth Estimation

We now introduce the proposed unsupervised adversarial depth estimation approach. Assuming we have a generative network G_l composed of two sub-networks, a generative sub-network $G_l^{(l)}$ with input $\mathbf{I}_l$ and a generative sub-network $G_l^{(r)}$ with input $\mathbf{I}_r$. These are used to produce two distinct left-to-right disparity maps $\mathbf{d}_r^{(l)}$ and $\mathbf{d}_r^{(r)}$ respectively, *i.e.* $\mathbf{d}_r^{(l)} = G_l^{(l)}(\mathbf{I}_l)$ and $\mathbf{d}_r^{(r)} = G_l^{(r)}(\mathbf{I}_r)$. The sub-network $G_l^{(l)}$ and $G_l^{(r)}$ exploit the same network structure using a convolutional encoder-decoder, where the encoders aim at obtaining compact image representations and could be shared to reduce the network capacity. Since the two disparity maps are produced from different input images, and show complementary characteristics, they are fused using a linear combination implemented as concatenation and 1×1 convolution, and we obtain an enhanced disparity map $\mathbf{d}'_r$, which is used to synthesize a right view image $\hat{\mathbf{I}}_r$ via the warping operation, *i.e.* $\hat{\mathbf{I}}_r = f_w(\mathbf{d}'_r, \mathbf{I}_l)$. Then we use an $L1$-norm reconstruction loss $\mathcal{L}_{rec}$ for optimization as follows:

$$\mathcal{L}_{rec}^{(r)} = \|\mathbf{I}_r - f_w(\mathbf{d}'_r, \mathbf{I}_l)\|_1 \tag{4.1}$$

To improve the generation quality of the image $\hat{\mathbf{I}}_r$ and benefit from the advantage of adversarial learning, we propose to use adversarial learning here for a better optimization due to its demonstrated powerful ability in the image generation task [6]. For the synthesized image $\hat{\mathbf{I}}_r$, a discriminators D_r outputting a scalar value which is used to discriminate if the image $\hat{\mathbf{I}}_r$ or $\mathbf{I}_r$ is fake or true, and thus the adversarial objective for the generative network can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{gan}^{(r)}(G_l, D_r, \mathbf{I}_l, \mathbf{I}_r) = \mathbb{E}_{\mathbf{I}_r \sim p(\mathbf{I}_r)}[\log D_r(\mathbf{I}_r)] \\ + \mathbb{E}_{\mathbf{I}_l \sim p(\mathbf{I}_l)}[\log(1 - D_r(f_w(\mathbf{d}'_r, \mathbf{I}_l)))]\end{aligned} \tag{4.2}$$

where we adopt a cross-entropy loss to measure the expectation of the image $\mathbf{I}_l$ and $\mathbf{I}_r$ against the distribution of the left and the right view images $p(\mathbf{I}_l)$ and $p(\mathbf{I}_r)$ respectively. Then the joint optimization loss is the combination of the reconstruction loss and the adversarial loss written as:

$$\mathcal{L}_o^{(r)} = \gamma_1 \mathcal{L}_{rec}^{(r)} + \gamma_2 \mathcal{L}_{gan}^{(r)} \tag{4.3}$$

where γ_1 and γ_2 are the weights for balancing the loss magnitude of the two parts to stabilize the training process. In the testing phase, the inferred $\mathbf{d}'_r$ is the final output.

4.1.2.3 Cycled Generative Networks for Adversarial Depth Estimation

In the previous section, we presented the adversarial learning-based depth estimation approach which reconstructs from one image view to the other one in a straightforward way. In order to make the image reconstruction from different views implicitly constrain on each other, we further propose a cycled generative network structure. An overview of the proposed network structure is shown in Fig. 4.3. The network produces two distinct disparity maps from different view directions, and synthesizes different-view images in a closed loop. In our network design, not only the different view reconstruction loss helps for better optimization of the generators, but also the two disparity maps are connected with a consistence loss to provide strong supervision from each half cycle.

We described the half-cycle generative network with adversarial learning in Section 4.1.2.2. The cycled generative network is based on the half-cycle structure. To simplify the description, we follow the notations used in Section 4.1.2.2. Assume we have obtained a synthesized image $\hat{\mathbf{I}}_r$ from the half-cycle network, and then $\hat{\mathbf{I}}_r$ is further used as input of the next cycle generative network. Let us denote the generator as G_r, which we exploit the encoder-decoder network structure similar as G_l in Sec. 4.1.2.2. The encoder part of G_r can be also shared with the encoder of G_l to have a more compact network model (we show the performance difference between using and not using the sharing scheme), and the two distinct decoders are used to produce two right-to-left disparity maps $\mathbf{d}_l^{(l)}$ and $\mathbf{d}_l^{(r)}$ corresponding the left- and the right-view input images respectively. The two maps are also combined with the combination and the convolution operation to have a fused disparity map $\mathbf{d}'_l$. Then we synthesize the left-view image $\hat{\mathbf{I}}_l$ via the warping operation as $\hat{\mathbf{I}}_l = f_w(\mathbf{d}'_l, \mathbf{I}_r)$. An $L1$-norm reconstruction loss is used for optimizing the generator G_r. Then the objective for optimizing the two generators of the full cycle writes

$$\mathcal{L}_{rec}^{(f)} = \|\mathbf{I}_r - f_w(\mathbf{d}'_r, \mathbf{I}_l)\|_1 + \|\mathbf{I}_l - f_w(\mathbf{d}'_l, \hat{\mathbf{I}}_r)\|_1 \tag{4.4}$$

We add a discriminator D_l for discriminating the synthesized image $\hat{\mathbf{I}}_l$, and then the adversarial learning strategy is used for both the left and the right image views in a closed loop. The adversarial objective for the full cycled model can be formulated as

$$\begin{aligned} &\mathcal{L}_{gan}^{(f)}(G_l, G_r, D_r, \mathbf{I}_l, \mathbf{I}_r) = \mathbb{E}_{\mathbf{I}_r \sim p(\mathbf{I}_r)}[\log D_r(\mathbf{I}_r)] \\ &+\mathbb{E}_{\mathbf{I}_l \sim p(\mathbf{I}_l)}[\log(1 - D_r(f_w(\mathbf{d}'_r, \mathbf{I}_l)))] + \mathbb{E}_{\mathbf{I}_l \sim p(\mathbf{I}_l)}[\log D_l(\mathbf{I}_l)] \\ &+\mathbb{E}_{\mathbf{I}_r \sim p(\mathbf{I}_r)}[\log(1 - D_l(f_w(\mathbf{d}'_l, \hat{\mathbf{I}}_r)))] \end{aligned} \tag{4.5}$$

Each half of the cycle network produces a disparity map corresponding to a different view translation, *i.e.* $\mathbf{d}'_l$ and $\mathbf{d}'_r$. To make them constrain on each other, we add an $L1$-norm consistence loss between these two maps as follows:

$$\mathcal{L}_{con}^{(f)} = ||\mathbf{d}'_l - f_w(\mathbf{d}'_l, \mathbf{d}'_r)||_1 \tag{4.6}$$

where since the two disparity maps are for different views and are not aligned, we use the warping operation to make them pixel-to-pixel matched. The consistence loss put a strong view constraint for each half cycle and thus facilitates the learning of both half cycles.

Full objective. The full optimization objective consists of the reconstruction losses of both generators, the adversarial losses for both view synthesis and the half-cycle consistence loss. It can be written as follows:

$$\mathcal{L}_{o}^{(f)} = \gamma_1 \mathcal{L}_{rec}^{(f)} + \gamma_2 \mathcal{L}_{gan}^{(f)} + \gamma_3 \mathcal{L}_{con}^{(f)}. \tag{4.7}$$

Where $\{\gamma_i\}_{i=1}^{3}$ represents a set of weights for controlling the importance of different optimization parts.

Inference. When the optimization is finished, given a testing pair $\{\mathbf{I}_l, \mathbf{I}_r\}$, the testing is performed by combining the output disparity maps $\mathbf{d}'_l$ and $\mathbf{d}'_r$ in a weighted averaging scheme. We treat the two half cycles with equal importance, and the final disparity map $\mathbf{D}$ is obtained as the mean of the two, *i.e.* $D = (\mathbf{d}'_l + f_w(\mathbf{d}'_l, \mathbf{d}'_r))/2$.

4.1.2.4 Network Implementation Details

To describe the details of the network implementation, in terms of the generators G_l and G_r, we use a ResNet-50 backbone network for the encoder part, and the decoder part contains five deconvolution with ReLU operations in which each 2 times up-samples the feature map. The skip connections are also used to pass information from the backbone representations to the deconvolutional feature maps for obtaining more effective feature aggregation. For the discriminators D_l and D_r, we employ the same network structure which has five consecutive convolutional operations with a kernel size of 3, a stride size of 2 and a padding size of 1, and batch normalization [63] is performed after each convolutional operation. Adversarial loss is applied to output patches. For the warping operation, a bilinear sampler is used as in [5].

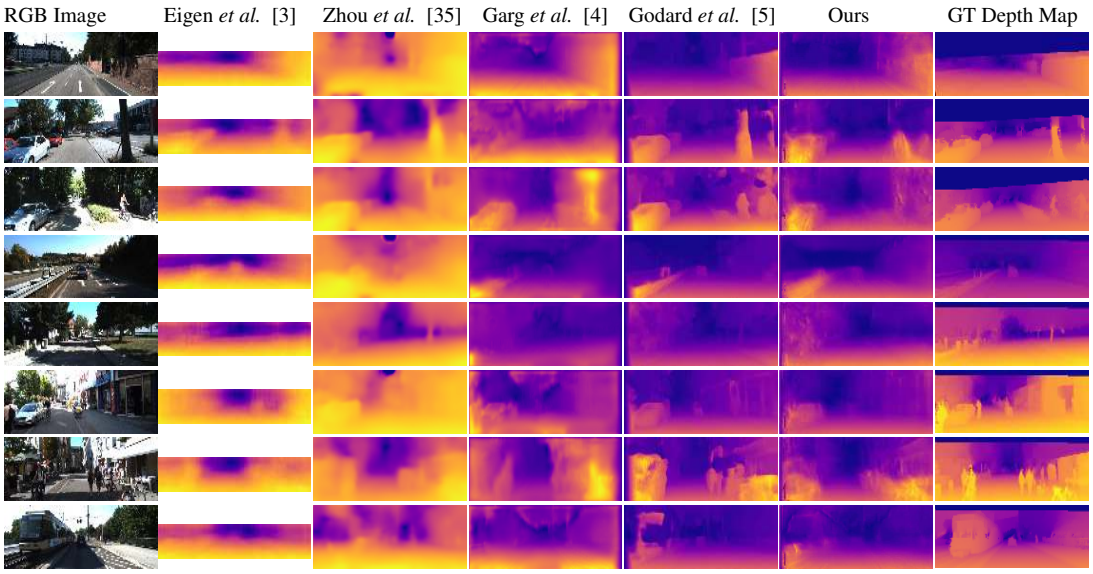


Figure 4.4: Qualitative comparison with different competitive approaches with both supervised and unsupervised settings on the KITTI test set. The sparse groundtruth depth maps are filled with bilinear interpolation for better visualization.

4.1.3 Experimental Results

We present both qualitative and quantitative results on publicly available datasets to demonstrate the performance of the proposed approach for unsupervised adversarial depth estimation.

4.1.3.1 Experimental Setup

Datasets and Evaluation We carry out experiments on two large datasets, *i.e.* KITTI [17] and Cityscapes [18] . We detailed in Chapter 3 the dataset splits and preprocessing used. Similarly, the evaluation metrics have beed throughly described in Chapter 3.

Parameter Setup. The proposed model is implemented using the deep learning library *TensorFlow* [64]. The input images are down-sampled to a resolution of 512×256 from 1226×370 in the case of the KITTI dataset, while for the Cityscapes dataset, at the bottom one fifth of the image is cropped following [5] and then is resized to 512×256. The output disparity maps from two input images are fused with a learned linear combination to obtain the final disparity map with a size 512×256. The batch size for training is set to 8 and the initial learning rate is 10^{-5} in all the experiments. We use the Adam optimizer for the optimization. The momentum parameter and the weight decay are set to 0.9 and 0.0002, respectively. The final optimization

Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	lower is better				higher is better		
Half-Cycle Mono	0.240	4.264	8.049	0.334	0.710	0.871	0.937
Half-Cycle Stereo	0.228	4.277	7.646	0.318	0.748	0.892	0.945
Half-Cycle + D	0.211	2.135	6.839	0.314	0.702	0.868	0.939
Full-Cycle + D	0.198	**1.990**	**6.655**	**0.292**	0.721	0.884	0.949
Full-Cycle + D + SE	**0.190**	2.556	6.927	0.353	**0.751**	**0.895**	**0.951**

Table 4.1: Quantitative evaluation results of different variants of the proposed approach on the KITTI dataset for the ablation study. We do not perform cropping on the depth maps for evaluation and the estimated depth range is from 0 to 80 meters.

objective has weighed loss parameters $\gamma_1 = 1$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$. The learning rate is reduced by half at both $[80k, 100k]$ steps. For our experiments we used an NVIDIA Tesla K80 with 12 GB of memory.

Detailed Training Procedure. We train the half-cycle model with a standard training procedure, *i.e.* initializing the network with random weights and making the network train for a full 50 epochs. For the cycled model we optimize the network with an iterative training procedure. After random weights initialization, we train the first half branch $\{\mathbf{I}_l, \mathbf{I}_r\} \rightarrow \hat{\mathbf{I}}_r$, with generator G_l and discriminator D_r for a 20k iteration steps. After that we train the second half branch $\{\hat{\mathbf{I}}_r, \mathbf{I}_l\} \rightarrow \hat{\mathbf{I}}_l$ with generator G_r and discriminator D_l for another 20k iterations. For the training of the first cycle branch, we do not use the cycle consistence loss since the second half branch is not trained yet. Finally we jointly train the whole network with all the losses embedded for a final round of 100k iterations.

4.1.3.2 Ablation Study

To validate the adversarial learning strategy is beneficial for the unsupervised depth estimation, and the proposed cycled generative network is effective for the task, we present an extensive ablation study on both the KITTI dataset (see Table 4.1) and on the Cityscape dataset (see Table 4.3).

Baseline Models. We have several baseline models for the ablation study, including (i) Half-cycle with a monocular setting (half-cycle mono), which uses a straight forward branch to synthesize from one image view to the other with a single disparity map output and the single RGB image is as input during testing; (ii) half-cycle with a stereo setting (half-cycle stereo),

Method	Sup	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
		lower is better				higher is better		
Saxena *et al.* [59]	Y	0.280	-	8.734	-	0.601	0.820	0.926
Eigen *et al.* [3]	Y	0.190	1.515	7.156	0.270	0.692	0.899	0.967
Liu *et al.* [21]	Y	0.202	1.614	6.523	0.275	0.678	0.895	0.965
AdaDepth [40], 50m	Y	0.162	1.041	4.344	0.225	0.784	0.930	0.974
Kuznietzov *et al.* [33]	Y	-	-	4.815	0.194	0.845	0.957	0.987
Xu *et al.* [10]	Y	0.132	0.911	-	0.162	0.804	0.945	0.981
Zhou *et al.* [35]	N	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Garg *et al.* [4]	N	0.169	1.08	5.104	0.273	0.740	0.904	0.962
AdaDepth [40], 50m	N	0.203	1.734	6.251	0.284	0.687	0.899	0.958
Godard *et al.* [5]	N	**0.148**	**1.344**	**5.927**	**0.247**	**0.803**	**0.922**	**0.964**
Ours	N	0.166	1.466	6.187	0.259	0.757	0.906	0.961
Ours with shared enc	N	0.152	1.388	6.016	0.247	0.789	0.918	0.965
Ours, 50m	N	0.158	1.108	4.764	0.245	0.771	0.915	0.966
Ours with shared enc, 50m	N	**0.144**	**1.007**	**4.660**	**0.240**	**0.793**	**0.923**	**0.968**

Table 4.2: Comparison with state of the art. Training and testing are performed on the KITTI [17] dataset. Supervised and semi-supervised methods are marked with Y in the supervision column, unsupervised methods with N. Numbers are obtained on Eigen test split with Garg image cropping. Depth predictions are capped at the common threshold of 80 meters, if capped at 50 meters we specify it.

Method	Sup	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
		lower is better				higher is better		
Half-Cycle Mono	N	0.467	7.399	5.741	0.493	0.735	0.890	0.945
Half-Cycle Stereo	N	0.462	6.097	5.740	0.377	0.708	0.873	0.937
Half-Cycle + D	N	**0.438**	**5.713**	5.745	0.400	0.711	0.877	0.940
Full-Cycle + D	N	0.440	6.036	**5.443**	**0.398**	**0.730**	**0.887**	**0.944**

Table 4.3: Quantitative evaluation results of different variants of the proposed approach on the Cityscapes dataset for the ablation study.

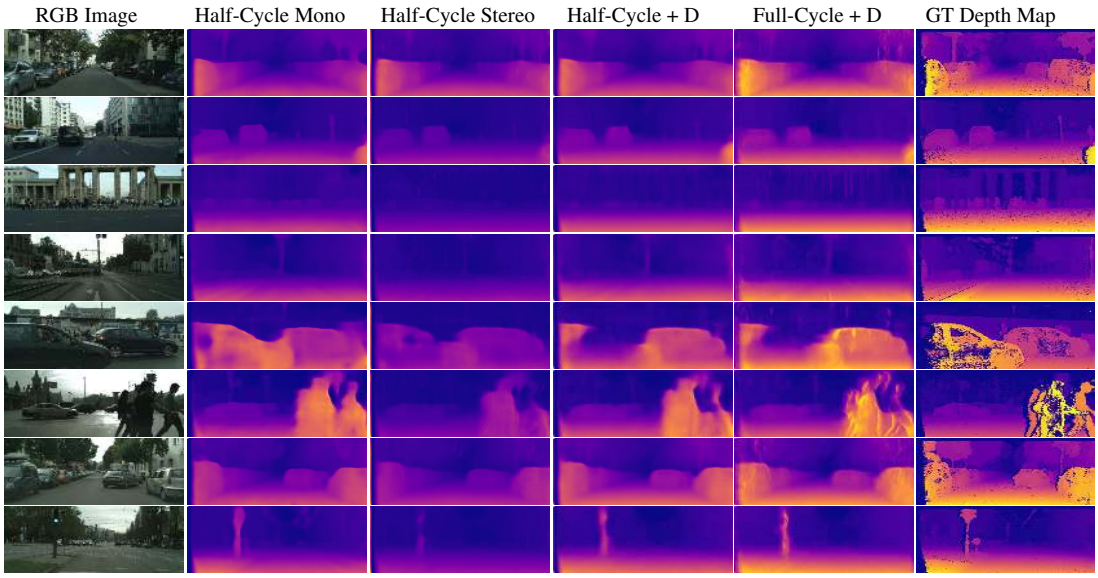


Figure 4.5: Qualitative comparison of different baseline models of the proposed approach on the Cityscapes testing dataset.

which uses a straight forward branch but with two disparity maps produced and combined; (iii) half-cycle with a discriminator (half-cycle + D), which use a single branch as in (ii) while adds a discriminator for the image synthesis; (iv) full-cycle with two discriminators (full-cycle + D), which is our whole model using a full cycle with two discriminators added; (v) full-cycle with two discriminators and sharing encoders (full-cycle + D + SE), which has the same structure as (iv) while the parameters of the encoders of the generators are shared.

Evaluation on KITTI. As we can see from Table 4.1, the baseline model Half-Cycle Stereo shows significantly better performance on seven out of eight evaluation metrics than the baseline model Half-Cycle Mono, demonstrating that the utilization of the stereo images and the combination of the two estimated complementary disparity maps clearly boosts the performance.

By using the adversarial learning strategy for the image synthesis, the baseline Half-Cycle + D outperforms the baseline Half-Cycle Stereo with around 1.7 points gain on the metric of Abs Rel, which verifies our initial intuition of using the adversarial learning to improve the quality of the image synthesis, and thus gain the improvement of the disparity prediction. In addition, we also observe in the training process, the adversarial learning helps to maintain a more stable convergence trend with small oscillations in terms of the training loss than the one without it (*i.e.* Half-Cycle Stereo), probably leading to a better optimized model.

It is also clear to observe that the proposed cycled generative network with adversarial learning (Full-Cycle + D) achieved much better results than the models with only half cycle (Half-Cycle + D) on all the metrics. Specifically, the Full-Cycle + D model improves the Abs Rel around 2 points, and also improves the accuracy a1 around 1.9 points over Half-Cycle + D. The significant improvement demonstrates the effectiveness of the proposed network design, confirming that the cycled strategy brings stronger constraint and supervision to optimize the both generators. Finally, we also show that the propose cycled model using a sharing encoder for the generator (Full-Cycle + D + SE). By using the sharing structure, we obtain even better results than the non-sharing model (Full-Cycle + D), which is probably because the shared one has a more compact network structure and thus is relatively easier to optimize with a limited number of training samples.

Evaluation on Cityscapes. We also conduct another ablation study on the Cityscapes dataset and the results are shown in Table 4.3. We can mostly observe similar trend of the performance gain of the different baseline models as we already analyzed on the KITTI dataset. The performance comparison of the baselines on this challenging dataset further confirms the advantage of the proposed approach. For the comparison of the model Half-Cycle + D and the model Full-Cycle + D, although the latter one achieves slightly worse results on the first two error metrics, it still produces clearly better performance on the remaining six evaluation metrics. Since there is no official evaluation protocol for depth estimation on this dataset, the results are evaluated with the protocol on the KITTI, and are directly evaluated on the disparity maps as they are directly proportional to each other. In Fig. 4.5, some qualitative comparison of the baseline models are presented.

State of the Art Comparison In Table 4.12, we compare the proposed full model with several state-of-the-art methods, including the ones with the supervised setting, *i.e.* Saxena *et al.* [59], Eigen *et al.* [3], Liu *et al.* [21], AdaDepth [40], Kuznietzov *et al.* [33] and Xu *et al.* [10], and the ones with the unsupervised setting, *i.e.* Zhou *et al.* [35], AdaDepth [40], Garg *et al.* [4] and Godard *et al.* [5]. Among all the supervised approaches, we have achieved very competitive performance to the best one of them (*i.e.* Xu *et al.* [10]), while ours is totally unsupervised without using any ground-truth depth data in training. For comparison with the unsupervised methods, we are also very close to the best competitor (*i.e.* Godard *et al.* [5]). AdaDepth [40] is the most technically related to our approach, which considers adversarial learning in a context of domain adaptation with extra synthetic training data. Ours significantly

outperforms their results with both the supervised and unsupervised setting, further demonstrating the effectiveness of the means we considered and proposed for unsupervised depth estimation with the adversarial learning strategy. As far as we know, there are not quantitative results presented in the existing works on the Cityscapes dataset.

Analysis on the Time Aspect. For the training of the whole network model, on a single Tesla K80 GPU, it takes around 45 hours on KITTI dataset with around 22k training images. For the running time, in our case with the resolution of 512×256, the inference of one image takes around 0.140 seconds, which is a near real-time processing speed.

4.1.4 Conclusions

We have presented a novel approach for unsupervised deep learning for the depth estimation task using the adversarial learning strategy in a proposed cycled generative network structure. The new approach provides a new insight to the community that shows depth estimation can be effectively tackled via an unsupervised adversarial learning of the stereo image synthesis. More specifically, a generative deep network model is proposed to learn to predict the disparity map between two image views under a calibrated stereo camera setting. Two symmetric generative sub-networks are respectively designed to generate images from different views, and they are further merged to form a closed cycle which is able to provide strong constraint and supervision to optimize better the dual generators of the two sub-networks. Extensive experiments are conducted on two publicly available datasets (*i.e.* KITTI and Cityscapes). The results demonstrate the effectiveness of the proposed model, and show very competitive performance compared to state-of-the-arts on the KITTI dataset.

4.2 Progressive Fusion for Unsupervised Binocular Depth Estimation using Cycled Networks [2]

Recent deep monocular depth estimation approaches based on supervised regression have achieved remarkable performance. However, they require costly ground truth annotations during training. To cope with this issue, in this Section we present a novel unsupervised deep learning approach for predicting depth maps. We introduce a new network architecture, named Progressive Fusion Network (PFN), that is specifically designed for binocular stereo depth estimation. This network is based on a multi-scale refinement strategy that combines the information provided by both stereo views. In addition, we propose to stack twice this network in order to form a cycle. This cycle approach can be interpreted as a form of data-augmentation since, at training time, the network learns both from the training set images (in the forward half-cycle) but also from the synthesized images (in the backward half-cycle). The architecture is jointly trained with adversarial learning. Extensive experiments on the publicly available datasets KITTI, Cityscapes and ApolloScape demonstrate the effectiveness of the proposed model which is competitive with other unsupervised deep learning methods for depth prediction.

4.2.1 Introduction

Most previous works considering deep architectures for predicting depth maps operate in a supervised learning setting [3, 10, 21, 58] and employ powerful deep regression models based on Convolutional Neural Networks (ConvNet). These models are usually designed for monocular depth estimation, *i.e.* they are trained to learn the transformation from a single RGB image to a depth map in a pixel-to-pixel fashion. However, supervised learning models require ground-truth depth data which are usually costly to acquire. This problem is especially relevant with deep learning architectures, as large amounts of data are typically required in order to produce satisfactory performance. Furthermore, monocular depth estimation is an inherently ill-posed problem due to the well-known scale ambiguity issue [59]. For instance, given an image patch of a blue sky, it is difficult to predict if this patch is infinitely far away (sky), or whether it is part of a blue object. Therefore, local information such as the texture must be combined

[2] "Progressive Fusion for Unsupervised Binocular Depth Estimation using Cycled Networks", A. Pilzer, S. Lathuilière, D. Xu, M. M. Puscas, E. Ricci and N. Sebe; IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2019.2942928

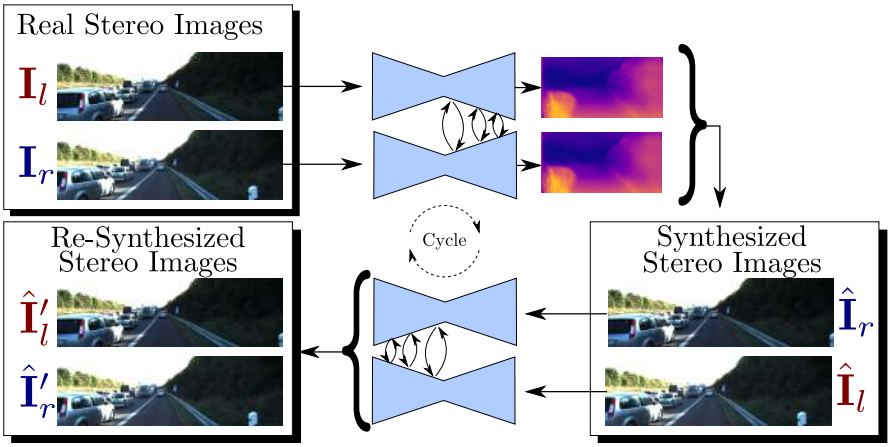


Figure 4.6: Motivation of the proposed unsupervised depth estimation approach using cycled generative networks.

with contextual information. Additionally, in complex environments as those encountered by autonomous driving cars, current depth estimation methods still have difficulties in predicting accurately depth maps from a single camera. These difficulties are encountered in particular when many objects are present in the scene due to the several occlusions.

To tackle these problems, *unsupervised* (also called self-supervised) learning-based approaches for depth estimation operating on a *stereo setting* have been introduced [5, 60, 61, 62]. These methods operate by learning the correspondence field (*i.e.* the disparity map) between two different image views of a calibrated stereo camera using only the left and right RGB images (no ground-truth depth map). The disparity refers to the difference in image location of an object seen by the left and right cameras. Importantly, the disparity value is inversely proportional to the object depth at the corresponding pixel location. Then, given the calibration parameters of the stereo cameras, the depth maps can be calculated using the predicted disparity maps. At test time, depending on the network architecture, depth is estimated either from a single stereo view [5] (referred to as *monocular depth estimation*) or stereo pairs[65] (referred to as *binocular* or *stereo depth estimation*). Thanks to this formulation, we avoid ground-truth data collection, using lidar for instance, that is much more complex (eg. multimodal-synchronization, hardware constraints) and expensive than adding a second camera. Another potential advantage of unsupervised depth estimation can be found in online adaptation as in [31], where the network is self-adapted in an online fashion at testing time when depth supervision is no more available. Most of previous works [5, 62] are based on a common strategy

introduced in [4]: given a pair of left and right images a neural network is trained for the task of predicting the right-to-left disparity map from the left image. The left image can then be re-synthesized by warping the right image accordingly to the predicted disparity. The network is trained by minimizing a left image reconstruction loss (see Sec 4.2.2.1 for technical details). This approach relies on the supervision from the image synthesis of an expected view, whose quality plays a direct influence on the performance of the estimated disparity map. Significant progress has been made recently along this research line [5, 35, 36].

We follow this research thread and propose a novel end-to-end trainable deep network model for adversarial learning-based depth estimation given stereo image pairs. Contrary to most recent works [4, 5, 62], we focus on the binocular scenario where stereo image pairs are available both at training and test time. The proposed approach consists of a generative sub-network which predicts the two disparity maps from the right to the left views and vice-versa. This sub-network is stacked twice in order to form a cycle as illustrated in Fig. 4.6. This novel network design provides strong constraints and supervision for each image view, facilitating the optimization of the network. It is important to mention that, despite the cycle shape and the use of adversarial loss, our cycle approach is not directly related to Cycle-GAN [7]. Cycle-GAN is designed for image-to-image translation when paired data are not available. In the case of binocular stereo depth estimation, paired data are available (*i.e.* corresponding left/right images). Our cycle approach can be interpreted as a form of data-augmentation since, at training time, the network learns to predict disparity maps from images of the training set (in the forward cycle pass), but also from synthesized images (in the backward cycle pass). In addition, it prevents the sub-network to predict blurred or deformed images in the forward cycle pass, since it would suffer the consequences in the backward cycle pass. The whole cycle is jointly learned and the final disparity map is produced by the first G network. The current Section extends 4.1 in several ways:

- First, we present a more detailed analysis of related works by including recently published works dealing with supervised and unsupervised depth estimation.

- Second, we propose a novel network architecture named Progressive Fusion Network (PFN), that is specifically designed for binocular stereo depth estimation. This network is based on a multi-scale refinement strategy that combines the information provided by the left and right images.

- Third, the cycle model proposed in 4.1 is adapted in order to benefit from the two disparity maps predicted by the proposed PFN.
- Finally, we significantly extend our quantitative evaluation by performing an in-detail ablation study and by comparing our binocular stereo model with the very recent works in this area. Our extensive experiments on three large publicly available datasets (*i.e.* KITTI [66], Cityscapes [18] and ApolloScape[19]) demonstrate the effectiveness of the proposed adversarial image synthesis, cycled generative network structure and Progressive Fusion Network. On the widely used KITTI dataset, our approach is competitive with state of the art methods on the static unsupervised setting.

The details of our method are presented in Section 4.2.2. Section 4.2.3 presents the experimental evaluation and conclusions are drawn in Section 4.2.4.

4.2.2 Proposed Approach

As mentioned in the introduction, our framework for unsupervised depth estimation has two main contributions. First, we propose to exploit cycle consistency in order to regularize better our model and therefore achieve better performance. The cycle consistency approach is combined with an adversarial learning strategy in order to further improve the predictions. The details of our model are given in Sections 4.2.2.2 and 4.2.2.3. Second, we propose a network architecture named Progressive Fusion Network, that uses a multi-scale approach to fuse the information provided by each image. The motivations and the details of this network architecture are given in Section 4.2.2.4. Before presenting the details of our approach, we briefly introduce in Sec. 4.2.2.1 the basics of unsupervised depth estimation and the notations used in the remaining of the paper.

4.2.2.1 Unsupervised Binocular Depth Estimation

In this work, we aim at estimating a depth map given a pair of images from calibrated stereo cameras. A supervised approach would consist in learning a stereo matching network that predicts depth [60]. In this scenario, the network is trained via minimization of a pixel-wise error measure between the predicted and the ground-truth disparities. Conversely, we follow an unsupervised approach: given a left image $\mathbf{I}_l$ and a right image $\mathbf{I}_r$, we are interested in predicting the disparity maps $\mathbf{d}_r$ and $\mathbf{d}_l$. The disparity map $\mathbf{d}_r$ is defined as the 2D map where each pixel value represents the offset of the corresponding pixel from the left and the

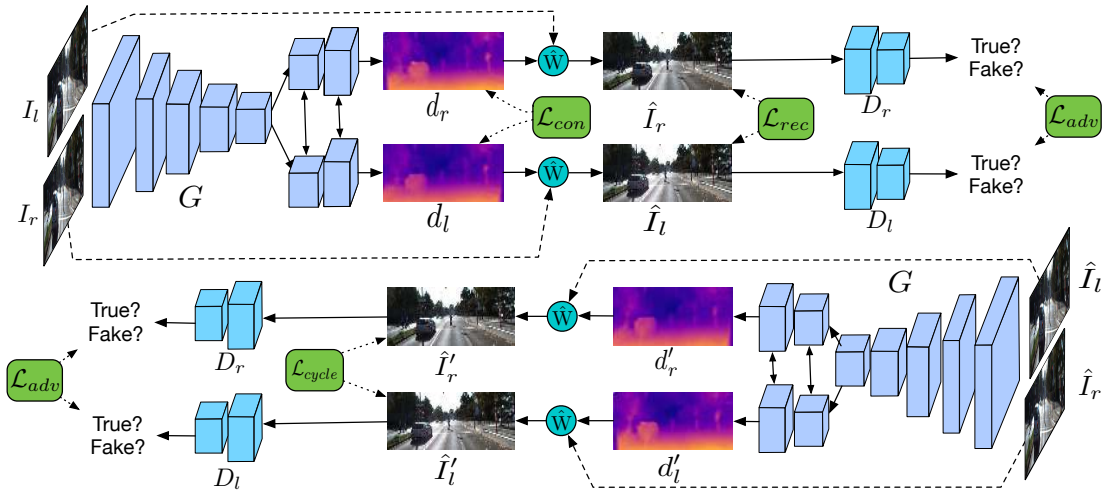


Figure 4.7: Illustration of the detailed framework of the proposed cycled generative networks with Progressive Fusion Network for unsupervised adversarial depth estimation. $\mathcal{L}_{rec}$ represents the reconstruction loss for different generators; $\mathcal{L}_{con}$ denotes a consistence loss between the disparity maps generated from the two generators.

right images. Symmetrically, $\mathbf{d}_l$ encodes the offsets from the right to the left images. We propose to estimate the disparity in an indirect way through image synthesis from different views. Specifically, the approach consists in training a network to predict disparity maps that can be used to generate the left images from the right images or vice-versa. Formally speaking, we assume that a right-to-left disparity map $\mathbf{d}_l$ is produced from a generator network G with both the left and right images, $\mathbf{I}_l$ and $\mathbf{I}_r$, as inputs. The warping function $f_w(\cdot)$ is used to perform the synthesis of the left image view by sampling from $\mathbf{I}_r$

$$\hat{\mathbf{I}}_l = f_w(\mathbf{d}_l, \mathbf{I}_r). \tag{4.8}$$

with

$$\mathbf{d}_l, \mathbf{d}_r = G(\mathbf{I}_l, \mathbf{I}_r). \tag{4.9}$$

Importantly, the image sampler used to implement the warping function $f_w(\cdot)$ needs to be differentiable in order to be able to train the whole model via gradient descent. Therefore, we use the image sampler from the spatial transformer network [67] that employs a bilinear sampler. A reconstruction loss between $\hat{\mathbf{I}}_l$ and $\mathbf{I}_l$ is thus utilized to provide supervision in optimizing the network G. Usually, the $\mathcal{L}_1$ loss is employed:

$$\mathcal{L}^l_{rec} = \|\mathbf{I}_l - \hat{\mathbf{I}}_l\|_1. \tag{4.10}$$

Symmetrically, we use the left-to-right disparity $\mathbf{d}_r$ to synthesize the left image:

$$\hat{\mathbf{I}}_r = f_w(\mathbf{d}_r, \mathbf{I}_l). \tag{4.11}$$

and obtain the corresponding loss:

$$\mathcal{L}_{rec}^r = \|\mathbf{I}_r - \hat{\mathbf{I}}_r\|_1. \tag{4.12}$$

Finally, if we assume that the images are rectified, and that we know the baseline distance b between the two cameras and the focal length f, we can obtain the depth at a pixel location (x, y) of the left image from the predicted disparity with $d_l = b\frac{f}{d(x,y)}$. We now detail how this general unsupervised approach can be extended to a cycle binocular model.

4.2.2.2 Network Training for Binocular Depth Estimation

In this section, we detail the training loss employed in our binocular depth estimation model. The reconstruction loss is defined as the sum of the reconstruction losses of the two images.

$$\mathcal{L}_{rec} = \mathcal{L}_{rec}^r + \mathcal{L}_{rec}^l \tag{4.13}$$

where $\mathcal{L}_{rec}$ is defined as in Eqn. (4.10). In order to constrain the predicted disparities on each other, we also add an $L1$-norm consistency loss as follows:

$$\mathcal{L}_{con} = ||\mathbf{d}_l - f_w(\mathbf{d}_l, \mathbf{d}_r)||_1 + ||\mathbf{d}_r - f_w(\mathbf{d}_r, \mathbf{d}_l)||_1 \tag{4.14}$$

Since the two disparity maps correspond to different views, they are not aligned and their consistency cannot be measured directly with an $\mathcal{L}_1$ loss . Inspired by [5], we use the warping operation to make them pixel-to-pixel aligned. More precisely, in the first term of Eqn. (4.14), since the left-to-right disparity $\mathbf{d}_r$ is aligned with the right image, we use the right-to-left warping $f_w(\mathbf{d}_l, .)$ introduced in Eqn. (4.8) to obtain a disparity aligned with $\mathbf{d}_l$.

In order to further improve the quality of the synthesized images, we also propose to use adversarial learning [6]. The key idea of adversarial learning is to train two networks simultaneously, a discriminator and a generator. The objective of the generator is to generate realistic images (in our case the right image from the left image and vice-versa). The goal of the discriminator is to distinguish real images of the training set from generated images. In our particular case, we add two discriminators D_r and D_l. The discriminator network D_r is trained to distinguish real right images $\mathbf{I}_r$ from right images that were synthesized from left images $\hat{\mathbf{I}}_r$. Similarly, D_l is trained to distinguish real left images $\mathbf{I}_l$ from left images $\hat{\mathbf{I}}_l$ that were synthesized

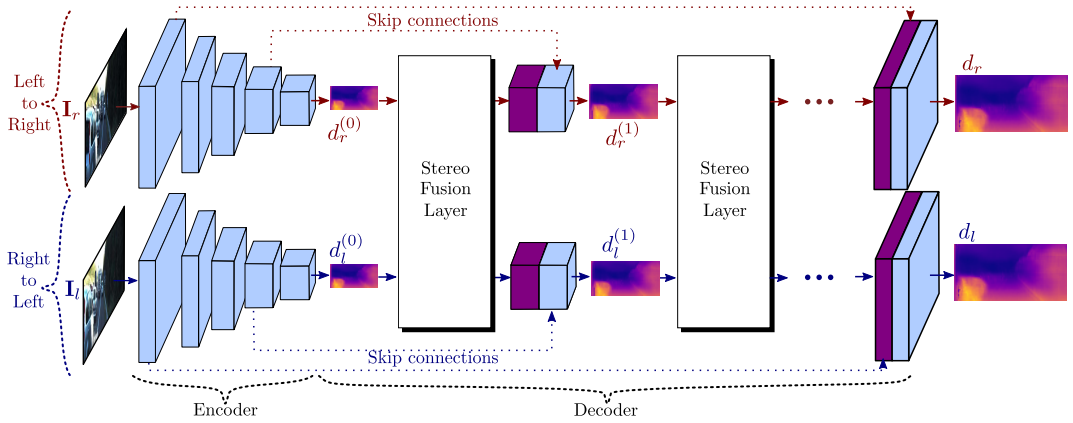


Figure 4.8: Illustration of the proposed Progressive Fusion Network (PFN). The network is composed of two streams that take as input the left and the right images respectively.

from right images. In [6] the proposed adversarial loss is formulated as a min-max objective function that involves a cross entropy loss. However, in this standard GAN formulation, the optimization generally suffers from vanishing gradients due to the sigmoid cross-entropy loss. There have been recent improvements in the GAN methodology to stabilize training and, to this aim, we use a least-square GAN loss [48] by substituting the cross-entropy loss by the least-squares function with binary coding (1 for real, 0 for synthesized). Consequently, the formulation in Eqn. (4.16) is split in two losses used to trained the discriminator and the generator respectively:

$$\begin{aligned}\mathcal{L}_{gan}^{D,r}(D_r) =& \mathbb{E}_{\mathbf{I}_r \sim p(\mathbf{I}_r)}[(D_r(\mathbf{I}_r) - 1)^2] \\ &+ \mathbb{E}_{\mathbf{I}_l \sim p(\mathbf{I}_l)}[D_r(f_w(\mathbf{d}_r, \mathbf{I}_l))^2]\end{aligned} \tag{4.15}$$

$$\mathcal{L}_{gan}^{G,r}(G) = \mathbb{E}_{\mathbf{I}_l \sim p(\mathbf{I}_l)}[(D_r(f_w(\mathbf{d}_r, \mathbf{I}_l)) - 1)^2] \tag{4.16}$$

The intuition behind Eqn. (4.15) is that the discriminator is trained to output 1 when the input image is a real right images and 0 when the input is a synthesized image. In Eqn. (4.15), the G network is trained in order to predict a d_r disparity map such that the discriminator confuses the synthesized image with real right images and, thus, outputs 1. The total adversarial loss for the left-to-right stream is given by:

$$\mathcal{L}_{gan}^{r} = \mathcal{L}_{gan}^{D,r} + \mathcal{L}_{gan}^{G,r} \tag{4.17}$$

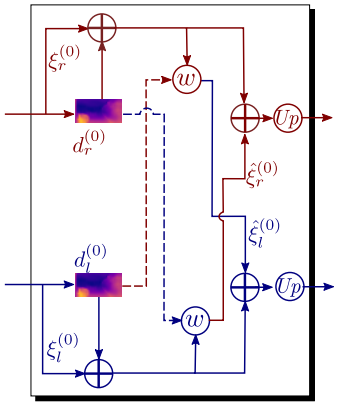


Figure 4.9: Detail of the PFN Stereo Fusion Layer in Fig. 4.8. In the left-to-right stream (in dark red) all tensors are aligned with right image. Conversely, in the right-to-left stream (in dark blue) all the tensors are aligned with the left image. The estimated left-to-right disparity $\mathbf{d}_r^{(0)}$ is used to align the left image feature map $\xi_l^{(0)}$ and the right-to-left disparity $\mathbf{d}_l^{(0)}$ with the left-to-right stream. The aligned tensor $\hat{\xi}_r^{(0)}$ is then concatenated with the right-to-left stream. Skip connections (dotted lines) are used to transfer local information from the encoder to the decoder. $\oplus$ denotes the concatenation operator, $\textcircled{w}$ denotes the warping operator introduced in (4.20), $\textcircled{Up}$ denotes the 2×2 Up-sampling operator.

We define similarly the adversarial loss for the right-to-left stream $\mathcal{L}_{gan}^{l}$ and obtain the total adversarial:

$$\mathcal{L}_{gan} = \mathcal{L}_{gan}^{r} + \mathcal{L}_{gan}^{l} \tag{4.18}$$

A major advantage of considering an adversarial loss is that it imposes a global consistency loss oppositely to the $\mathcal{L}_1$ loss used in Eqn.(4.13) that acts only locally. Note that, at test time, the inferred $\mathbf{d}_l$ and $\mathbf{d}_r$ are used as final outputs of the model and the discriminators are not used anymore.

4.2.2.3 Cycled Generative Networks for Binocular Depth Estimation

In order to further exploit the left and right images synthesized by our half-cycle network, we propose a cycled network structure. An overview of the proposed framework is shown in Fig. 4.7. A first generator network, forward half-cycle, produces two distinct disparity maps (d_r, d_l) from different view directions, and synthesizes different-view images as described in Section 4.2.2.2, namely $\hat{I}_r$,$\hat{I}_l$. In the second half-cycle, the generator network G takes as inputs these two synthesized images $\hat{I}_r$,$\hat{I}_l$ and predicts new disparity maps d'_r, d'_l that are again used to synthesize the opposite views $\hat{I}'_r$,$\hat{I}'_l$ from the synthesized images. The overall model we

obtain in this way forms a cycle. This cycle formulation can be seen as a data augmentation approach since, at training time, the network learns to predict disparity maps from the images of the training (in the forward half-cycle), but also from synthesized images (in the backward half-cycle). In the literature, standard methods using GANs for data augmentation [68], use generally two separated networks: a data generator network and the network finally used for prediction. In our case, we employ only a single network exploiting the left-right consistency of the data since the G network is used both to generate training data and to estimate depth. The second half-cycle prevents the first half-cycle network from predicting inconsistent disparity pairs. Indeed inconsistencies in the disparities predicted in the forward half-cycle would harm the estimations of the second half-cycle network. Consequently, imposing cycle consistency favors consistent predictions in the first half-cycle. At inference time, the second half-cycle is not used anymore. More formally, the cycled generative network is based on the half-cycle structure previously described. Assuming that we obtained the synthesized image $\hat{\mathbf{I}}_r$ and $\hat{\mathbf{I}}_l$ from the half-cycle network, we now aim at predicting the original left image from $\hat{\mathbf{I}}_r$ and the original right image from $\hat{\mathbf{I}}_l$. To this aim, $\hat{\mathbf{I}}_r$ and $\hat{\mathbf{I}}_l$ are used as input of the next cycle generative network G. G produces again two disparity maps $\mathbf{d}'_l$ and $\mathbf{d}'_r$. Again, we synthesize the left-view image $\hat{\mathbf{I}}'_l$ from $\hat{\mathbf{I}}_r$ and $\hat{\mathbf{I}}'_r$ from $\hat{\mathbf{I}}_l$ via the warping operation f_w. Similarly to the forward half-cycle, a reconstruction loss $\mathcal{L}'_{rec}$ is used for the backward half-cycle as in Eqn. (4.13). We also add a consistency loss $\mathcal{L}'_{rec}$ and an adversarial $\mathcal{L}'_{gan}$ as in Eqn.(4.14) and Eqn.(4.18), respectively. During adversarial learning, synthesized and real images are independently passed to the discriminator networks.

The full optimization objective consists on the combination of the reconstruction, adversarial and consistency losses for both half-cycles and can be written as follows:

$$\begin{aligned}\mathcal{L} = \gamma_1(\mathcal{L}_{rec} + \mathcal{L}'_{rec}) + \gamma_2(\mathcal{L}_{gan} + \mathcal{L}'_{gan}) \\ + \gamma_3(\mathcal{L}_{con} + \mathcal{L}'_{con}).\end{aligned} \tag{4.19}$$

where $\{\gamma_i\}_{i=1}^3$ represents a set of weights for controlling the importance of different terms.

Importantly, when the optimization is finished, given a testing pair $\{\mathbf{I}_l, \mathbf{I}_r\}$, the testing is performed using only the first half-cycle. Therefore, the proposed cycle approach does not increase the testing computation time but only the training complexity. Nevertheless, note that this cycle framework increases the training complexity by increasing the number of computation operations but it does not increase the number of parameters. Note that the discriminators of the two half-cycles share their parameters in order to avoid an increase in the number of

Layer	K	S	Channels	left to right branch		right to left branch	
Encoder (layers share weights among branches)				Input	Output	Input	Output
conv	7	2	64	I_L	$conv1_{L2R}$	I_R	$conv1_{R2L}$
pool	3	1	64	$conv1_{L2R}$	$pool1_{L2R}$	$conv1_{R2L}$	$pool1_{R2L}$
ResNetBlock	3		256	$pool1_{L2R}$	$resblock1_{L2R}$	$pool1_{R2L}$	$resblock1_{R2L}$
ResNetBlock	4		512	$resblock1_{L2R}$	$resblock2_{L2R}$	$resblock1_{R2L}$	$resblock2_{R2L}$
ResNetBlock	6		1024	$resblock2_{L2R}$	$resblock3_{L2R}$	$resblock2_{R2L}$	$resblock3_{R2L}$
ResNetBlock	3		2048	$resblock3_{L2R}$	$resblock4_{L2R}$	$resblock3_{R2L}$	$resblock4_{R2L}$
Decoder (layers do not share weights among branches)				Input	Output	Input	Output
UpConv	3	2	512	$resblock4_{L2R}$	$upconv6_{L2R}$	$resblock4_{R2L}$	$upconv6_{R2L}$
conv	3	1	512	$upconv6_{L2R} + resblock3_{L2R}$	$iconv6_{L2R}$	$upconv6_{R2L} + resblock3_{R2L}$	$iconv6_{R2L}$
UpConv	3	2	256	$iconv6_{L2R}$	$upconv5_{L2R}$	$iconv6_{R2L}$	$upconv5_{R2L}$
conv	3	1	256	$upconv5_{L2R} + resblock2_{L2R}$	$iconv5_{L2R}$	$upconv5_{R2L} + resblock2_{R2L}$	$iconv5_{R2L}$
UpConv	3	2	128	$iconv5_{L2R}$	$upconv4_{L2R}$	$iconv5_{R2L}$	$upconv4_{R2L}$
conv	3	1	128	$upconv4_{L2R} + resblock1_{L2R}$	ξ_r^0	$upconv4_{R2L} + resblock1_{R2L}$	ξ_l^0
conv	3	1	1	ξ_r^0	d_r^0	ξ_l^0	d_l^0
UpConv	3	2	64	ξ_r^0	$upconv3_{L2R}$	ξ_l^0	$upconv3_{R2L}$
bilinear upsampling	-	-	1	d_r^0	$up\text{-}d_r^0$	d_l^0	$up\text{-}d_l^0$
iconv3	3	1	64	$upconv3_{L2R} + pool1_{L2R} + up\text{-}d_r^0 + \hat{\xi}_l^0$	ξ_r^1	$upconv3_{R2L} + pool1_{R2L} + up\text{-}d_l^0 + \hat{\xi}_r^0$	ξ_l^1
disp3	3	1	1	ξ_r^1	d_r^1	ξ_l^1	d_l^1
upconv2	3	2	32	ξ_r^1	$upconv2_{L2R}$	ξ_l^1	$upconv2_{R2L}$
updisp3	-	-	1	d_r^1	$up\text{-}d_r^1$	d_l^1	$up\text{-}d_l^1$
iconv2	3	1	32	$upconv2_{L2R} + conv1_{L2R} + up\text{-}d_r^1 + \hat{\xi}_l^1$	ξ_r^2	$upconv2_{R2L} + conv1_{R2L} + up\text{-}d_l^1 + \hat{\xi}_r^1$	ξ_l^2
disp2	3	1	1	ξ_r^2	d_r^2	ξ_l^2	d_l^2
upconv1	3	2	16	ξ_r^2	$upconv1_{L2R}$	ξ_l^2	$upconv1_{R2L}$
updisp2	-	-	1	d_r^2	$up\text{-}d_r^2$	d_l^2	$up\text{-}d_l^2$
iconv1	3	1	16	$upconv1_{L2R} + up\text{-}d_r^2 + \hat{\xi}_l^2$	ξ_r^3	$upconv1_{R2L} + up\text{-}d_l^2 + \hat{\xi}_r^2$	ξ_l^3
disp1	3	1	1	ξ_r^3	d_r^3	ξ_l^3	d_l^3

Table 4.4: Detailed architecture of the proposed network, for readability reasons we show the half-cycle structure. K, S and $Channels$ denote convolutions kernel size, stride and output channels respectively. For ResNet blocks K denotes the number of blocks. The $+$ indicates concatenation between feature maps.

parameters. In other words, we use a right view discriminator for the right images ($\mathbf{I}_r$, $\hat{\mathbf{I}}_r$, $\hat{\mathbf{I}}'_r$) and a left view discriminator for the left view images ($\mathbf{I}_l$, $\hat{\mathbf{I}}_l$, $\hat{\mathbf{I}}'_l$), they are denoted in Fig. 4.7 as D_r and D_l respectively.

4.2.2.4 Progressive Fusion Network

When it comes to binocular depth estimation, the question of how to fuse the information provided by each image needs to be addressed. A standard approach, as used in [5], consists in simply concatenating the two images over the color axis. We denote this approach as *early fusion*. On the contrary, in the previous Section 4.1, we used a *late fusion* approach that consists in estimating the two disparities separately employing two separated networks, before fusing them. In this section, we first explain why these two approaches suffer from the misalignment between the input images and the output disparity map. Secondly, we propose a neural network architecture to face this issue. Let us consider again the case in which we aim at estimating the

right-to-left disparity $\mathbf{d}_l$ from the images $\mathbf{I}_l$ and $\mathbf{I}_r$. When looking at the images, we can notice that the edges in $\mathbf{I}_l$ are perfectly aligned with the edges of $\mathbf{d}_l$. This observation results directly from the disparity definition. Therefore, in order to estimate $\mathbf{d}_l$ at a pixel location (u, v), the model needs to look at the pixel values of $\mathbf{I}_l$ in the neighbour of the pixel $\mathbf{I}_l(u, v)$. Conversely, the edges in $\mathbf{d}_l$ and $\mathbf{I}_r$ are not aligned. More precisely, the model would need to look at the pixel around $\mathbf{I}_r(u + \mathbf{d}_l(u, v), v)$ in order to estimate $\mathbf{d}_l(u, v)$.

In the context of convolutional neural network, this observation leads to two conclusions. First, when using an early fusion approach, the local information can be fused by the network, in practice, only after several layers, when the receptive field of the activations are larger than the disparity value we want to estimate. Second, in the case of a late fusion network, the d_l disparity estimated from $\mathbf{I}_l$ will have better edges since the network can benefit from the alignment. Conversely, $\mathbf{I}_r$ will have a lower quality since the corresponding network has to handle the input-output misalignement. Therefore, the benefit brought by the use of $\mathbf{I}_r$ will not be substantial.

To tackle this issue, we propose a Progressive Fusion Network (PFN). The key idea behind PFN is to first estimate low resolution disparity maps that are then used to align the image features. These aligned feature maps are employed to refine the disparity maps at the higher resolutions. This method is applied iteratively until we obtain the desired high resolution disparity maps. This iterative procedure is well in line with the multi-scale approaches that have shown good performances in the monocular supervised setting (see Sec 2). The details of the architecture are given in Fig. 4.8 and Fig. 4.9.

We first apply an encoder network on each input images obtaining two feature maps $\xi_r^{(0)}$ and $\xi_l^{(0)}$. In our particular case, we use a ResNet-50 architecture since it has already shown good performances on the depth estimation problem [5, 23]. We then estimate the left and right low resolution disparities ($\mathbf{d}_l^{(0)}$ and $\mathbf{d}_r^{(0)}$ respectively) from $\xi_l^{(0)}$ and $\xi_r^{(0)}$ respectively. To do so, we employ a single 3×3 convolutional layer with sigmoid activations. Now that we have a first estimation of the disparity from the left to the right image, we can employ this disparity $\mathbf{d}_r^{(0)}$ to warp the feature maps $\xi_l^{(0)}$ and the disparity $\mathbf{d}_l^{(0)}$ from the opposite stream:

$$\hat{\xi}_r^{(0)} = f_w(\mathbf{d}_r^{(0)}, \xi_l^{(0)} \oplus \mathbf{d}_l^{(0)}) \tag{4.20}$$

where $\oplus$ denotes the concatenation operator. By concatenating the features and the disparity, we provide to the left-to-right stream all the information currently available in the right-to-left stream. We obtain a complete left image representation that is aligned with the right image.

Symmetrically, $\mathbf{d}_l^{(0)}$ is used to warp the feature maps $\xi_r^{(0)}$ and $\mathbf{d}_l^{(0)}$ computed in the opposite stream according to $\hat{\xi}_l^{(0)} = f_w(\mathbf{d}_l^{(0)}, \xi_r^{(0)} \oplus \mathbf{d}_r^{(0)})$. Then, we concatenate $\xi_r^{(0)}$, $\hat{\xi}_r^{(0)}$ and $\mathbf{d}_r^{(0)}$ and perform 2×2 up-sampling. Finally, the next resolution feature map $\xi_r^{(1)}$ is obtained by concatenation with the feature maps of the encoder with the same dimension as in a standard U-Net [69]. The skip connections are employed to transfer directly the local information from the encoder to the decoder. A similar operation is applied on the left network leading to $\xi_l^{(1)}$. All these operations are performed four times in order to obtain the full resolution disparities.

In order to further benefit from the multi-scale approach, we employ the $L1$-norm reconstruction loss $\mathcal{L}_{rec}$ at every resolution $i \in \{0..3\}$ for both the right and left images:

$$\mathcal{L}_{rec}^{(i)} = \|\mathbf{I}_l^{(i)} - f_w(\mathbf{d}_l^{(i)}, \mathbf{I}_r^{(i)})\|_1 + \|\mathbf{I}_r^{(i)} - f_w(\mathbf{d}_r^{(i)}, \mathbf{I}_l^{(i)}).\|_1,$$

Note that $\mathcal{L}_{rec}^{(3)}$ corresponds to the highest dimension, and in this way, to the loss given in Eqn. (4.13). Consequently, when training this multi-scale model, Eqn. (4.13) is replaced by the following multi-scale loss:

$$\mathcal{L}_{rec} = \sum_{i=0}^{3} \mathcal{L}_{rec}^{(i)}. \tag{4.21}$$

A multi-scale loss is also employed in [4, 5], however, in our model, the low resolution depth maps are not only used to deeply supervise the network as in [5]. Instead, the low resolutions depth maps are used to correct the misalignment between the images and, in this way, help the network to better predict the higher resolutions.

4.2.2.5 Network Implementation Details

We now describe the details of the network implementation. For the encoder of G, we use a ResNet-50 backbone network as in [23]. The left and right encoders share the weights. Conversely, the forward and the backward cycle paths share their parameters. For the discriminators D_l and D_r, we employ a network structure which has five consecutive convolutional operations with a kernel size of 3, a stride size of 2 and a padding size of 1, and batch normalization [63] is performed after each convolutional operation. The adversarial loss is applied to output patches and is implemented following[48]. The encoder network takes as input images of size 256×512. The ResNet-50 encoder outputs high level features of size $(4 \times 8 \times 2048)$. As mentioned in Sec 4.2.2.4, each up-sampling is followed by a convolution layer. We employ 3×3 convolution layers with number of channels of 512, 256, 128 and 64 respectively with *Elu* activations.

Method	Warping	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
		lower is better				higher is better		
Half-Cycle Mono+ L_{gan}4.1	*w/o*	0.165	1.756	6.164	0.257	0.773	0.914	0.962
Half-Cycle Stereo + L_{gan}4.1	*w/o*	0.163	1.620	6.129	0.254	0.770	0.913	0.962
Cycle Stereo + L_{gan}4.1	*w/o*	0.153	1.388	6.016	0.247	0.789	0.918	0.965
Half-Cycle Stereo	d	0.159	1.374	6.105	0.261	0.764	0.909	0.960
Half-Cycle Stereo	$d \oplus \xi$	0.153	1.260	5.960	0.254	0.777	0.915	0.963
Half-Cycle Stereo + L_{gan}	$d \oplus \xi$	0.148	1.209	5.827	0.246	0.789	0.921	0.966
Cycle Stereo	d	0.146	1.246	5.833	0.239	0.791	0.922	0.968
Cycle Stereo	$d \oplus \xi$	0.141	1.235	5.661	0.234	0.807	0.930	0.970
Cycle Stereo + L_{gan}	$d \oplus \xi$	0.137	1.199	5.721	0.234	0.806	0.928	0.970
Cycle Stereo + L_{gan} + SSIM	$d \oplus \xi$	**0.102**	**0.802**	**4.657**	**0.196**	**0.882**	**0.953**	**0.977**

Table 4.5: Quantitative evaluation results of different variants of the proposed approach on the KITTI dataset for the ablation study. The estimated depth range is from 0 to 80 meters. $\mathcal{L}_{gan}$ denotes the use of the adversarial loss

The detailed architecture of our network is described in Table 4.4 where we present the structure of our half-cycle network. We specify the inputs and outputs of each part of the network. In particular, following the notation of previous Section 4.2.2.4 we denoted with $\hat{\xi}$ the concatenation of features and disparity of one branch of the network after the warping that aligns them with the other branch.

4.2.3 Experimental Results

4.2.3.1 Experimental Setup

Datasets and Evaluation We carry out experiments on three large stereo images datasets, *i.e.* KITTI [17], Cityscapes [18] and ApolloScape [19]. For the details about dataset preparation, preprocessing and evaluation metrics please refer to Chapter 3.

Training Procedure and Parameter Setup. We train the models denoted with *Half-Cycle Stereo* with a standard training procedure, *i.e.* initializing the network with random weights and making the network train for 10 epochs. This corresponds to $\approx 28K$ steps for both the KITTI and Cityscapes datasets and $11.5K$ steps for ApolloScape. The models denoted with *Cycle Stereo* are optimized starting from the corresponding pre-trained half-cycle model. We train the full cycle model for 10 additional epochs with the same optimization hyper-parameters. We use the Adam optimizer for the optimization. The momentum parameter and the weight

Method	Warping	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
		lower is better				higher is better		
Half-Cycle Mono 4.1	*w/o*	0.468	7.399	5.741	0.493	0.735	0.890	0.945
Half-Cycle Stereo 4.1	*w/o*	0.462	6.098	5.740	0.377	0.708	0.873	0.937
Half-Cycle + $\mathcal{L}_{gan}$ 4.1	*w/o*	0.439	5.714	5.745	0.400	0.711	0.877	0.940
Cycle + $\mathcal{L}_{gan}$ 4.1	*w/o*	0.440	6.037	5.443	0.398	0.730	0.887	0.944
Half-Cycle Stereo	d	0.465	6.783	5.503	0.429	0.732	0.887	0.945
Half-Cycle Stereo	$d \oplus \xi$	0.436	6.357	4.877	0.364	0.778	0.915	0.958
Half-Cycle Stereo +$\mathcal{L}_{gan}$	$d \oplus \xi$	0.429	6.304	5.051	0.343	0.778	0.913	0.957
Cycle Stereo	d	0.445	6.008	5.372	0.488	0.743	0.893	0.947
Cycle Stereo	$d \oplus \xi$	0.420	5.767	4.749	0.379	0.790	0.919	0.959
Cycle Stereo + $\mathcal{L}_{gan}$	$d \oplus \xi$	0.418	5.799	4.698	0.343	0.787	0.917	0.959
Cycle Stereo + $\mathcal{L}_{gan}$+ SSIM	$d \oplus \xi$	**0.404**	**5.677**	**4.534**	**0.324**	**0.792**	**0.922**	**0.962**

Table 4.6: Quantitative evaluation results of different variants of the proposed approach on the Cityscapes dataset for the ablation study. $\mathcal{L}_{gan}$ denotes the use of the adversarial loss.

decay are set to 0.9 and 0.0002, respectively. The final optimization objective has weighed loss parameters $\gamma_1 = 1$, $\gamma_2 = 0.1$ and $\gamma_3 = 0.1$. The batch size for training is set to 8 stereo image pairs and the learning rate is 10^{-5} in all the experiments. Unlike in 4.1 where the learning rate is reduced, in this work it is constant and each experiment is performed for 10 epochs, significantly reducing the training time. In addition, in Section 4.1, the network is trained for 50 epochs, while the model proposed in this work requires only 20 epochs to converge. The simpler training procedure can be explained by the lower number of parameters of our proposed model. Indeed the four decoders used in 4.1 are replaced by two decoders that share parameters. Importantly, in our preliminary experiments, we observed that the SSIM loss is really sensitive to the training schedule (number of iterations and learning decay) on KITTI. In order to draw a fair comparison with [5], we employ the training schedule of [5], when using SSIM on KITTI.

With respect to the time aspect, the training of the *Half-Cycle Stereo* network models, on two Titan Xp GPUs and KITTI dataset for 10 epochs, takes around 4.5 hours for the simpler to 7 hours for the more complex model. The full model *Cycle Stereo* requires 10 additional epochs of training that take from 5 to 8 hours depending on the complexity of the model. Regarding testing, in our experiments the inference time for each stereo pair is 45 ms.

The proposed model is implemented using the deep learning library *TensorFlow* [64]. The input images are down-sampled to a resolution of 512×256 from 1226×370 in the case of the

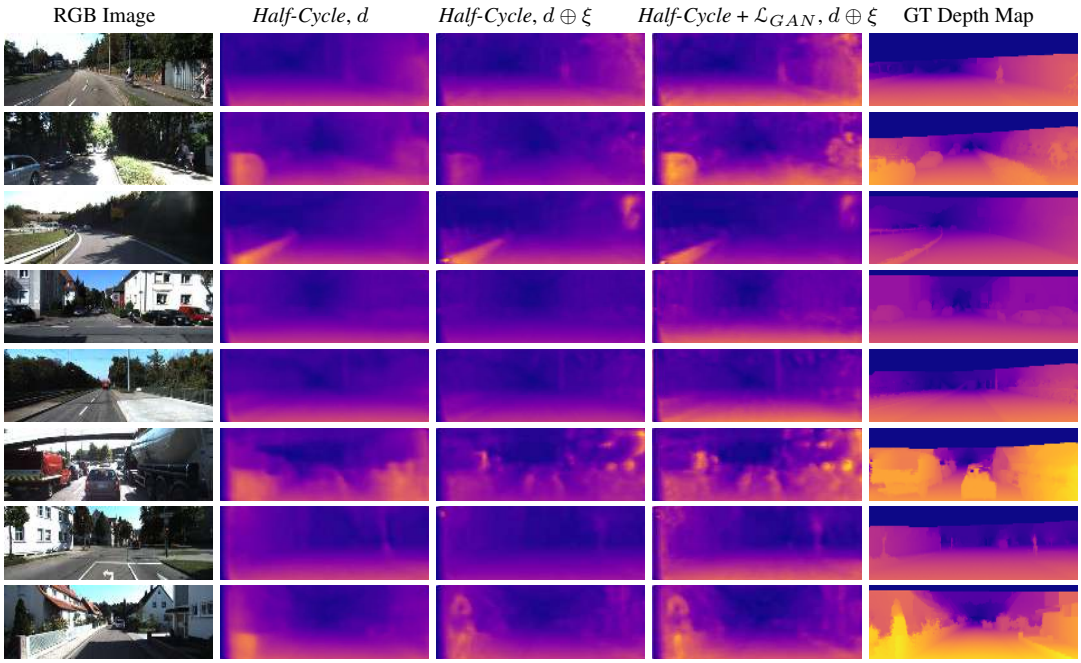


Figure 4.10: Qualitative comparison of different baseline models of the proposed *Half-Cycle* approach on KITTI Eigen test split. From left to right our stereo disparity progressive fusion, then stereo disparity and features progressive fusion and in the fourth column the full *Half-Cycle* model with adversarial learning. First column on the left is the RGB images and right the ground truth depth.

KITTI dataset, while for the Cityscapes dataset, at the bottom one fifth of the image is cropped following [5] and then is resized to 512×256. The resolution of the ApolloScape original images is 3384×2710 pixels. After rectification and cropping using the API of [19], we obtain 2048×1268 pixel images. We then adopt the standard preprocessing used for the Cityscapes dataset.

4.2.3.2 Ablation study: Baseline Models

We compare several baseline models for the ablation study:

1. Half-cycle with a monocular setting (*Half-Cycle Mono*), which uses the forward branch to synthesize from one image view to the other with a single disparity map output and the single RGB image is as input during testing;

2. Half-cycle with a stereo setting (*Half-Cycle Stereo*), which uses the forward branch but

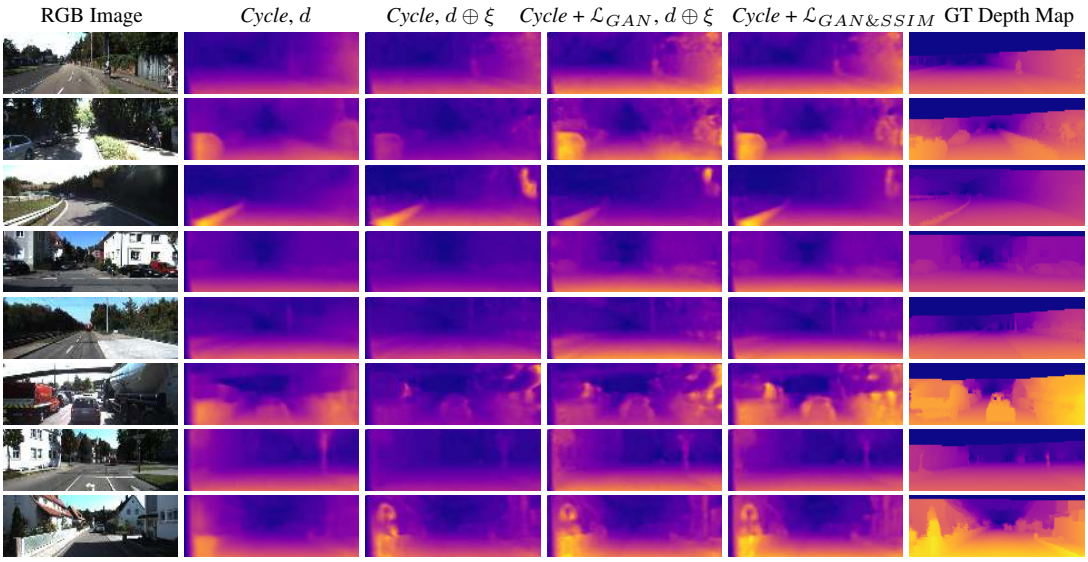


Figure 4.11: Qualitative comparison of different baseline models of the proposed *Cycle Stereo* approach on KITTI Eigen test split. From left to right RGB images, *Cycle Stereo* with continuous disparity fusion, *Cycle Stereo* with disparity and features fusion, in column four the full model trained with adversarial learning, in column five the futher refined full model with SSIM (self-similarity) loss and in the right column the ground truth depth maps.

the network takes as input the two images. It corresponds to the model as described in Sec 4.2.2.2 where G is a PFN as described in Sec 4.2.2.2;

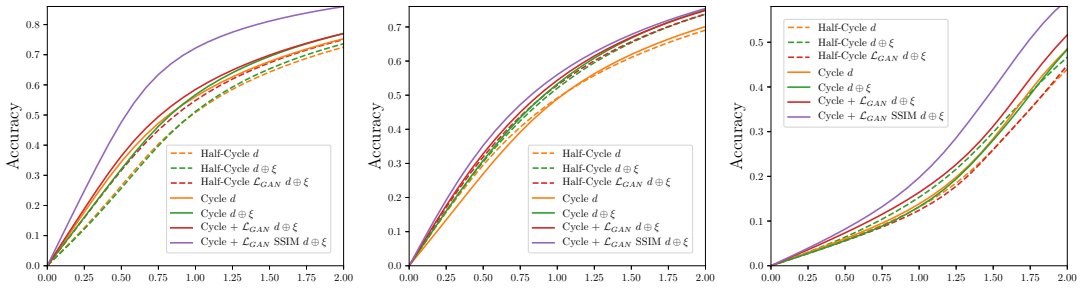
3. Cycle Stereo, which corresponds to the model as described in Sec 4.2.2.3 where G is also a PFN as described in Sec 4.2.2.2.

We propose to evaluate each model with and without the use of the adversarial loss. In addition, in order to understand the role of our PFN, we propose to compare three variants of the compared models:

1. The model without warping (referred to as *w/o* in Tables 4.5). In that case, we adopt a late fusion approach as in 4.1;

2. A model in which only the estimated disparities (referred to as d in Tables 4.5 and 4.6) are shuttled to the other stream. Formally speaking, Eqn. (4.20) is replaced by $\hat{\xi}_r^{(0)} = f_w(\mathbf{d}_r^{(0)}, \mathbf{d}_l^{(0)})$.

3. The full model in which both the disparities and the feature maps are shuttled (referred to as $d \oplus \xi$ in Tables 4.5 and 4.6).



(a) KITTI dataset **(b)** Cityscape dataset **(c)** ApolloScape dataset

Figure 4.12: Accuracy plot with a varying threshold parameter value τ for the KITTI, Cityscapes and ApolloScape datasets. The accuracy threshold α is set to $\alpha = 1.10$ for better visualization

Method	Warping	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
		lower is better				higher is better		
Half-Cycle	d	0.485	14.585	16.098	0.452	0.533	0.802	0.897
Half-Cycle	$d \oplus \xi$	0.469	13.443	15.779	0.448	0.568	0.797	0.886
Half-Cycle + L_{gan}	$d \oplus \xi$	0.446	12.283	14.600	0.432	0.571	0.822	0.903
Cycle	d	0.452	13.079	14.927	0.437	0.573	0.815	0.902
Cycle	$d \oplus \xi$	0.436	11.699	14.661	0.426	0.595	0.810	0.897
Cycle + L_{gan}	$d \oplus \xi$	0.423	11.582	14.415	0.422	0.624	0.824	0.905
Cycle + L_{gan} + SSIM	$d \oplus \xi$	**0.387**	**10.097**	**13.449**	**0.396**	**0.669**	**0.843**	**0.915**

Table 4.7: Quantitative evaluation results of different variants of the proposed approach on the ApolloScape dataset.

4.2.3.3 Ablation study: Results and discussion

To validate that the proposed cycled generative network helps and that the proposed PFN is effective for the task, we present an extensive ablation study on both the KITTI dataset (see Table 4.5), on the Cityscape dataset (see Table 4.6) and on the ApolloScape dataset (see Table 4.7).

First, we observe, that the cycle approach consistently outperforms the *Half-Cycle* approach. For instance on KITTI, if we compare the *Half-Cycle Stereo*+$\mathcal{L}_{GAN}$ model with *Cycle Stereo*+$\mathcal{L}_{GAN}$ in which we warp $d \oplus \xi$, we observe a 0.0114 gain according to the Abs Rel metric, that corresponds to a 7.76% improvement. Similar gain can be observed for all the metrics used in the comparison. Second, we consistently observe a gain when we employ our *PFN* network with respect to the fusion model proposed in 4.1 independently of the use of

Method	Cycle Inputs	Warping	Dataset	KITTI				ApolloScape			
				Abs Rel	Sq Rel	RMSE	RMSE log	Abs Rel	Sq Rel	RMSE	RMSE log
Half-Cycle + L_{gan}	-	$d \oplus \xi$	K	0.148	1.209	5.827	0.246	0.446	12.283	14.600	0.432
Cycle + L_{gan}	($\hat{I}_r$,I_l)	$d \oplus \xi$	K	0.146	1.466	5.918	0.244	0.441	12.292	14.877	0.438
Cycle + L_{gan}	($\hat{I}_r$,$\hat{I}_l$)	$d \oplus \xi$	K	**0.137**	**1.199**	**5.721**	**0.234**	**0.423**	**11.582**	**14.415**	**0.422**

Table 4.8: Ablation study: exploiting resynthesized images. We compare two different approaches for exploiting the resynthesized images in our cycle network. We present results for the KITTI dataset (left) and for the ApolloScape dataset (right).

the cycle approach. Again, this boost in performance brought by the *PFN* is observed on both datasets and according to all the metrics employed in this comparison. Interestingly, we notice that, independently of the use of cycle or adversarial loss, warping both the features and the disparities, performs better than warping only the disparities. We observe that, on both datasets, the adversarial loss helps predicting better depth maps. It confirms that adding a loss that acts globally can be beneficial for depth estimation. Finally, we report the results when we add the self-similarity loss proposed in [5] (referred to as +SSIM in Table 4.5), Intuitively, the SSIM loss measures how the object structure in the scene, is preserved in the synthesized image, independently of the average luminance and contrast. For more technical details, please refer to [70]. We observe that it further improves the results of our proposed model.

In order to further compare the different baselines, we propose to plot the accuracy metric described in Sec. **??** for different threshold values. More precisely, considering that $\hat{d}_i$, d_i are the estimated and ground truth depth values for pixel i, we measure the percentage of $\hat{d}_i$ such that $\delta = \max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) < a^{\tau}$ when τ varies. Note that contrary to the scores reported in Table 4.5 and 4.6, we chose $\alpha = 1.1$ for the sake of better visualization. The obtained plots are reported in Fig. 4.12.

On the KITTI dataset, we first notice that the results in the plot are in line with those presented in Table 4.5 since we clearly observe the benefit of the use of both our cycle setting and the proposed *PFN*. Interestingly, for both the *Half-Cycle* and the *Cycle* models, the use of the adversarial loss (red lines) reduces the amount of small errors ($\tau < 1.00$). The amount of large errors ($\tau > 1.00$) is similar to what is obtained without adversarial loss (green lines). We also observe that the performance gain of the *Cycle* approach (solid lines) is spread uniformly over the whole range of errors. Similarly, adding the SSIM loss reduces uniformly the errors and leads to the best performing model. Concerning the Cityscapes dataset, the accuracy plot confirms the benefit of using our *Cycle* approach. Similarly to the KITTI dataset, for both

the *Half-Cycle* and the *Cycle* models, the use of the adversarial loss (red lines) reduces the amount of small errors ($\tau < 1.00$) but the amount of large errors ($\tau > 1.00$) is similar to what is obtained without adversarial loss (green lines). Nevertheless, the boost of the *Cycle* approach is smaller than that on the KITTI dataset. When warping only the disparities, the *Half-Cycle* and *Cycle* models perform similarly but the use of a cycle improves the predictions when warping the feature maps (green and red lines). As observed on the KITTI dataset, the SSIM loss further reduces the prediction errors.

In addition to KITTI and Cityscapes, we report an ablation study on the ApolloScape dataset in Table 4.7. This dataset provides dense depth annotation and therefore allows more accurate evaluation. We observe that, for both the *Half-Cycle* and the *Cycle*, our proposed feature alignment and sharing among stereo branches improves the perfomance. Moreover, it is clear that adversarial learning contributes to improving the results. In both *Half-Cycle* and *Cycle* settings, every evaluation metric shows an improvement. These observations are well in-line with the numbers reported on the KITTI and Cityscapes and further demonstrate the effectiveness of our approach.

We also perform a qualitative comparison of the different baseline models with the proposed model. This qualitative evaluation is performed on the KITTI dataset and the results are shown in two figures in which we compare the *Half-Cycle* models (Fig. 4.10) and the *Cycle* models (Fig. 4.11) respectively. First, we observe that the *Cycle* setting generates smoother disparity maps than the *Half-Cycle* setting. In addition, when only the disparities are exchanged between the two streams of the *PFN*, we obtain very smooth predictions but with a low level of detail. When both the disparity and the feature maps are warped ($d \oplus \xi$ models), the predicted depth maps are more detailed and have sharper edges. In addition, by looking at the rows 6 and 8 of Figs. 4.10) and 4.11), we notice that the models without feature warping have difficulty in estimating the depth of nearby objects. Predicting accurate depth maps for these examples is challenging since the network needs to handle larger misalignment between the two input images. These examples illustrate the benefit of our proposed model which is better handle these difficult cases. Finally, by looking at the rows 1, 2, 4, 5 and 7, we can see that the models without adversarial loss underestimate the depth of the roads in foreground. Estimating the depth of the road is challenging since the image is almost uniform in these regions and the network cannot exploit the edges to estimate the disparity values. The fact that the GAN loss seems to help predicting the depth better for the uniform image regions may explain the reduction of small errors observed with the adversarial loss in Fig.4.12. Concerning the Cityscapes dataset,

Method	Discriminator	Warping	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
			lower is better				higher is better		
Half-Cycle + L_{gan}	D shared	$d \oplus \xi$	0.155	1.398	5.951	**0.245**	0.785	0.919	**0.966**
Half-Cycle + L_{gan}	D_r and D_l non-shared	$d \oplus \xi$	**0.148**	**1.209**	**5.827**	0.246	**0.789**	**0.921**	**0.966**

Table 4.9: Ablation study: discriminator usage. We evaluate on KITTI the impact of sharing weights among discriminators.

Method	Discriminator	Warping	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
			lower is better				higher is better		
Cycle	w/o	$d \oplus \xi$	0.149	1.338	5.837	0.247	0.792	0.921	0.966
Cycle + L_{gan}	1^{st} half-cycle	$d \oplus \xi$	0.144	1.399	5.794	0.237	0.801	0.927	0.969
Cycle + L_{gan}	2^{nd} half-cycle	$d \oplus \xi$	0.141	1.229	**5.685**	0.235	0.804	0.927	0.969
Cycle + L_{gan}	Both half-cycles	$d \oplus \xi$	**0.137**	**1.199**	5.721	**0.234**	**0.806**	**0.928**	**0.970**

Table 4.10: Ablation study: discriminator usage. We evaluate on KITTI the impact of discriminators on different part of the architecture.

we observe a similar trend to the KITTI dataset by looking at the qualitative results reported in Figs. 4.13 and 4.14. The proposed *PFN* produces very smooth disparities but without much details when exchanging only the disparity maps between the two streams. For example the road sign in row 6 (Fig. 4.13 and 4.14) is barely distinguishable and the parked cars in row 5 appear in the disparity maps as a single continuous object. Models trained with feature and disparity warping ($d \oplus \xi$) improve the estimations allowing to capture better the details of the objects and the background for images in row 1, 2 and 5 (Fig. 4.13 and 4.14). This is further improved by the cycle setting and the adversarial training that, as shown in Fig. 4.14, produces more detailed disparities especially in challenging image areas such as those corresponding to the background.

Ablation study: exploiting resynthesized images. Our proposed model is designed in two *Half-Cycle* blocks. The first reconstructs images that are used as input in the second *Half-Cycle*. In Table 4.8, we compare two different approaches for exploiting the resynthetised images on both KITTI and Apolloscape. We perform an experiment where we input in the second *Half-Cycle* the right resynthetised image and the real left image. This approach is compared to our approach where we input both synthesized images. We observed that the performances decrease when we use $(\hat{I}_r, I_l)$ as input compared to our proposed *Cycle* model that takes in input $(\hat{I}_r, \hat{I}_l)$. The performances are similar to our half-cycle baseline according to several metrics.

Method	Alignment	Warping	KITTI				ApolloScape			
			Abs Rel	Sq Rel	RMSE	RMSE log	Abs Rel	Sq Rel	RMSE	RMSE log
Half-Cycle	w/o	d	0.169	1.503	6.204	0.265	0.501	14.352	16.134	0.462
Half-Cycle	w/o	$d \oplus \xi$	0.159	1.468	6.087	0.251	0.491	14.383	16.366	0.458
Half-Cycle + L_{gan}	w/o	$d \oplus \xi$	0.151	1.501	5.905	0.249	0.468	13.107	15.023	0.445
Half-Cycle	With	d	0.160	1.374	6.105	0.261	0.485	14.585	16.098	0.452
Half-Cycle	With	$d \oplus \xi$	0.154	1.260	5.960	0.254	0.469	13.443	15.779	0.448
Half-Cycle + L_{gan}	With	$d \oplus \xi$	**0.148**	**1.209**	**5.827**	**0.246**	**0.446**	**12.283**	**14.600**	**0.432**

Table 4.11: Ablation study: impact of feature alignment: we compare the results without alignment of disparities and features between the two stereo branches (upper half) and with alignment (bottom half). We conducted experiments on KITTI and Apolloscape.

It validates experimentally our design choice for the cycle inputs.

Ablation study: discriminator usage. In our model as described in Sec.4.2.2, we employ two discriminators, the first D_r is applied to the right reconstructed images and the second D_l to the left reconstructed images. We perform an experiment where we apply a single discriminator D on both images. In Table 4.9, we observed that using two discriminators (D_r for I_r and D_l for I_l) is more effective. A possible explanation is that the reconstruction errors are different between the two synthesized images. Indeed, when generating the right image, the pixels located in the right side of the objects are not visible in the left image because of self-occlusion. Therefore errors are larger on the right of objects. Symmetrically, when synthesizing the left image, errors are larger on the left side of objects. The difference distribution of reconstruction errors can explain why separated discriminators work better. The second set of experiments concerning the use of disciminator focuses on the question of where the discriminators should be employed. Table 4.10 presents results obtained by using the discriminators at different locations on the KITTI dataset. These experiments illustrate the contribution of both the discriminators, on the first and the second *Half-Cycles*.

Ablation study: impact of feature alignment. In each stereo fusion layer, we align the feature maps as formulated in Eqn. (4.20) in order to avoid misalignment issues . We now present experiments to measure the impact of this design choice. We evaluate a variant of our model without using our proposed disparity and feature alignment. More precisely, instead of aligning the feature maps before concatenation, we simply concatenate the disparities and the feature maps in each *PFN* layer as in the the U-Net architecture. Results are reported in Table 4.11 for both KITTI and ApolloScape. These experiments demonstrate the impact of feature alignment

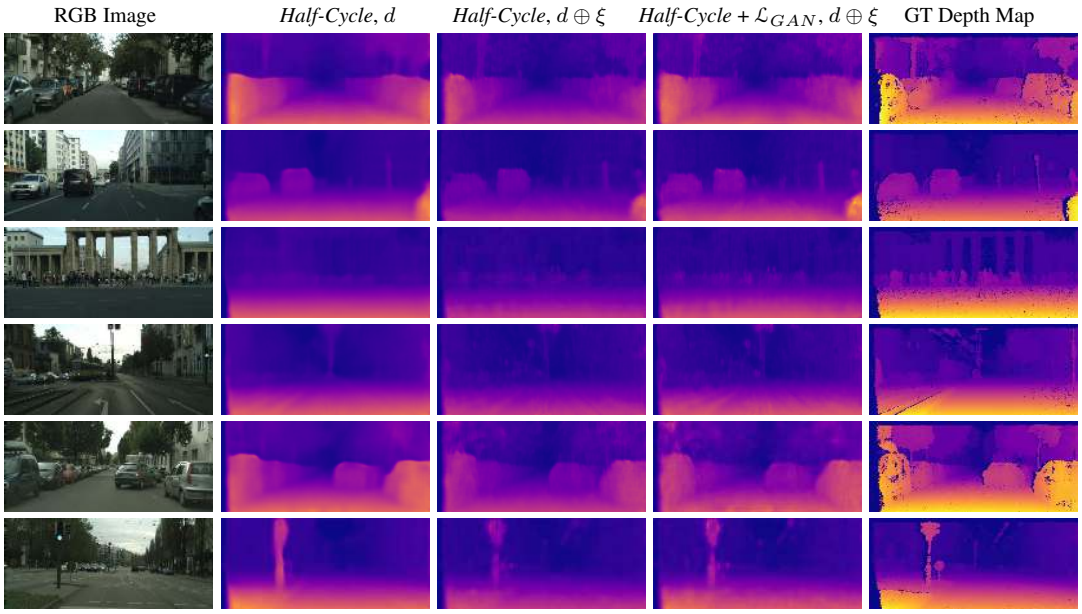


Figure 4.13: Qualitative comparison of different *Stereo Half-Cycle* models on the Cityscapes testing dataset. The second column presents progressive disparity fusion, they are in general smoother but don't present the level of detail that we can find in third and fourth column where we have progressive feature fusion. Columns three and four present results from models learned with adversarial loss.

and illustrate that a U-Net-based architecture performs better when handling feature misalignment.

4.2.3.4 Comparison with the State of the Art

In Table 4.12, we compare the proposed full model with several state-of-the-art methods, including the ones with the supervised setting, *i.e.* Saxena *et al.* [59], Eigen *et al.* [3], Liu *et al.* [21], AdaDepth [40], Kuznietzov *et al.* [33], Xu *et al.* [10], Jiang *et al.* [71], Gan *et al.* [72] and Guo *et al.* [73], and the ones with the unsupervised setting, *i.e.* Zhou *et al.* [35], AdaDepth [40], Garg *et al.* [4], DispNet [61], MADNet [31] and Godard *et al.* [5]. As far as we know, there are not quantitative results presented in the existing works on the Cityscapes dataset and for this reason, we perform the comparison on the KITTI dataset. These results further demonstrate the potential of unsupervised training for depth estimation. Note that we do not include the recent work in [75] in Table 4.12 as a different experimental setup is considered (different training/test split). Furthermore in [75] additional information (ego-motion

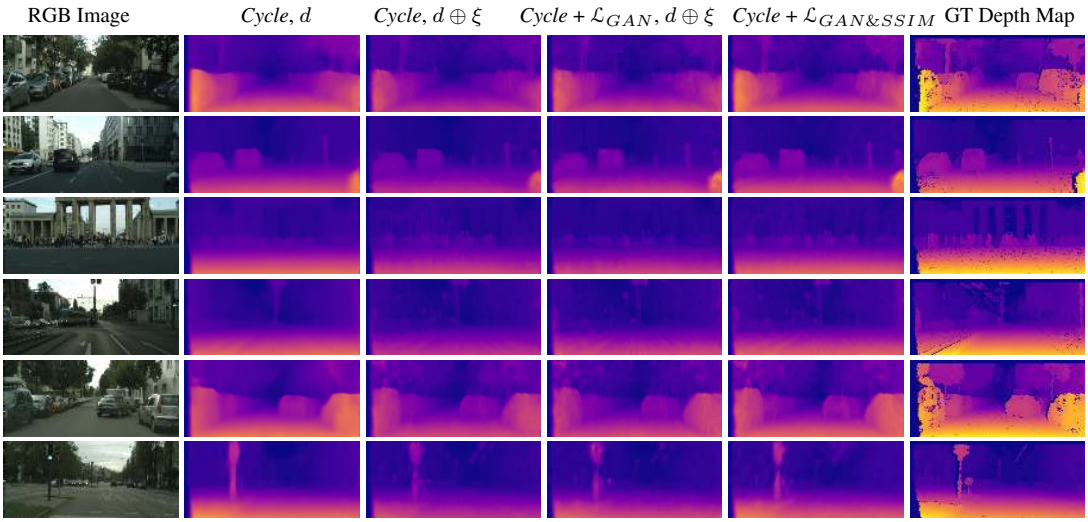


Figure 4.14: Qualitative comparison of different *Stereo Cycle* models on the Cityscapes testing dataset. Similarly to Fig. 4.13 in column two we present the results with progressive disparity fusion, while columns three, four and five have progressive features fusion.

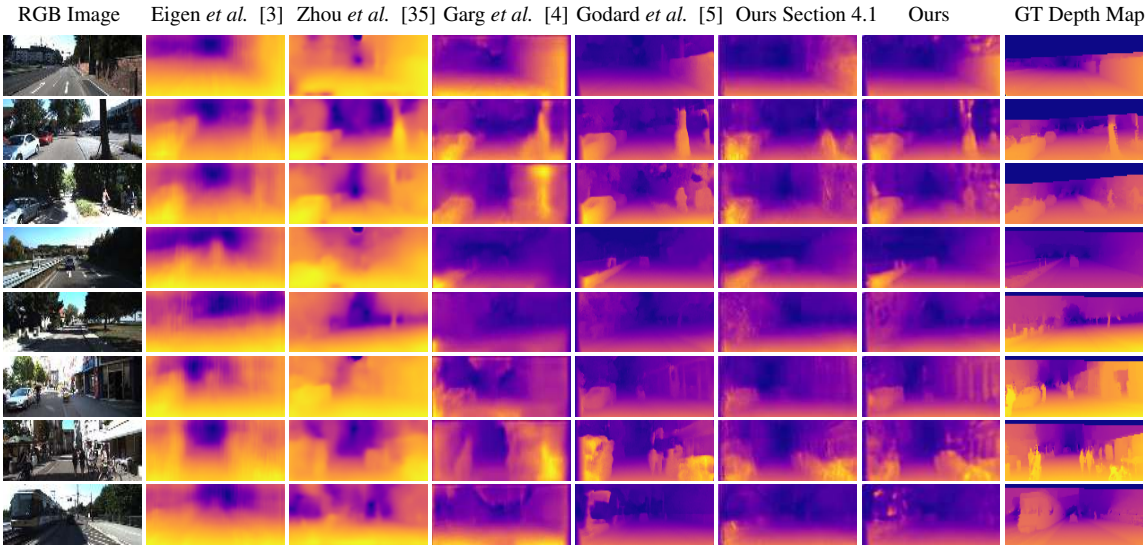


Figure 4.15: Qualitative comparison with different competitive approaches with both supervised and unsupervised settings on the Eigen test set of KITTI dataset. The sparse groundtruth depth maps are filled with bilinear interpolation for better visualization.

information) is exploited for depth prediction.

For comparison with the unsupervised methods, we outperform previous methods, according to four metrics: Abs Rel, Sq Rel, RMSE and RMSE log. In particular, we outperform

Method	Sup	Camera	Video	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
				lower is better				higher is better		
Saxena *et al.* [59]	Y	M	N	0.280	-	8.734	-	0.601	0.820	0.926
Eigen *et al.* [3]	Y	M	N	0.190	1.515	7.156	0.270	0.692	0.899	0.967
Liu *et al.* [21]	Y	M	N	0.202	1.614	6.523	0.275	0.678	0.895	0.965
AdaDepth [40], 50m	Y	M	N	0.162	1.041	4.344	0.225	0.784	0.930	0.974
Kuznietzov *et al.* [33]	Y	M	N	-	-	4.815	0.194	0.845	0.957	*0.987*
Xu *et al.* [10]	Y	M	N	0.132	0.911	-	*0.162*	0.804	0.945	0.981
Jiang *et al.* [71]	Y	M	N	0.131	0.937	5.032	0.203	0.827	0.946	0.981
Gan *et al.* [72]	Y	M	N	0.098	0.666	*3.933*	0.173	*0.890*	0.964	0.985
Guo *et al.* [73]	Y	M	N	*0.097*	*0.653*	4.170	0.170	0.889	*0.967*	0.986
DF-Net [74]	N	M	Y	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Godard *et al.* [38]	N	M	Y	*0.115*	*1.010*	*5.164*	*0.212*	*0.858*	*0.946*	*0.974*
Zhou *et al.* [35]	N	M	N	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Garg *et al.* [4]	N	M	N	0.169	1.08	5.104	0.273	0.740	0.904	0.962
Godard *et al.* [5]	N	M	N	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Godard Stereo [5]	N	S	N	0.109	1.120	5.013	0.205	**0.908**	**0.954**	0.973
Ours Section 4.1	N	S	N	0.152	1.388	6.016	0.247	0.789	0.918	0.965
DispNet [61]	N	S	N	0.126	0.919	4.733	0.200	0.885	**0.954**	**0.978**
MADNet [31]	N	S	N	0.118	1.090	4.926	0.213	0.896	**0.954**	0.973
PFN (Ours)	N	S	N	**0.102**	**0.802**	**4.657**	**0.196**	0.882	0.953	0.977
AdaDepth [40], 50m	N	M	N	0.203	1.734	6.251	0.284	0.687	0.899	0.958
Ours Section 4.1, 50m	N	S	N	0.144	1.007	4.660	0.240	0.793	0.923	0.968
DispNet [61], 50m	N	S	N	0.131	0.712	3.288	0.189	0.901	0.961	**0.982**
MADNet [31], 50m	N	S	N	0.112	0.753	3.648	0.200	**0.907**	**0.958**	0.976
PFN (Ours), 50m	N	S	N	**0.097**	**0.586**	**3.502**	**0.185**	0.893	0.957	0.979

Table 4.12: Comparison with the state of the art. Training and testing are performed on the KITTI [17] dataset. Supervised and semi-supervised methods are marked with Y in the supervision (Sup.) column, unsupervised methods with N. Monocular methods are marked M and binocular methods using stereo images at inference time are marked with S in the *Camera*. Methods using a frame sequence in input and, thus, exploiting temporal information, are marked with Y in the *Video* column. Numbers are obtained on Eigen test split with Garg image cropping. Depth predictions are capped at the common threshold of 80 meters, if capped at 50 meters we specify it. Best scores among static unsupervised methods are in bold. Best scores among other method categories are in italic.

the binocular methods DispNet [61], MADNet [31] and the method proposed by Godard *et al.* [5]. According to accuracy metrics, we are on par with theses recent approaches. These results illustrate the benefit of our approach. Note that we compared with DispNet [61], MADNet [31] since they are two recent architectures for stereo matching with a code that is publicly available and ready-to-use. Even though, these architectures are trained with supervision in the original

Method	Sup	Camera	Video	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
				lower is better				higher is better		
Eigen [3]	N	M	N	1.006	16.840	20.620	1.156	0.229	0.435	0.583
Godard [5]	N	M	N	0.432	12.199	14.497	0.426	0.591	0.832	0.911
Godard Stereo [5]	N	S	N	0.397	10.468	13.865	0.402	0.594	**0.848**	**0.933**
Ours Section 4.1	N	S	N	0.473	13.660	15.556	0.451	0.558	0.800	0.893
PFN (Ours)	N	S	N	**0.387**	**10.097**	**13.449**	**0.396**	**0.669**	0.843	0.915

Table 4.13: Comparison with the state of the art on ApolloScape. We compare out model trained on ApolloScape with those of Godard *et al.* [5] and Ours Section 4.1 using the code provided by the authors.

work, we report the performances obtained when training in the self-supervised setting. Concerning AdaDepth [40], we must mention that their approach is, to some extent, related to our approach since they employ adversarial learning in a context of domain adaptation with extra synthetic training data. Therefore, the better performance of our model illustrates the superiority of the proposed cycle-based data-augmentation approach compared to their use of synthetic data. When we consider the methods that use several frames at training and test time, only [38] performs better. Regarding ApolloScape dataset, to the best of our knowledge, we are the first to benchmark depth estimation methods on this dataset. We compare our approach with the competitive unsupervised models proposed in [5] and Section 4.1 using publicly available codes. Results are presented in Table 4.13, our proposed model improves by a large margin over the method presented in 4.1 and the monocular model of [5], and, importantly, it improves also with respect to the binocular model of [5] according to five metrics over seven.

The conclusion drawn in this quantitative comparison are confirmed by the qualitative evaluation reported in Fig 4.15. Compared to the previous Section 4.1, we see that our model estimates better the edges of the objects that appear in the image. For instance in the second row, we can distinguish better the shape of the trunk of the tree in foreground. The same remark stands for the reconstruction of the cars in the $2^{nd}, 3^{rd}, 4^{th}$ and 6^{th} rows. Comparing with [5], we distinguish similarly the edges of the objects but we estimate better the depth of large horizontal regions of the images. For instance, the depth of the road is much better estimated by our proposed model. This is especially true for the rows 1, 5 and 6 in which[5] underestimates the depth of the road. It can be explained by the difficulty of handling large displacements between the left and the right object when the image region does not contain many edges.

In addition to the several novelties presented in this Section with respect to our previous Section 4.1, the newly proposed model has fewer parameters and a lower training complexity. The best performing model of 4.1 consisted of seven main blocks, an encoder extracting the features from the images, four decoders, trained to reconstruct disparities and two discriminators, one for the right stereo view and one for the left stereo view. This complex model is trained iteratively to guarantee a good starting point for fine-tuning the network. Despite the good performance, the model proposed in our previous work has a complex optimization process.

4.2.4 Conclusions

We have presented a novel approach for unsupervised deep learning for the depth estimation employing a cycle structure. This new approach uses cycle consistency such that the network does not only learn from the training set images but also from the images generated in the first half-cycle. In addition, we proposed a generative deep network model specifically designed for binocular stereo depth estimation. By combining a refinement approach with a multi-scale strategy we improve the quality of the predicted depth map. It is worth noticing that although tested in the unsupervised setting, the proposed *PFN* can be also used in a supervised stereo scenario. In this work we decided to focus on the unsupervised setting because of the aforementioned practical advantages. However, monocular depth estimation methods can also benefit from the proposed adversarial approach. Extensive experiments were conducted on three publicly available datasets *i.e.* the popular KITTI, Cityscapes and ApolloScape datasets. Our results demonstrate the effectiveness of the proposed model, which is competitive with state of the art approaches for unsupervised depth estimation.

5

Structured Coupled Depth Estimation

After devising adversarial learning for self-supervised depth estimation in Chapter 4, we take a step forward in the use of adversarial learning for self-supervised depth estimation. We propose to couple the loss of discriminator and generator in a structured way. This is particulalrly useful in dense regression tasks for scene unserstanding because a lot of objects and features of the real world are characterized by precise shapes. The Conditional Random Field (CRF) coupling is devised to highlight these patterns and improve the optimization process.

5.1 Structured Coupled Generative Adversarial Networks for Unsupervised Monocular Depth Estimation [3]

Inspired by the success of adversarial learning, we propose a new end-to-end unsupervised deep learning framework for monocular depth estimation consisting of two Generative Adversarial Networks (GAN), deeply coupled with a structured Conditional Random Field (CRF) model. The two GANs aim at generating distinct and complementary disparity maps and at improving the generation quality via exploiting the adversarial learning strategy. The deep CRF coupling model is proposed to fuse the generative and discriminative outputs from the dual GAN nets. As such, the model implicitly constructs mutual constraints on the two network branches and between the generator and discriminator. This facilitates the optimization of the whole network for better disparity generation. Extensive experiments on the KITTI, Cityscapes, and Make3D

[3] "Structured Coupled Generative Adversarial Networks for Unsupervised Monocular Depth Estimation", M. M. Puscas, D. Xu, A. Pilzer and N. Sebe; 2019 International Conference on 3D Vision (3DV), Québec City, QC, Canada, 2019, pp. 18-26

datasets clearly demonstrate the effectiveness of the proposed approach and show superior performance compared to state of the art methods.

5.1.1 Introduction

Estimating scene depth from monocular images is a fundamental task in computer vision which can be potentially applied in various applications such as autonomous driving [76], Visual SLAM [77]. The main drawback of supervised-based systems is their dependence on costly depth-map annotations. As such, researchers have proposed unsupervised-based deep models using self-supervised view synthesis based on photometric error estimation [4, 5]. Within this pipeline, the quality of the view synthesis directly affects the performance of the final depth prediction. Adversarial learning has been introduced to improve the synthesis process in depth estimation systems [40, 65] by simply adding a frame-level discriminative loss for the image synthesis. However, the depth prediction maps and the discriminative error maps share meaningful structural information, e.g., objects in the input images are recognizable in both maps, and similar/close regions with higher generative errors tend to output higher discriminative errors. These structured relationships cannot be directly modeled in a standard GAN as the generator and the discriminator are not directly connected and thus do not explicitly flow gradients between them during the network optimization process. We argue that the discriminative and generative sub-networks hold complimentary structural information and jointly modeling it leads to a concurrent refinement of the produced discriminative error maps and the disparity maps used in the synthesis process, further improving the learned depth prediction model.

In this Chapter, we propose a structured adversarial deep model for unsupervised monocular depth estimation. The model consists of a dual generative adversarial network (DGAN), which takes stereo training images as input and performs image synthesis with the two branches containing separate generators and discriminators formulated as GANs [6]. The produced disparity maps are used to synthesize images from a single view, the complimentary stereo information is learned by a hallucinatory sub-network such that during inference the system can operate in a monocular fashion.

We further propose a deep CRF model to couple the network on two levels: We bind the two branches corresponding to each stereo image together, such that the complimentary stereo information is modeled. At the same time we model the complimentary structured information observed between the synthesized depth maps and discriminative error maps, through the

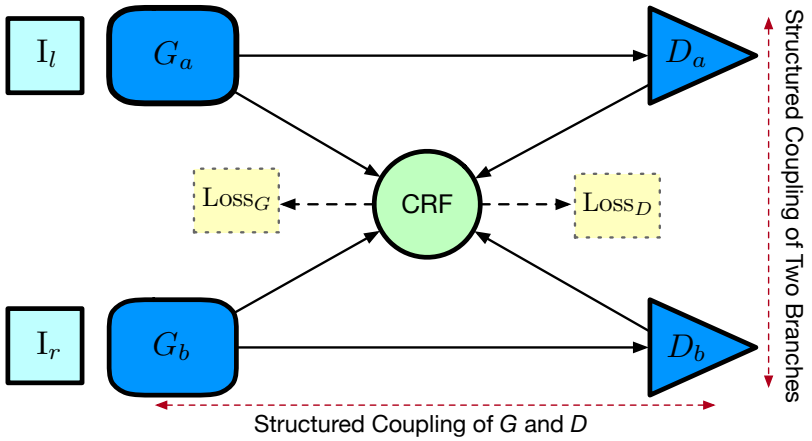


Figure 5.1: Illustration of the proposed structured coupling approach for adversarial monocular depth estimation. G_a, G_b and D_a, D_b denote generators and discriminators respectively.

linkage of the generative and discriminative sub-networks (see Fig. 5.1). This 2D linkage constrains the generative process through the use of a structured error, allowing for a structured refinement of the final synthesized depth map. The learning of the CRF model is thus jointly determined by the errors from both the generators and the discriminators.

We show how the proposed coupled CRF model can be solved with the mean-field theory, and present a neural network implementation for the CRF inference, enabling the model to be jointly optimized with the backbone DGAN in an end-to-end fashion. In the testing phase, only one single image is required. To summarize, our main contribution is threefold:

- We propose a novel CRF coupled Dual Generative Adversarial Network (CRF-DGAN) for unsupervised monocular depth estimation, which implicitly explores making the adversarial and structured learning benefit each other in an unified deep model for the task.

- Our model contains a dual GAN structure able to exploit the inherent relationship between stereo images to better learn the disparity maps. A coupled CRF model, implemented as a CNN, is presented to provide a structured fusion of the two sub-networks, as well as a structured connection between the discriminator and the generator.

- We conduct extensive experiments on the KITTI, Cityscapes, and Make3D datasets, clearly demonstrating the advantage of structured coupling in the designed dual GAN networks for the monocular depth estimation task. The proposed model is potentially useful for other GAN based applications possessing rich structural information. A very

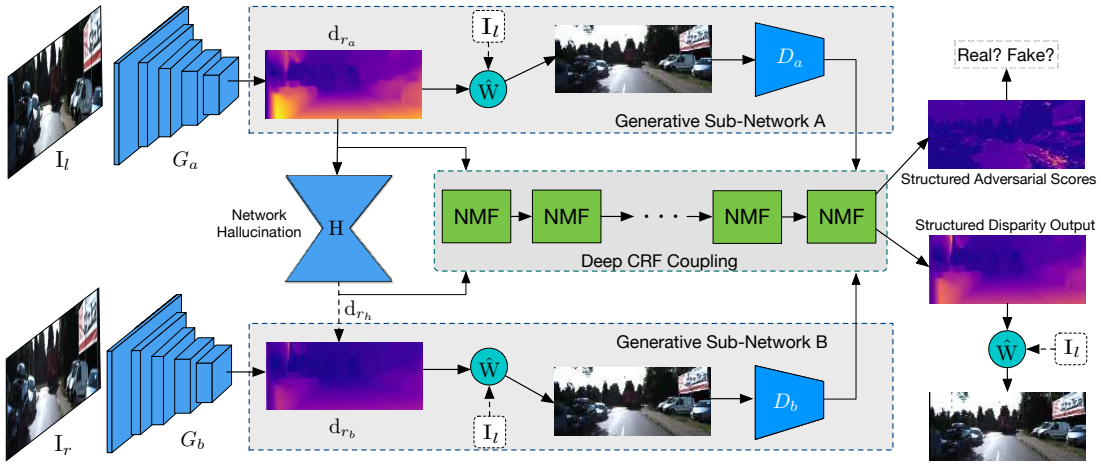


Figure 5.2: Framework overview of the proposed CRF-DGAN for unsupervised monocular depth estimation. $\hat{\mathrm{W}}$ is a warping operation to obtain a synthesized image. D_a and D_b are two discriminators corresponding to the two generative sub-networks. NMF denotes the neural network implementation of the continuous mean-field updating which composes the deep CRF model for structured coupling of the dual GANs. The training phase utilizes a pair of stereo images $\mathbf{I}_l$ and $\mathbf{I}_r$ as input, while in the testing phase, only one single image is required.

competitive performance is reported on KITTI as compared to state-of-the-art methods. The code will be made publicly available upon acceptance.

5.1.2 Proposed Approach

In this section, we present the proposed approach for unsupervised monocular depth estimation. A framework overview is depicted in Fig. 5.2. We first introduce the designed dual generative adversarial network, and then elaborate how we couple the two sub-networks upon both the generator and the discriminator, and perform a structured refinement of the outputs within a joint CRF model. Finally, we describe how the whole model can be organized into a unified deep network and can be simultaneously optimized in an end-to-end fashion.

5.1.2.1 Dual Generative Adversarial Networks

Basic Network Structure. As formalized in previous works [5, 34], unsupervised monocular depth estimation can be treated as a problem of learning a dense correspondence field between two calibrated image spaces. Given a set of N stereo image pairs $\{(\mathbf{I}_l^n, \mathbf{I}_r^n)\}_{n=1}^N$, the target is to learn a generator G which is able to estimate the dense correspondence (*i.e.* the disparity

map) $\mathbf{d}_r^n$ from $\mathbf{I}_l^n$ to $\mathbf{I}_r^n$, and the supervision is obtained from a reconstruction of $\mathbf{I}_r^n$ using a warping function f_w, *i.e.* $\tilde{\mathbf{I}}_r^n = f_w(\mathbf{d}_r^n, \mathbf{I}_l^n)$. The network can be optimized by minimizing the difference between $\mathbf{I}_r^n$ and $\tilde{\mathbf{I}}_r^n$. As shown in Fig. 5.2, we propose a dual generative adversarial network with a pair of stereo images $(\mathbf{I}_l^n, \mathbf{I}_r^n)$ as input in the training phase. The two generative networks G_a and G_b are designed to estimate two disparity maps $\mathbf{d}_{r_a}^n$ and $\mathbf{d}_{r_b}^n$ respectively, and part of shallow layers of G_a and G_b are shared to reduce the network capacity. Then two warping functions f_{w_a} and f_{w_b} are separately used to generate two synthesized right-view images via sampling from the same left-view image $\mathbf{I}_l^n$. Since $\mathbf{d}_{r_a}^n$ and $\mathbf{d}_{r_b}^n$ are generated from different inputs while similar images and the warping is performed on the the same image, the two disparity maps are well aligned and are complementary to each other. For the synthesized images, we use two discriminators D_a and D_b to benefit from the advantage of adversarial learning. To only learn the dual generative adversarial network, the optimization objective is:

$$\begin{aligned}
&\mathcal{L}_{gan}(G_a, G_b, D_a, D_b, \mathbf{I}_l^n, \mathbf{I}_r^n) = \\
&\mathbb{E}_{\mathbf{I}_r^n \sim p(\mathbf{I}_r^n)}[\log D_b(\mathbf{I}_r^n)] + \mathbb{E}_{\mathbf{I}_l^n \sim p(\mathbf{I}_l^n)}[\log(1 - D_b(G_a(\mathbf{I}_l^n)))] + \\
&\mathbb{E}_{\mathbf{I}_r^n \sim p(\mathbf{I}_r^n)}[\log D_a(\mathbf{I}_r^n)] + \mathbb{E}_{\mathbf{I}_r^n \sim p(\mathbf{I}_r^n)}[\log(1 - D_a(G_b(\mathbf{I}_r^n)))]
\end{aligned} \tag{5.1}$$

We adopt a sigmoid cross entropy to measure the expectation of the image $\mathbf{I}_l$ and $\mathbf{I}_r$ against the distribution $p(\mathbf{I}_l)$ and $p(\mathbf{I}_r)$ of the left- and right-view images respectively. Along with the adversarial objective, we have also an L_1 reconstruction objective for the generators:

$$\mathcal{L}_{rec}(G_a, G_b, \mathbf{I}_r^n, \mathbf{I}_l^n) = \| \tilde{\mathbf{I}}_r^n - \mathbf{I}_r^n \|_1 + \| \tilde{\mathbf{I}}_l^n - \mathbf{I}_r^n \|_1 \tag{5.2}$$

where $\tilde{\mathbf{I}}_l^n = f_w(\mathbf{d}_r^n, \mathbf{I}_l^n)$ and $\tilde{\mathbf{I}}_r^n = f_w(\mathbf{d}_l^n, \mathbf{I}_r^n)$ are the synthesized images with the disparity maps $\mathbf{d}_r^n$ and $\mathbf{d}_l^n$ estimated by the two generators G_a and G_b respectively.

Network Hallucination. Monocular depth estimation uses only a single image as input in the test phase. To achieve this, we designed a hallucination sub-network $H(\cdot)$ with a convolutional encoder-decoder structure, which aims at approximating the disparity map $\mathbf{d}_{r_b}^n$ using $\mathbf{d}_{r_a}^n$, *i.e.* $\mathbf{d}_{r_h}^n = H(\mathbf{d}_{r_a}^n, \mathbf{W}_h)$, where $\mathbf{W}_h$ are the parameters of the network H. In this way, the network H preserves the information coming from the image $\mathbf{I}_r^n$, while only the input image $\mathbf{I}_l^i$ is required in the testing. During the training we use an L_1 loss to optimize the network parameters $\mathbf{W}_h$ as follows: $\mathcal{L}_h(\mathbf{d}_{r_a}^n, \mathbf{d}_{r_b}^n, \mathbf{W}_h) = \sum_{n=1}^{N} \| H(\mathbf{d}_{r_a}^n, \mathbf{W}_h) - \mathbf{d}_{r_b}^n \|_1$. The proposed approach is general, if we have the stereo images in the testing phase, the network H can be disabled to support testing with stereo images.

5.1.2.2 Structured Coupling via Deep CRFs

Probabilistic graphical models such as conditional random fields (CRFs) have shown great success in supervised-based approaches [21, 78]. We investigate here how the CRF can be used for structured unsupervised monocular depth estimation. Since we have two generative adversarial networks, we propose a CRF coupling model for a structured fusion of the outputs of the two nets from both the generator and the discriminator. We first give the formulation of our model in coupling two disparity maps from the two generators, and then illustrate how this can also be done together with the two adversarial score maps.

Given the observed disparity maps $\mathbf{d}_{r_a}$ and $\mathbf{d}_{r_h}$ from the backbone network, let us denote $\mathbf{d}_r$ as a hidden disparity map to be inferred, and d_r^i is an element of $\mathbf{d}_r$ at position i (in analogy to $\mathbf{d}_{r_h}$ and $\mathbf{d}_{r_a}$). The model can be expressed as a Gibbs conditional distribution $P(\mathbf{d}_r|\mathbf{d}_{r_a}, \mathbf{d}_{r_h}, \mathbf{I}_r, \mathbf{\Theta}) = \exp(-E(\mathbf{d}_r|\mathbf{d}_{r_a}, \mathbf{d}_{r_h}, \mathbf{I}_r, \mathbf{\Theta}))/Z(\mathbf{I}_r, \mathbf{\Theta})$, where $\mathbf{\Theta}$ is a set of parameters and, E and Z are an energy and a normalization function, respectively. We formally define the energy in Eq. 5.3, where $\mathbf{f}_i$ and $\mathbf{f}_j$ are features calculated from the input image $\mathbf{I}_r$ at position i and j; $\alpha_1 > 0$ and $\alpha_2 > 0$ are weighting factors for the two unary terms; k_l is a gaussian kernel for similarity between the features:

$$\begin{aligned} E(\mathbf{d}_r|\mathbf{d}_{r_a},\mathbf{d}_{r_h},\mathbf{I}_r,\mathbf{\Theta})) = \\ \sum_i (\alpha_1(d_r^i - d_{r_a}^i)^2 + \alpha_2(d_r^i - d_{r_h}^i)^2) \\ + \sum_{i \neq j} \sum_l (\beta_l k_l(\mathbf{f}_i^{(l)}, \mathbf{f}_j^{(l)})(d_r^i - d_r^j)^2), \end{aligned} \tag{5.3}$$

For the unary term, an isotropic Gaussian function is used to describe the potential between the observation and the hidden disparity map, as a constrain that the hidden map to be as close as possible to the observation ones. For the pairwise term, following [79] we use both an appearance and a smoothness kernel (*i.e.* for l=1, 2 then β_l are weights for the kernels) to have structured constraints on the hidden disparity map.

Inference. Exact inference of the fully connected model requires high complexity because of the calculation of inverse matrices [21, 80]. We approximate the inference using mean-field theory. The target is to approximate the distribution $P(\mathbf{d}_r|\mathbf{d}_{r_a}, \mathbf{d}_{r_h})$ with another simpler distribution $Q(\mathbf{d}_r|\mathbf{d}_{r_a}, \mathbf{d}_{r_h})$ which can be expressed with a set of independent marginal distributions, *i.e.* $Q(\mathbf{d}_r|\mathbf{d}_{r_a}, \mathbf{d}_{r_h}) = \prod_i Q_i(d_r^i|\mathbf{d}_{r_a}, \mathbf{d}_{r_h})$. We obtain an optimal solution $\tilde{Q}$ by minimizing the KL divergence between the distribution P and Q, *i.e.* $\log \tilde{Q}_i(d_r^i|\mathbf{d}_{r_a}, \mathbf{d}_{r_h}) =$

$\mathbb{E}_{j\neq i}[\log P(\mathbf{d}_r|\mathbf{d}_{r_a},\mathbf{d}_{r_h})]+\mathrm{C}$ with C as a const. The mean-field inference for Q can be derived as follows:

$$\tilde{Q}_i(d_r^i) \propto \exp\big(-(\alpha_1+\alpha_2){d_r^i}^2 + 2d_r^i(\alpha_1 d_{r_a}^i + \alpha_2 d_{r_h}^i) - \sum_l \beta_l k_l(\mathbf{f}_i^{(l)},\mathbf{f}_j^{(l)})({d_r^i}^2 - 2d_r^i d_r^i])\big). \quad (5.4)$$

The equation implies that the log distribution of $\tilde{Q}_i$ takes a Gaussian distribution and its expectation produces the maximum probability. Then we have the mean-field updating for the continuous hidden variable d_r^i written as

$$d_r^i = \frac{\alpha_1 d_{r_a}^i + \alpha_2 d_{r_h}^i + \sum_l \sum_{j\neq i} k_l(\mathbf{f}_i^{(l)},\mathbf{f}_j^{(l)}) d_r^j}{\alpha_1+\alpha_2+\sum_l \sum_{j\neq i} k_l(\mathbf{f}_i^{(l)},\mathbf{f}_j^{(l)})} \quad (5.5)$$

The updating of d_r^i is an iterative operation, and we are able to achieve a local minimum after T iterations. In the following we discuss how we implemented the continuous mean-field in neural network (NMF) for the inference of the hidden variables, enabling a joint end-to-end optimization with the proposed backbone dual GAN network.

Mean-Field Updating in Neural Network (NMF). In Eq. 5.5, we have three steps to perform the mean-field updating. The first step is a linear combination of the unary terms, *i.e.* $\alpha_1 d_{r_a}^i + \alpha_2 d_{r_h}^i$, which can be implemented with 1×1 convolutions with a ReLU operation, and then an element-wise addition operation. The second step is the message passing. To calculate the message with the Gaussian convolution operation, *i.e.* $\sum_{j\neq i} k_l(\mathbf{f}_i^{(l)},\mathbf{f}_j^{(l)}) d_r^j$, due to the high complexity, we utilize a local receptive field considering a locally connected graph. The message passing can be performed using element-wise addition operation. The third step is a normalization step. The calculation of the normalization factor (*i.e.* the denominator) is similar to that of the previous steps, and an element-wise division operation is used to perform the normalization. We have in total four parameters to optimize, *i.e.* two linear combination weights for the observation maps d_{r_a} and d_{r_h}, and other two weights for the gaussian kernels. Since each forward step is differentiable, the mean-field updating can be optimized with the back-propagation, and we can stack several mean-field blocks by sharing parameters for a deep CRF inference.

Joint Coupling of the Generator and Discriminator. To model the structured relationship between the generator and discriminator, we use one single CRF model to learn the fusion and refinement of both. The discriminators and the generators from the dual GAN produce the

Method	Error (lower is better)				Accuracy (higher is better)		
	rel	sq rel	rms	rms log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
CRF-DGAN (baseline model)	0.1650	1.7563	6.164	-	0.773	0.914	0.962
CRF-DGAN (w/ deep network hallucination)	0.1617	1.4834	5.991	0.242	0.779	0.917	0.964
CRF-DGAN (w/ adversarial learning)	0.1528	1.4005	6.029	0.247	0.785	0.918	0.965
CRF-DGAN (w/ coupled adversarial learning)	0.1423	1.3067	5.687	0.238	0.813	0.928	0.968
CRF-DGAN (w/ dual coupled adversarial learning)	**0.1407**	**1.2831**	**5.677**	**0.237**	**0.815**	**0.930**	**0.968**

Table 5.1: Quantitative analysis of the main components of our method on the KITTI dataset. The evaluation is conducted on the predicted depth maps following the standard evaluation protocol.

same number of outputs, *i.e.* two disparity maps and correspondingly two real/fake adversarial score maps, where we consider a pixel-level discriminator. Then we respectively input them into the deep CRF coupling model introduced above with two separate forward computations, and collect gradients from both to perform one backward computation to update the model parameters during learning. By doing so, the outputs from the generators and the discriminators will jointly affect the model learning, contributing implicitly as mutual constraints to better optimize both parts. Fig. 5.3 shows examples of structured outputs of the generated disparity and the discriminative errors. We use only one adversarial loss using the refined and fused the adversarial score map from the deep coupling CRF model. Let us denote $D^{\text{crf}}_{\text{real}}$ and $D^{\text{crf}}_{\text{fake}}$ as the adversarial score for the real and fake samples, and thus we replace Eq. 5.1 as:

$$\begin{aligned}&\mathcal{L}^{\text{crf}}_{gan}(G_a, G_b, D_a, D_b, \mathbf{I}^n_l, \mathbf{I}^n_r) = \\ &\mathbb{E}_{\mathbf{I}^n_r \sim p(\mathbf{I}^n_r)}[\log D^{\text{crf}}_{\text{real}}] + \mathbb{E}_{\mathbf{I}^n_l, \mathbf{I}^n_r \sim p(\mathbf{I}^n_l, \mathbf{I}^n_r)}[\log(1 - D^{\text{crf}}_{\text{fake}})].\end{aligned} \tag{5.6}$$

End-to-End Joint Optimization. The learning of the whole network involves optimization of both the dual generative adversarial network and the deep CRF model. For the CRF model, the final output disparity map is used to synthesize another right-view image $\mathbf{d}^n_{rc}$, and we use an l_1 reconstruction loss $\mathcal{L}_{crf}$ to supervise the learning of the CRF model with $\mathcal{L}_{crf} = \sum_{n=1}^{N} ||\mathbf{I}_{rc,n} - \mathbf{I}_{r,n}||_1$. To combine the loss functions of the dual generative adversarial network, the whole deep network optimization objective becomes: $\mathcal{L}_o = \gamma_1 \mathcal{L}_{rec} + \gamma_2 \mathcal{L}_h + \gamma_3 (\mathcal{L}^{\text{crf}}_{gan} + \mathcal{L}_{crf})$, where $\{\gamma_i\}^3_{i=1}$ is a set of weights for balancing the loss from different parts.

5.1.3 Experimental Results

We now present the experimental setup and results to demonstrate the effectiveness of the proposed approach.

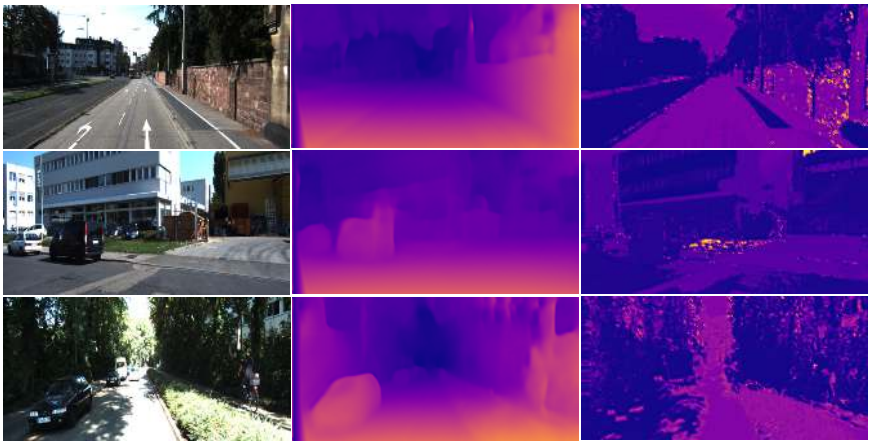


Figure 5.3: Examples of the structured output of the disparity maps and the adversarial score maps on KITTI using the proposed CRF-DGAN. The CRF model couples not only two GAN sub-networks but also connects the generators and the discriminators with mutual constraints in joint optimization.

Method	Error (lower is better)				Accuracy (higher is better)		
	rel	sq rel	rms	rms log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
CRF-DGAN (baseline model)	0.4676	7.3992	5.741	0.493	0.735	0.890	0.945
CRF-DGAN (w/ deep network hallucination)	0.4397	6.3369	5.444	0.456	0.730	0.887	0.944
CRF-DGAN (w/ adversarial learning)	0.4327	6.2006	5.541	0.424	0.738	0.890	0.944
CRF-DGAN (w/ coupled adversarial learning)	**0.4109**	**5.9848**	**4.636**	**0.403**	**0.756**	**0.897**	**0.953**

Table 5.2: Quantitative analysis of the main components of our method on the Cityscapes dataset. Cityscapes does not provide a standard evaluation protocol for depth estimation. We directly evaluate the performance on the predicted disparity maps.

5.1.3.1 Experimental Setup

Datasets. We have conducted experiments on the KITTI [66], Cityscapes [18] and Make3D [16, 59] datasets. The **KITTI** dataset contains depth images captured with a LiDAR sensor mounted on a driving vehicle. In our experiments we follow the experimental protocol proposed by [3] containing 22,600 training images and 697 images test images. The RGB image resolution is reduced by half with respect to the original 1224×368 pixels. To evaluate the transfer learning capabilities of our method, we test the model trained on Cityscapes and evaluate it on the **Make3D** dataset, which contains only 400 single training RGB and depth map pairs, and 134 test samples. The **Cityscapes** is a large-scale dataset mainly used for semantic urban scene

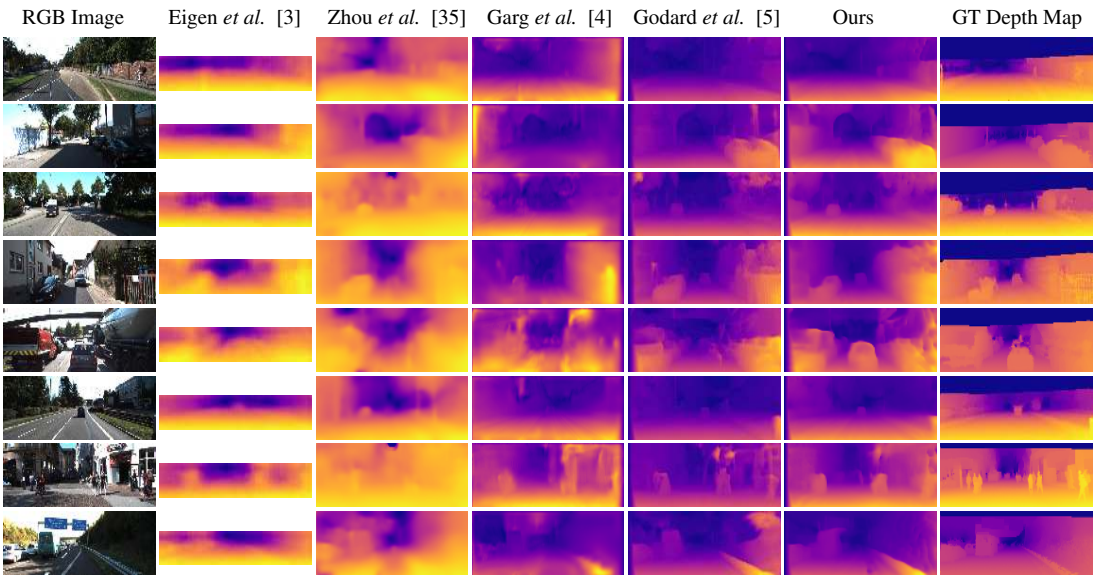


Figure 5.4: Examples of depth prediction results on the KITTI raw dataset. Qualitative comparison with other depth estimation methods on this dataset is presented. The sparse ground-truth depth maps are interpolated for better visualization.

understanding. The annotated split contains 2975 training, 500 validation, and 1525 test images. The dataset also provides pre-computed disparity maps associated with the rgb images. As the images of the dataset have a high resolution (2048×1024), we resize the image to size of 512×256 as in [5] for training due to the limitation of the GPU memory, and the bottom one fifth of the image is removed.

Implementation Details. Messages are passed via locally connected convolutions *i.e.* considering a local receptive field for the Gaussian convolution with a kernel window size of 15×15. In our CRF model we consider dependencies only for the last scale. The initial learning rate is set to 1e-4 in all our experiments, and decreases 5 times after for each step reached in $[30000, 55000]$. The momentum and weight decay parameters are set to 0.9 and 0.0002, as in [81]. The batch size of the algorithm is set to 8.

5.1.3.2 Experimental Results

We first conduct an in-depth analysis of the proposed approach, and then carry out a state-of-the-art comparison with other competing methods, and finally provide a discussion on the qualitative results.

5.1 Structured Coupled Generative Adversarial Networks for Unsupervised Monocular Depth Estimation [3]

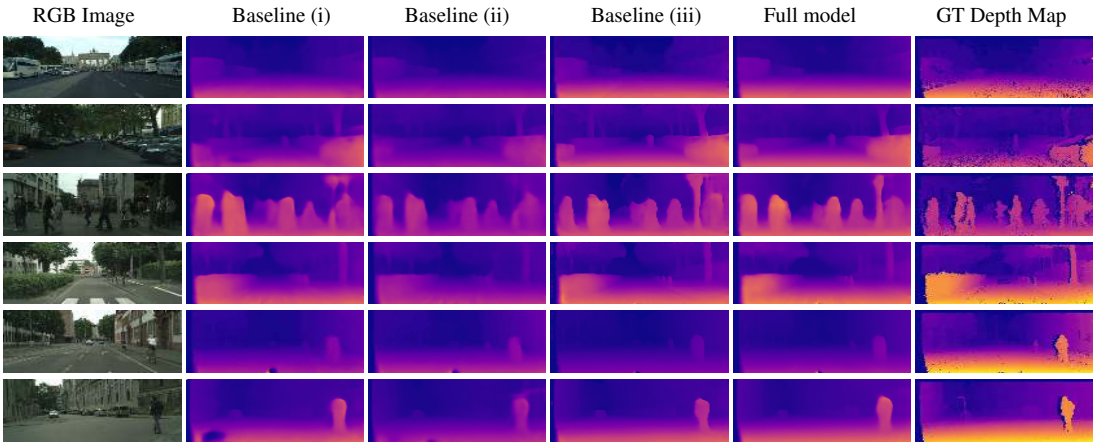


Figure 5.5: Qualitative comparison of different variants of the proposed CRF-DGAN model on the Cityscapes dataset.

Method	Setting		Error (lower is better)				Accuracy (higher is better)		
	cap	supervised?	rel	sq rel	RMSE	RMSE (log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena *et al.* [16]	80m	√	0.280	-	8.734		0.601	0.820	0.926
Eigen *et al.* [3]	80m	√	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al.* [21]	80m	√	0.202	1.614	6.523	0.275	0.678	0.895	0.965
AdaDepth [40]*	50m	√	0.162	1.041	4.344	0.225	0.784	0.930	0.974
Kuznietsov *et al.* [33]	80m	√	-	-	4.815	0.194	0.845	0.957	0.987
Xu *et al.* [26]	80m	√	0.120	0.764	4.341	0.181	0.852	0.959	0.987
Gan *et al.* [72]	80m	√	0.098	0.666	3.933	0.173	0.890	0.964	0.985
Garg *et al.* [4]	80m	×	0.177	1.169	5.285	0.282	0.727	0.896	0.962
Garg *et al.* [4] L12 + Aug 8x	50m	×	0.169	1.080	5.104	0.273	0.740	0.904	0.958
Godard *et al.* [5]	80m	×	0.148	1.344	5.927	0.247	0.803	0.922	0.963
Kuznietsov *et al.* [33]	80m	×	-	-	8.700	0.367	0.752	0.904	0.952
Zhou *et al.* [35]	80m	×	0.208	1.768	6.858	0.283	0.678	0.885	0.957
AdaDepth [40]	50m	×	0.203	1.734	6.251	0.284	0.687	0.899	0.958
Mahjourian *et al.* [36]†	80m	×	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Pilzer *et al.* 4.1	80m	×	0.152	1.388	6.016	0.247	0.789	0.918	0.965
Wang *et al.* [62]	80m	×	0.151	1.257	5.583	**0.228**	0.810	**0.936**	**0.974**
Zou *et al.* [74]†	80m	×	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Zhan *et al.* [34]†	80m	×	0.144	1.391	5.869	0.241	0.803	0.933	0.971
Guo *et al.* [73]*	80m	×	0.105	0.811	4.634	0.189	0.874	0.959	0.982
CRF-DGAN (ours)	80m	×	**0.1354**	**1.1815**	**5.582**	0.235	**0.828**	0.933	0.967
CRF-DGAN (ours)	50m	×	**0.1283**	**0.8681**	**4.223**	**0.222**	**0.840**	**0.941**	**0.971**

Table 5.3: State of the art comparison on the KITTI dataset. Methods that require additional image data are marked with *, and those that require video data are marked with †. We bold the metrics where our method achieves the best results under the same settings.

Baseline Models. We mainly aim to demonstrate the effectiveness of the proposed approach from three aspect: first, the monocular depth estimation with adversarial learning strategy, second, the proposed dual GAN network structure, and third, the coupling scheme to fuse and refine the proposed dual GAN in a structured fashion. Thus we present an ablation study based on several baselines, including **(i)** CRF-DGAN (baseline model): a single branch model which uses only the generator without using the adversarial loss; **(ii)** CRF-DGAN (w/ deep network hallucination): a dual-branch model with network hallucination, which has two branches each synthesizing a right view new image, and sharing the parameters of the encoder part. The dual-branch model is used as the backbone network structure of our approach. A hallucinator is added in order to predict images in a monocular fashion in the testing phase; **(iii)** CRF-DGAN (w/ adversarial learning): we train the backbone adversarially, *i.e.* adding a discriminator per branch; **(iv)** CRF-DGAN (w/ coupled adversarial learning): the two discriminators of the dual-GAN are coupled with the proposed CRF model; **(v)** CRF-DGAN (w/ dual coupled adversarial learning): both the discriminators and the generators are coupled with the proposed CRF model.

Model Analysis. We conduct the ablation study on the KITTI raw and Cityscapes datasets, as shown in Table 5.1 and 5.2. Comparing baseline (i) and (ii), we observe a minor improvement in absolute error, but a more substantial improvement in all accuracy metrics, especially on Cityscapes dataset. This performance boost is likely caused by the network hallucination learning the complimentary information between the two stereo viewpoints, resulting in a better learned model. The effectiveness of adversarial learning has been demonstrated in previous chapters for GAN-based depth estimation, a benefit also observed between the baseline models (iii) and (ii). Baseline models (iv) and (v) evaluate the effectiveness of the proposed CRF model using different coupling strategies, between the disparity maps produced by the generators and the adversarial score maps by the discriminators. By comparing (v) and (iii), we have 1.2 points gain on the metric rel on KITTI. We should note that this is not a trivial gain on this very challenging and almost performance-saturated dataset. From the accuracy aspects: we improve 4 points from 0.779 to 0.813, clearly demonstrating the effectiveness of the proposed CRF-based structured coupling approach. We observe a more significant boost (around 2.7 points) on the rel metric on the Cityscapes dataset, and that coupling both the discriminators and the generators achieves better performance than coupling only the discriminators, meaning that the joint coupling brings extra constraints for each part, and facilitates the network optimization, confirming our initial motivation. In overall, our approach has 2.5 points gain on the

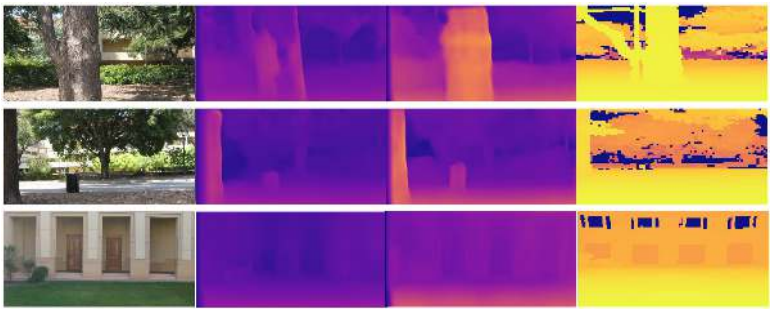


Figure 5.6: Examples of depth prediction results on the Make3D dataset. Qualitative comparison between structured and non-structured disparity maps is presented. From left to right, RGB image, non-structured disparity prediction, structured disparity prediction, ground truth depth. Structured disparities show improved prediciton over non-structured disparities.

difficult rel metric over the single branch basic baseline model, demonstrating the effectiveness of the CRF coupled dual GAN structure.

State-of-the-art Comparison. Table 5.3 compares the depth estimation results of our full CRF-DGAN model with other supervised and unsupervised methods. We outperform most competitors due to a joint structured optimization of both the discriminators and the generators sign. The concise network design also facilitates the overall optimization process. With regard to [4], we also report results for a 50m depth cap. The full CRF-DGAN model achieves better performance in both the 50m and 80m settings. Of interest is that CRF-DGAN outperforms [5] in all metrics. Our performance is much better than AdaDepth which also considers generative adversarial networks while used extra synthetic training data. [34], DF-Net [74], and [36] do not use the same setting as our approach, requiring video training data for extra temporal information. In contrast, CRF-DGAN requires only image pairs in training and single images in testing. Although our approach is not directly comparable to them, it outperforms their results on all the metrics. Our approach is outperformed by [72], which uses stereo-matching techniques to improve upon available sparse LiDaR ground truth. As so it is a method with a different setting to ours and is not directly comparable. [73] uses a stereo model trained on the KITTI raw and then synthetic SceneFlow [61] are used to distil a monocular model reaches higher performance than CRF-DGAN, but it requires both a large amount of additional stereo data for training and a more complex optimization process.

Qualitative Analysis. The performance can be qualitatively observed in Fig. 5.4 and 5.5 for KITTI and Cityscapes, respectively. The advantage of structured modeling between the generator and the discriminator can be observed in Fig. 5.4, where our method is able to capture

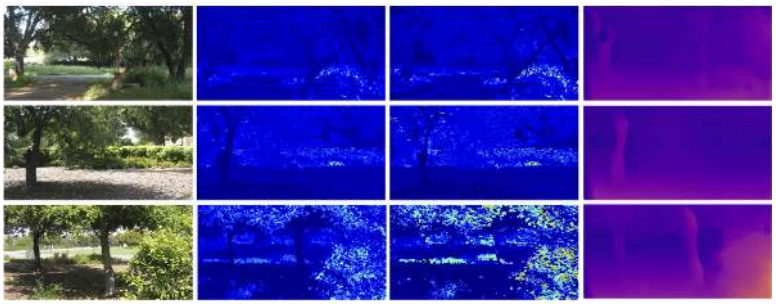


Figure 5.7: Examples of structured outputs of the real and the fake discriminative score-maps on the Make3D dataset, with the associated depth predictions. From left to right, RGB image, discriminator predictions for RGB image, discriminator prediction for synthesized RGB image and depth prediction. The discriminator infers higher error

object details as well as objects in their entirety. Furthermore, we qualitatively evaluate a model learned on the Cityscapes dataset and tested on the Make3D dataset. The results are shown in Figure 5.6. The importance of adding structural information when inferring on unfamiliar data can be clearly observed. Conditioning on the input images allows the approach to maintain a good detail consistency. Figure 5.7 shows the structured output of the discriminative score-maps generated from associated real and synthesized samples. The areas in which the synthesized disparity values with low accuracy produce a high discriminative error. Fig. 5.5 showcases different variants of the proposed CRF-DGAN approach and the improvement in quality.

Discussion on the Time Aspect. On a single Titan V-100, with a batch size of 4, the model can infer 6 images with a resolution of 512×256 per second, which is near real-time speed. Further performance improvements in speed can be achieved through decreasing the size of the CRF receptive field and also consider approximation approaches in the expensive Gaussian convolutional operations, *e.g.* permutohedral lattice algorithm [82].

5.1.4 Conclusions

We have presented an end-to-end unsupervised deep learning framework for monocular depth estimation. The proposed framework consists of two generative adversarial sub-networks, aiming at on one hand generating distinct while complementary disparity maps, through accepting images from different views as input, and on the other hand, improving the generation quality via exploiting the adversarial learning strategy. We couple the dual-GAN by a deep CRF model,

which is able to perform structured refinement and fusion of the predicted disparity maps from the generators and the adversarial scoremaps from the discriminators. The deep CRF coupling also makes the discriminator and the generator explicitly constrain on each other, and thus facilitates the optimization of the whole network for better disparity generation. We conducted extensive experiments on the challenging KITTI, Cityscapes, and Make3D datasets, clearly demonstrating the effectiveness of the proposed approach.

6

Monocular Depth Refinement

Chapters 4 and 5, presented stereo binocular methods. Despite suggesting in Chapter 5, a *hallucinator module* to avoid using stereo images at testing time, the method is still based on stereo a setting. Improving on that we devise a model that at inference time is totally monocular and more interestingly has the ability to self-improve the predictions at inference time, without using any ground-truth data.

6.1 Refine and Distill: Exploiting Cycle-Inconsistency and Knowledge Distillation for Unsupervised Monocular Depth Estimation [4]

Nowadays, the majority of state of the art monocular depth estimation techniques are based on supervised deep learning models. However, collecting RGB images with associated depth maps is a very time consuming procedure. Therefore, recent works have proposed deep architectures for addressing the monocular depth prediction task as a reconstruction problem, thus avoiding the need of collecting ground-truth depth. Following these works, we propose a novel self-supervised deep model for estimating depth maps. Our framework exploits two main strategies: refinement via cycle-inconsistency and distillation. Specifically, first a *student* network is trained to predict a disparity map such as to recover from a frame in a camera view the associated image in the opposite view. Then, a backward cycle network is applied to

[4] "Refine and Distill: Exploiting Cycle-Inconsistency and Knowledge Distillation for Unsupervised Monocular Depth Estimation", A. Pilzer, S. Lathuilière, N. Sebe, E. Ricci; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9768-9777

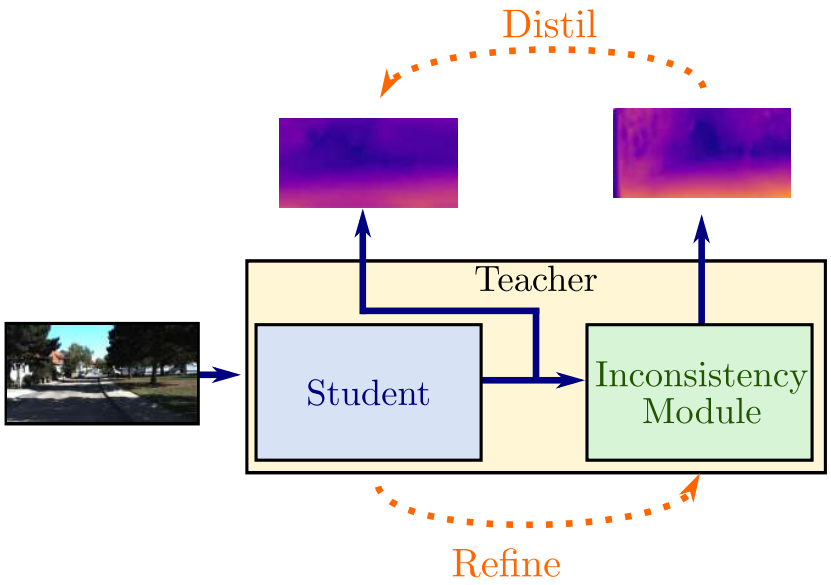


Figure 6.1: Outline of the proposed approach: from the right view image, we predict the left image from which we re-synthesize the right image. The inconsistencies are used by the inconsistency-module to improve the depth estimation. The refined depth maps are used to improve the Student Network via knowledge distillation.

the generated image to re-synthesize back the input image, estimating the opposite disparity. A third network exploits the inconsistency between the original and the reconstructed input frame in order to output a refined depth map. Finally, knowledge distillation is exploited, such as to transfer information from the refinement network to the student. Our extensive experimental evaluation demonstrate the effectiveness of the proposed framework which outperforms state of the art unsupervised methods on the KITTI benchmark.

6.1.1 Introduction

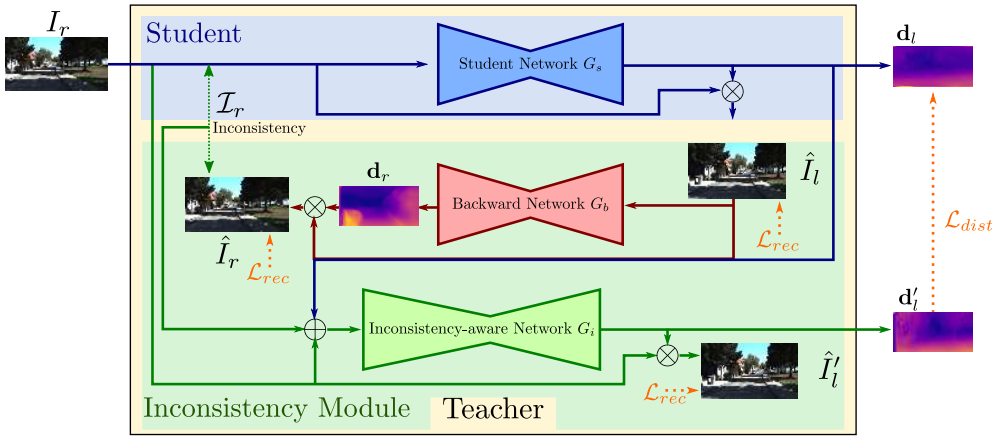
In the last few years, deep learning-based approaches for depth estimation [3, 5, 10, 21, 35, 36, 58, 65] have attracted a growing interest, motivated, on the one hand, by their ability to predict very accurate depth maps and, on the other hand, by the importance of recovering depth information in several applications, such as robot navigation, autonomous driving, virtual reality and 3D reconstruction.

Exploiting the availability of very large annotated datasets, Convolutional Neural Networks (ConvNets) trained in a supervised setting are now state-of-the-art in many computer vision tasks such as object detection [83], instance segmentation [84], human pose estimation [85].

However, a major weakness of these approaches is the need of collecting large-scale labeled datasets. In the case of depth estimation, acquiring data is especially costly. For instance, in the scenario of depth estimation for autonomous driving, it implies driving a car equipped with a laser LiDaR scanner for hours under diverse lighting and weather conditions. Self-supervised depth estimation, also referred to as unsupervised, recently emerged as an interesting paradigm and an effective alternative to supervised methods [4, 5, 60, 61, 62]. Roughly speaking, in the self-supervised setting, stereo image pairs are considered as input and a deep predictor is learned in order to estimate the associated disparity maps. Specifically, the predicted disparity is employed to synthesize, from a frame in a camera view (*e.g.* from the left camera), the opposite view through warping. The deep network is trained via gradient descent by minimizing the discrepancy between the original and the reconstructed image. Importantly, even if stereo images pairs are required for training, depth can be recovered from a single image at test time.

In this Chapter, we follow this research thread and propose a novel self-supervised deep architecture for monocular depth estimation. The proposed approach, illustrated in Fig 6.1, consists of a first sub-network, referred to as the *student* network, which receives as input an image from a camera view and predicts a disparity map such as to recover the opposite view. On top of this network, we propose several contributions. First, from the generated image, we propose to re-synthesize the input image by estimating the opposite disparity. The resulting network forms a cycle. Second, a third network exploits the cycle inconsistency between the original and the reconstructed input images in order to refine the estimated depth maps. Our intuition is that inconsistency maps provide rich information which can be further exploited, as they indicate where the first two networks fail to predict disparity pixels. Finally, we propose to use the principle of distillation in order to transfer knowledge from the whole network, seen as a *teacher*, to the *student* network. Interestingly, our framework produce two outputs, corresponding to the depth maps estimated respectively by the *student* and the *teacher* networks. This is extremely relevant in practical applications, as the *student* network can be exploited in case of low computation power or real-time constraints.

Our extensive experiments on two large publicly available datasets, *i.e.* the KITTI [66] and the Cityscapes [18] datasets, demonstrate the effectiveness of the proposed framework. Notably, by combining the proposed cycle structure with our inconsistency-aware refinement, our unsupervised framework outperforms previous usupervised approaches, while obtaining comparable results with the state-of-the-art supervised methods on the KITTI dataset.



.

Figure 6.2: The proposed approach is composed of two modules. A first network G_s predicts the right-to-left disparity map d_l from the right image and synthesizes the left image as described in Sec. 6.1.2.4. In the second module, a generator network G_b predicts the left-to-right disparity map d_r in order to re-synthesize the right image. The model obtained in this way forms a cycle. The cycle inconsistency is used by a third network to predict the final disparity map. We use a set of losses (orange dot arrows) detailed in Sec. 6.1.2.4

6.1.2 Proposed Approach

6.1.2.1 Overview

The aim of this chapter is to estimate the depth of a scene from a single image. However, at training time, we consider that we dispose of pairs of images $\{\mathbf{I}_l, \mathbf{I}_r\}$ of size $H \times W$, derived from a stereo pair and corresponding to the same time instant. Here, $\mathbf{I}_l$ denotes the left camera view and $\mathbf{I}_r$ is the right camera view. Given $\mathbf{I}_r$, we are interested in predicting a correspondence map $\mathbf{d}_l \in \mathbb{R}^{H \times W}$, namely the right-to-left disparity, in which each pixel value represents the offset of the corresponding pixel between the right and the left images. Finally, assuming that the images are rectified, the depth at a pixel location (x, y) of the left image can be recovered from the predicted disparity with $d_l = \frac{f.b}{d(x,y)}$, where b is the distance between the two cameras and f is the camera focal length.

An overview of the proposed framework is shown in Fig. 6.2. A first network G_s predicts the right-to-left disparity map $\mathbf{d}_l$ from the right image $\mathbf{I}_r$, and synthesizes the left image by warping $\mathbf{I}_r$ according to $\mathbf{d}_l$. Roughly speaking, the network G_s is trained to minimize the discrepancy between the real and the reconstructed left image (Sec. 6.1.2.4).

We employ a second generator network G_b that takes as input the synthesized left image and

predicts a left-to-right disparity map $\mathbf{d}_r$ that is used to re-synthesize the right image. The model obtained in this way forms a cycle. This cycle design has three advantages. First, at training time, by sharing weights between G_s and G_b, the networks learn to predict disparity maps from the images of the training set (in the forward half-cycle G_s) but also from the synthesized images (in the backward half-cycle G_b). In that sense, the use of the cycle can be seen as a sort of data augmentation. Second, in order to re-synthesize correctly the right image, the second network G_b requires a correct input left image. Thus, G_b imposes a global constraint on the estimated disparity $\mathbf{d}_l$ oppositely to standard pixel-wise discrepancy losses, such as $\mathcal{L}_1$ or $\mathcal{L}_2$ that act only locally. Third, by comparing the input right image $\mathbf{I}_r$ and the output right image $\hat{\mathbf{I}_r}$ synthesized after applying our cycle framework, we can measure the cycle inconsistency. At a given location of the input image, if we observe no inconsistency, G_s and G_b must have predicted correctly the disparity maps. Conversely, in case of inconsistency, G_s or G_b (or both) must have predicted incorrectly the disparity maps. Note that inconsistencies may also appear on objects regions that are visible in only one of the two views. Interestingly, these regions are usually located on the object edges. Therefore, looking at cycle inconsistency also provides information about object edges that can help to predict better depth maps. Importantly, this inconsistency can be measured both at training and testing times, even if at testing time, we dispose only of the right image.

The main contribution of this chapter consists in exploiting the cycle inconsistency by training a third network in order to improve the prediction performance and output a refined depth map $\mathbf{d}'_l$. In addition, since employing our inconsistency-aware network leads to more accurate depth predictions, we propose to use the disparity maps predicted by G_i in order to improve G_s training via a knowledge distillation approach.

Note that, another possible cycle approach, as proposed in [75], would consist in using a single network to predict the two disparity maps. The two disparities can be used to obtain the synthesized left image and then the re-synthesized right image. Nevertheless, this approach has a major disadvantage with respect to our approach, *i.e.* , since only the warping operator in employed between the two synthesized images, and consequently the receptive field of $\hat{\mathbf{I}_r}$ in $\mathbf{I}_l$ is very small. In particular, when implementing the warping operator via bilinear sampling, the receptive field of the warping operator in only 2×2. Therefore, the right image reconstruction loss can act on the reconstructed left image only locally. Conversely, our backward network G_b imposes a global consistency on $\mathbf{d}_l$ thanks to its large receptive field.

The outputs of our method correspond to the estimated depth maps $\mathbf{d}_l$ and $\mathbf{d}'_l$. While the estimated depth $\mathbf{d}'_l$ corresponding to the teacher model is typically more accurate, in some applications, *e.g.* in resource-constrained settings, it could be convenient to exploit only a small student network.

In the following, we describe the design of our cycled network. Then, we introduce our novel inconsistency-aware network. Finally, we present the optimization objective including our proposed distillation approach.

6.1.2.2 Unsupervised Monocular Cycled Network

We adopt a setting in which the model is trained without the need of ground truth depth maps. This approach is often referred to as unsupervised or self-supervised depth estimation. Roughly speaking, it consists in training a network to predict a disparity map that can be used to generate the left image from the right image. Formally speaking, we employ a first network G_s that takes as input the right image $\mathbf{I}_r$ and predicts the right-to-left disparity $\mathbf{d}_l$. Following [5], we adopt a U-Net architecture for G_s. We employ a warping function $f_w(\cdot)$ that synthesizes the left view image by sampling from $\mathbf{I}_r$ according to $\mathbf{d}_l$:

$$\hat{\mathbf{I}}_l = f_w(\mathbf{d}_l, \mathbf{I}_r). \tag{6.1}$$

Importantly, $f_w(\cdot)$ is implemented using the bilinear sampler from the spatial transformer network [67] resulting in a fully differentiable model. Consequently, the network can be trained via gradient descent by minimizing the discrepancy between $\hat{\mathbf{I}}_l$ and $\mathbf{I}_l$ (see Sec. 6.1.2.4 for details about network training).

Inspired by [65], we employ a second network G_b in order to re-synthesize the right image according to:

$$\hat{\mathbf{I}}_r = f_w(\mathbf{d}_r, \hat{\mathbf{I}}_l). \tag{6.2}$$

where:

$$\mathbf{d}_r = G_b(\hat{\mathbf{I}}_l) \tag{6.3}$$

The G_b and G_s networks share their encoder parameters. Note that, differently from the stereo depth model proposed in [65], our second half-cycle network takes only the synthesized left image as input. This crucial difference allows the use of this cycle in the monocular setting at testing time. Concerning the decoder networks, we adopt an architecture composed of a sequence of up-convolution layers in which the disparity is estimated and gradually refined

from low to full resolutions similarly to [5]. We obtain the estimated left and the right disparity maps at each scale $\mathbf{d}_l^n$ and $\mathbf{d}_r^n$, $n \in \{0, 1, 2, 3\}$, with sizes $[H/2^n, W/2^n]$. More precisely, $\mathbf{d}_r^n$ is computed from the decoder feature map ${\xi_r}^n$ of size $[H/2^n, W/2^n]$ via a convolutional layer. Then, $\mathbf{d}_r^n$ is concatenated with ${\xi_r}^n$ obtaining a tensor that is input to an up-convolution layer in order to estimate the disparity at the next resolution $\mathbf{d}_r^{n-1}$.

6.1.2.3 Inconsistency-Aware Network

We define the inconsistency tensor as the difference between the input image $\mathbf{I}_r$ and the image $\hat{\mathbf{I}}_r$ predicted by the backward network G_b:

$$\mathbb{I}_r = \mathbf{I}_r - \hat{\mathbf{I}}_r \tag{6.4}$$

The proposed inconsistency-aware network G_i takes as input the concatenation of $\mathbf{I}_r$, $\mathbb{I}_r$ and $\mathbf{d}_l$. We employ a network architecture similar to the half-cycle monocular network described in Sec. 6.1.2.2. However, we propose to provide to the encoder network the disparity maps $\mathbf{d}_l^n, n \in \{1, 2, 3\}$ estimated by G_s at each scale. More precisely, we concatenate along the channel axis each disparity $\mathbf{d}_l^n$ with network features of corresponding dimensionality.

The inconsistency-aware network G_i estimates the right-to-left disparity $\mathbf{d}'_l = G_i(\mathbf{I}_r, \mathbb{I}_r, \mathbf{d}_l, \mathbf{d}_l^{\{1,2,3\}})$ and we reconstruct the left view image $\hat{\mathbf{I}}_l{}'$ by applying the warping function f_w:

$$\hat{\mathbf{I}}'_l = f_w(\mathbf{d}'_l, \mathbf{I}_r) \tag{6.5}$$

Similarly to G_s and G_b, G_i estimates low resolution disparity maps $\mathbf{d}_\mathbf{l}'^n, n \in \{1, 2, 3\}$ that are gradually refined from low to full resolutions.

6.1.2.4 Network Training and Knowledge Self-Distillation

In this section, we detail the losses employed to train the proposed network in an end-to-end fashion.

Reconstruction. First, we employ a reconstruction and stucture similarity loss for each network. Following [5], we adopt the $\mathcal{L}_1$ loss to measure the discrepancy between the synthesized and the real images and the structure similarity loss $\mathcal{L}_{SSIM}$ to measure the discrepancy between the synthesized and the real images structure. By summing the losses of the three networks G_s,

G_b and G_i, we obtain:

$$\begin{aligned}\mathcal{L}_{rec}^{(0)} = \lambda_s[\alpha\mathcal{L}_{SSIM}(\hat{\mathbf{I}}_l, \mathbf{I}_l) + (1-\alpha)||\hat{\mathbf{I}}_l - \mathbf{I}_l||_1] \\ +\lambda_b[\alpha\mathcal{L}_{SSIM}(\hat{\mathbf{I}}_r, \mathbf{I}_r) + (1-\alpha)||\hat{\mathbf{I}}_r - \mathbf{I}_r||_1] \\ +\lambda_t[\alpha\mathcal{L}_{SSIM}(\hat{\mathbf{I}}'_l, \mathbf{I}_l) + (1-\alpha)||\hat{\mathbf{I}}'_l - \mathbf{I}_l||_1]\end{aligned} \tag{6.6}$$

where λ_s, λ_b and λ_t are adjustment parameters and $\alpha = 0.85$. Similarly, we also compute a reconstruction loss $\mathcal{L}_{rec}^{(n)}$ for the low resolution disparity maps. Following [38], we upsample the low resolution $\mathbf{d}_l^n$, $\mathbf{d}_r^n$ and $\mathbf{d}_\mathbf{l}'^n$ to $H \times W$ and use the warping operator f_w to re-synthesize full resolution images that are compared with the real images according to the $\mathcal{L}_1$ loss. The total reconstruction loss is:

$$\mathcal{L}_{rec} = \sum_{n=0}^{4} \mathcal{L}_{rec}^{(n)} \tag{6.7}$$

Self-Distillation. Finally, we propose to introduce a knowledge distillation loss. As detailed in the experimental section (Sec 6.1.3), the inconsistency-aware network outperforms by a significant margin the simple half-cycle network G_s. This boost is at the cost of a higher computation complexity. The idea of the proposed self-distillation loss consists in distilling knowledge from inconsistency-aware network to the half-cycle network G_s. Thus, we improve the performance of G_s without adding any computation complexity at testing time. To do so, we evaluate disparity and feature distillation. For the first, we impose that the network G_d predicts disparity maps similar to the output of inconsistency-aware network. It can be seen as a distillation approach where G_s plays the role of the *student* and the whole network (composed of G_s, G_b and G_i) is the *teacher*. However, in our particular case, the *student* network is a sub-network of the *teacher*. From this perspective, we name this approach self-distillation. The self-distillation loss is given by:

$$\mathcal{L}_{dist} = ||\mathbf{d}_l - \mathcal{S}(\mathbf{d}'_l)||_1 \tag{6.8}$$

where $\mathcal{S}$ denotes the stop-gradient operation. In particular, the stop-gradient operation equals the identity function when computing the forward pass of the back-propagation algorithm but it has a null gradient when computing the backward pass. The purpose of the stop-gradient is to avoid that $\mathbf{d}'_l$ converges to $\mathbf{d}_l$. On the contrary, the goal is to help $\mathbf{d}_l$ to become as accurate as $\mathbf{d}'_l$.

For the second, we impose that the decoder features $\xi_{r'}^n, n \in 0, 1, 2$ of the *teacher* are similar to the features ξ_r^n of the *student*. The self-distillation loss is given by:

$$\mathcal{L}_{dist} = ||\xi_r^n - \mathcal{S}(\xi_{r'}^n)||_2 \tag{6.9}$$

The total training loss is given by:

$$\mathcal{L}_{tot} = \mathcal{L}_{rec} + \lambda_{dist}\mathcal{L}_{dist} \tag{6.10}$$

6.1.3 Experimental Results

We evaluate our proposed approach on two publicly available datasets and compare its performance with state of the art methods.

6.1.3.1 Experimental Setup

Datasets. We perform experiments on two large stereo images datasets, *i.e.* KITTI [17] and Cityscapes [18].

6.1.3.2 Baselines for Ablation.

To perform the ablation study presented in Sec.6.1.3.3, we consider the following baselines:

- *half-cycle*: our basic building block, uses the forward branch that takes $\mathbf{I}_r$ as input and generates $\mathbf{d}_l$ to reconstruct the other stereo view $\hat{\mathbf{I}}_l$. Neither cycle-consistency nor self-distillation are used in this model.

- *cycle*: a backward network is added to the *half-cycle* model in order to reconstruct $\hat{\mathbf{I}}_r$ from the estimated $\hat{\mathbf{I}}_l$. Note that the backward network is used only at training time. At test time, the output is the same as for the *half-cycle* model.

- *teacher*, we stack the inconsistency-aware network after the *cycle* as described in Sec 6.1.2.3.

- *student*: the output of the inconsistency-aware network is distilled in order to refine the first *half-cycle*. At test time, the output and the computation complexity are the same as in the *half-cycle* model.

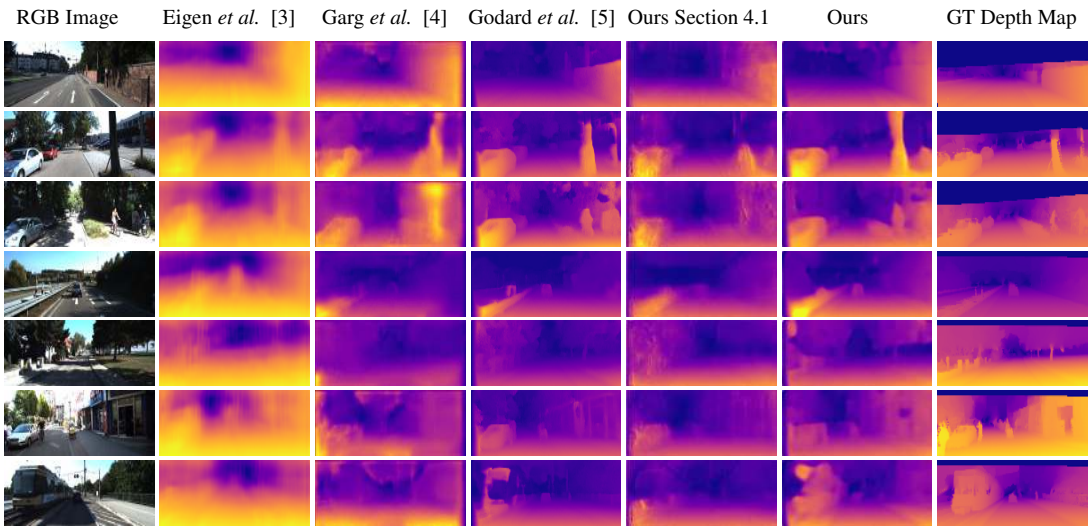


Figure 6.3: Qualitative comparison of different state-of-the-art models with our *teacher* network on the KITTI testing split proposed by [3]. The sparse KITTI ground truth depth maps are interpolated with bilinear interpolation for better visualization.

In Tables 6.1, 6.2 and 6.3 we indicate with *HC*, *C*, *T* and *S*, the *half-cycle*, *cycle*, *teacher* and *student* respectively; *feat* and *disp* denote self-distillations of features and disparities.

Training Procedure. The whole network is trained following an iterative procedure. First, we start by training the forward *half-cycle* network for 10 epochs. In a second step, we train the backward network decoder for 5 epochs without updating the first half-cycle network. The whole cycle is then jointly trained for further 10 epochs. Then, the inconsistency-aware module is pretrained for 5 epochs. Finally, the whole network is jointly fine-tuned for 10 epochs.

Parameters. The model is implemented with the deep learning library *TensorFlow*. Similarly to [5], the input images are down-sampled to a resolution of 512×256 from the original sizes which are 1226×370 for the KITTI dataset and for CityScapes. In all our experiments we use a batch size equal to 8 stereo image pairs and the Adam optimizer with learning rate set to 10^{-5} following the recommendations of [86].

The *half-cycle* and *cycle* networks are trained with the following loss parameters $\lambda_s = 1$, $\lambda_b = 0.1$ and $\lambda_t = 0$. When training the *teacher* network we use $\lambda_s = 0$, $\lambda_b = 0$ and $\lambda_t = 1$. We weight the distillation loss $\mathcal{L}_{dist}$ with $\lambda_{dist} = 0.005$ and $\lambda_{dist} = 0.1$ respectively, if feature distillation or disparity distillation is applied. The joint training of the full network is done with learning rate $l_r = 10^{-5}$, loss parameters $\lambda_s = 1$, $\lambda_b = 0.1$, $\lambda_t = 1$ and λ_{dist} equal to 0.005 in the case feature distillation and 0.1 in the case of disparity distillation, respectively.

Method	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	lower is better				higher is better		
HC	0.1487	1.2942	5.800	0.246	0.805	0.925	0.965
C	0.1451	1.2943	5.850	0.242	0.796	0.924	0.967
T *feat*	*0.1220*	*1.0433*	5.321	0.229	*0.834*	0.933	*0.968*
T *disp*	0.1234	1.0509	*5.283*	*0.228*	*0.834*	*0.934*	*0.968*
S *feat*	0.1438	1.2806	5.834	0.241	0.797	0.926	0.968
S *disp*	0.1438	1.2551	5.771	0.238	0.797	0.927	0.969
[38] $\mathcal{L}_1$ loss							
T *feat*	0.1017	0.8930	4.768	0.206	0.878	0.946	0.972
T *disp*	**0.0983**	**0.8306**	**4.656**	**0.202**	**0.882**	**0.948**	**0.973**
S *feat*	0.1474	1.2416	5.849	0.241	0.788	0.923	0.968
S *disp*	0.1424	1.2306	5.785	0.239	0.795	0.924	0.968

Table 6.1: Ablation study on KITTI dataset using the training and testing split proposed by Eigen *et al.* [3]. The upper part shows the results with the multiscale reconstruction $\mathcal{L}_1$ loss in [5], the bottom part with the $\mathcal{L}_1$ loss proposed in [38].

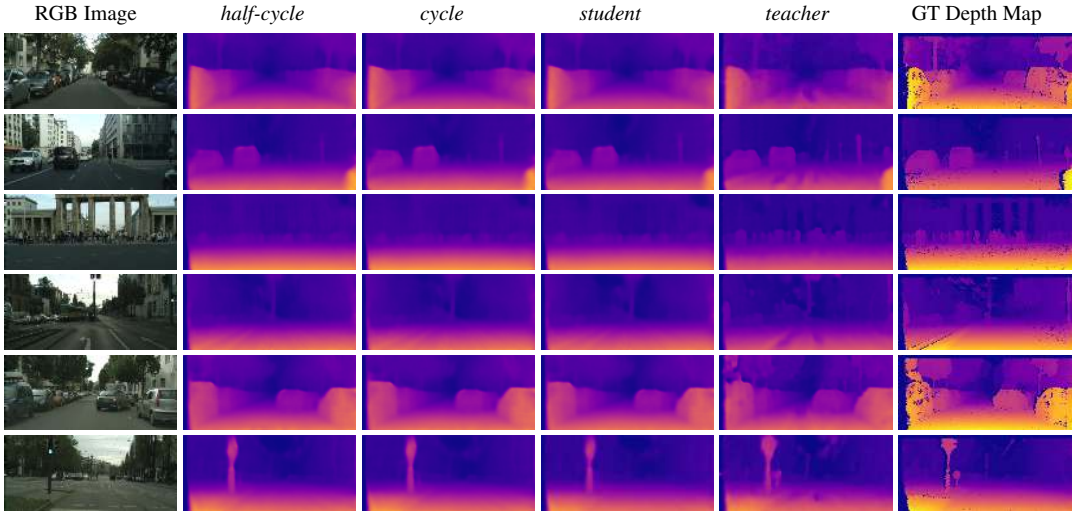


Figure 6.4: Qualitative comparison of different baseline models of the proposed approach on the Cityscapes testing dataset.

Method	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	lower is better				higher is better		
1-CN C	0.1533	1.3326	5.837	0.240	0.785	0.919	0.967
1-CN S *disp*	0.1503	1.2622	5.868	0.243	0.783	0.918	0.967
Ours S *disp*	**0.1438**	**1.2551**	**5.771**	**0.238**	**0.797**	**0.927**	**0.969**
1-CN T *disp*	0.1478	1.3609	5.952	0.243	0.793	0.921	0.966
Ours T *disp*	**0.1234**	**1.0509**	**5.283**	**0.228**	**0.834**	**0.934**	**0.968**

Table 6.2: Ablation study where our two-network *cycle* is replaced by the single-network *cycle* from Yang *et al.* [75] (referred as to 1-CN).

6.1.3.3 Results

Ablation Study. To demonstrate the validity of the proposed contributions we first conduct an ablation study on the KITTI dataset [17] and the CityScapes dataset [18]. Results are shown in Table 6.1 and Table 6.3, respectively.

We split the ablation in two parts where we employ two different reconstruction loss variants. For the first part, as in [5], we use a multi-scale reconstruction loss where the smaller scale reconstruction is compared with a downsampled version of the stereo image. In contrast with that, for the second part, we employ a more effective reconstruction loss, upsampling to input scale all the disparities before warping as described in Sec. 6.1.2.4.

In Table 6.1 it is interesting to note that our intuition of self-constraining the monocular student network with cycled design improves, without requiring additional losses, in several of the metrics compared to the simple forward branch. This comes at the cost of doubling the forward propagation time at training but not at testing time. Moreover, the monocular cycled structure has the big advantage of automatically computing the inconsistency of the reconstruction both at training and testing time. Therefore, stacking a network aware of the inconsistencies and previous estimations, the *teacher* network, improves the performance. We observe that our proposed inconsistency-aware network brings an important improvement consistent over all the metrics, *e.g.* 14% and 18% in *Abs Rel* and *Sq Rel*, respectively, comparing *cycle* and *teacher*.

Student-teacher distillation leads to a consistent improvement over all metrics, demonstrating that self-distillation improves the *student*, while keeping the performance of teacher constant. Regarding the two distillation strategies, we found that network with disparity distil-

Method	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	lower is better				higher is better		
HC	0.4676	7.3992	5.741	0.493	0.735	0.890	0.945
C	0.4523	6.2604	5.381	0.557	0.736	0.888	0.946
T *feat*	0.4087	5.8777	4.394	0.334	0.846	0.940	0.967
T *disp*	*0.3988*	*5.8752*	*4.293*	*0.316*	*0.848*	*0.941*	*0.968*
S *feat*	0.4494	6.2599	5.343	0.421	0.739	0.891	0.947
S *disp*	0.4467	5.9012	5.297	0.473	0.736	0.890	0.946
[38] $\mathcal{L}_1$ loss							
T *feat*	0.3878	**5.8190**	**4.123**	0.397	0.861	**0.945**	**0.969**
T *disp*	**0.3846**	6.2007	4.476	**0.318**	**0.864**	**0.945**	**0.969**
S *feat*	0.4455	6.2748	5.366	0.468	0.739	0.891	0.946
S *disp*	0.4305	5.9552	5.281	0.519	0.740	0.891	0.946

Table 6.3: Ablation study on the Cityscapes dataset. The upper part shows the results with the multiscale reconstruction $\mathcal{L}_1$ loss in [5], the bottom part with the $\mathcal{L}_1$ loss proposed in [38].

lation converges faster than that with the feature distillation. This is not unexpected, given the much more compact size of the disparity compared to the several channels of the features.

For demonstrating the validity of the design of our cycle network, we perform an ablation study where our two-network *cycle* structure is replaced by the single-network *cycle* proposed by Yang *et al.* [75]. In this experiment, we use our proposed inconsistency-aware module to exploit the inconsistency estimated by the single network cycle in [75]. Contrary to [75], we trained the models without supervision in order to compare the two different approaches in the unsupervised setting. We use the $\mathcal{L}_1$ loss from [5] for fair comparison. Results are reported in Table 6.2. We observe that the inconsistency estimates obtained with the single-network cycle of [75] are associated with worse performance with respect to those of our method.

We also performed an ablation study on the Cityscapes dataset in Table 6.3, following the evaluation procedure proposed in 4.1. The results confirm the trends observed on KITTI. The *cycle* network improves over the *half-cycle* in five metrics out of seven. The *teacher*, effectively exploiting inconsistencies, is associated with an improvement on all error metrics (ranging from 7% to 20%). Distillation further provides a boost in performance of about 1.5% to 5%. In the second part of the ablation study, the *teacher* further improves its estimations gaining over 20% over the initial *cycle* setting. More interesting is the gain in performance of the *student* that improves from 2% to 5%.

Method	Sup	Video	Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
			lower is better				higher is better		
Eigen *et al.* [3]	Y	N	0.190	1.515	7.156	0.270	0.692	0.899	0.967
Xu *et al.* [10]	Y	N	0.132	0.911	-	*0.162*	0.804	0.945	0.981
Jiang *et al.* [71]	Y	N	0.131	0.937	5.032	0.203	0.827	0.946	0.981
Gan *et al.* [72]	Y	N	0.098	0.666	*3.933*	0.173	*0.890*	0.964	0.985
Guo *et al.* [73]	Y	N	*0.097*	*0.653*	4.170	0.170	0.889	*0.967*	*0.986*
Yang *et al.* [75]	Y	Y	*0.097*	*0.734*	*4.442*	*0.187*	*0.888*	*0.958*	*0.980*
Zou *et al.* [74]	N	Y	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Godard *et al.* [38]	N	Y	0.115	1.010	5.164	0.212	0.858	0.946	0.97
Zhou *et al.* [35]	N	N	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Garg *et al.* [4]	N	N	0.169	1.08	5.104	0.273	0.740	0.904	0.962
Kundu *et al.* [40], 50m	N	N	0.203	1.734	6.251	0.284	0.687	0.899	0.958
Godard *et al.* [5]	N	N	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Ours Section 4.1	N	N	0.152	1.388	6.016	0.247	0.789	0.918	0.965
Ours Student	N	N	0.1424	1.2306	5.785	0.239	0.795	0.924	0.968
Ours Teacher	N	N	**0.0983**	**0.8306**	**4.656**	**0.202**	**0.882**	**0.948**	**0.973**

Table 6.4: Comparison with the state of the art. Training and testing are performed on the KITTI [17] dataset. Supervised and semi-supervised methods are marked with Y in the supervision (Sup.) column, unsupervised methods with N. Methods using a frame sequence in input and, thus, exploiting temporal information either at training or testing time, are marked with Y in the *Video* column. Numbers are obtained on Eigen [3] test split with Garg [4] image cropping. Depth predictions are capped at the common threshold of 80 meters, if capped at 50 meters we specify it. Best scores among static unsupervised methods are in bold. Best scores among other method categories are in italic.

In Fig. 6.4, we present qualitative results for Cityscapes. *half-cycle* and *cycle* images are smooth and do not present artifacts. The *teacher* provides more accurate depth maps with sharper edges for small objects and better background estimations (*e.g.* third row, people in the back). After distillation also the *student* inherits this ability and we observe more detailed predictions compared to the original *cycle*.

6.1.3.4 Comparison with State-of-the-Art

In Table 6.4 we compare with several state-of-the-art works, considering both supervised learning-based (Eigen *et al.* [3], Xu *et al.* [10], Jiang *et al.* [71], Gan *et al.* [72], Guo *et al.* [73], Yang *et al.* [75]) and unsupervised learning-based (Zhou *et al.* [35], Garg *et al.* [4], Kundu *et al.* [40], Godard *et al.* [5], Ours Section 4.1, Godard *et al.* [38] and Zou *et al.* [74])

methods.

The *teacher* network reaches state-of-the-art performance for the frame-level unsupervised setting, even improving over the state-of-the-art method that use depth supervision as [10], and is competitive with those using depth and video clues [72, 73, 75]. Note that Yang *et al.* [75] consider a similar setting to ours proposing to use errors to refine the depth estimation with a stacked network. Our method has several advantages though: it is unsupervised, it does not consider multiple video frames and it avoids the use of several losses whose hyper-parameters are hard to tune. Furthermore, as demonstrated by our experiments in Table 6.2, our approach adopts a more effective network structure for computing cycle inconsistencies. The *student* network, after distillation, improves on unsupervised approaches with similar network capacity like [4, 5] and it is only outperformed by previous unsupervised methods that exploit additional information during training like [38].

Qualitative results in Figure 6.3 show that our model predicts more accurately challenging areas, *i.e.* sky, trees in background and shadowed areas difficult to interpret, compared to competitive unsupervised models [4, 5]. Note that small details are better reconstructed by [5] but, overall, our estimations look smoother and have fewer large errors, as the train windshield in row seven.

6.1.4 Conclusions

We proposed a monocular depth estimation network which computes the inconsistencies between input and cycle-reconstructed images and exploit them to generate state-of-the-art depth predictions through a refinement network. We proved that distillation is an effective paradigm for depth estimation and improve the student network performance by transferring information from the refinement network.

7

Final Remarks

In this thesis, we studied unsupervised depth estimation from binocular stereo images. As already explained in the Introduction, depth estimation is a task of growing importance in computer vision, given the multiple application that could benefit from it. We mentioned robotics and autonomous driving previously, but substantially depth estimation could be applied to any system as an additional form of sensing the real world, for example, virtual reality and 3D reconstruction. Another example, and very active research topic, is depth completion which, in other words, infers a dense prediction from a very sparse LiDaR ground truth.

Unsupervised binocular stereo depth estimation, is based on a image reconstruction optimization objective. Obtaining higher quality of reconstruction, corresponds to better accuracy in depth predictions. In this framework, our proposal to use adversarial learning proved to be beneficial for model training. Furthermore, we made the best use of the reconstructed images during training for data augmentation. Our model was trained not only on the real images but also on the resynthesized ones. Tackling the stereo scenario allowed us to have both stereo images feature representations. Thus, we propose PFN (Progressive Fusion Network) to make the best use of the two stereo views complementary information.

Next, we focused on improving adversarial depth estimation. For this reason, we couple the loss of discriminator and generator, with a structured approach as CRFs, to guide the optimization process where it is more needed. In this way we exploit more efficiently the wrong reconstructions highlighted by the reconstruction loss as well as the critic opinion of the discriminator. A secondary contribution, is the *hallucination subnetwork* that allows inference with monocular images.

Finally, we revisited data augmentation cycle for monocular depth estimation. This scenario is more challenging but allows for inference with monocular images. The peculiarity of our approach is the ability to estimate the reconstruction error at the end of the cycle from the image without any supervision. We were then able to exploit this very precious information as best as possible. First, with a refinement block that, having access to images, predictions and errors of the cycle block, could achieve much higher performance. Second, we applied the knowledge distillation principle. Considering the cycle block as a student model, and the refinement block as a teacher model, we could improve the student accuracy in a self-improvement fashion.

7.1 Future Research Directions

This final section includes a short overview of the main challenges for depth estimation and then depicts future research directions that could come out from this thesis.

Computer vision is very fortunate to see growing interest from the academic and industrial worlds. However, deployment of deep learning models for real applications is still not straight forward, and future studies will require a great effort. In the last couple of years, for depth estimation, researchers started to tackle this problem by developing more efficient models. Moreover, domain adaptation methods have been applied to depth estimation model in order to obtain better performing model on different data. Another exciting trend, strongly encouraged by recent monocular dataset, is research towards fully monocular methods that exploit temporal consistency for training. All these research treds, in the long run, will allow the practical use of deep learning models.

Finally, we detail a few research ideas for future works. For the sake of clarity, we list them by chapter.

- Chapter 4: The proposed adversarial learning and data augmentation have been evaluated for depth estimation. Other dense regression tasks could potentially benefit from these findings. One example is optical flow, where a model can be trained by image reconstruction between consecutive frames similarly to self-supervised depth estimation. There are more challenges due to the increased pixel occlusions in consecutive time frames, but with the proper adjustments, it is a feasible application scenario. Another area of interest, is to transform the discriminator decision from a global image perspective to a pixel-wise decision. In other words, our discriminator outputs an global score for the

image. A natural extension would be to have a fine-grained decision for each pixel of the image. By doing this, we could more pointedly address reconstruction errors at the generator level.

- Chapter 5: While our experiments show that a structured coupling as CRF is very effective for depth estimation, we also noticed that our model is very time and resource consuming for training. This finding motivates us to research other, more efficient solutions in the future: from more efficient network architecture than the used ResNet50, to reformulating the CRF coupling for a more streamlined approach.

- Chapter 6: The monocular setting is the most interesting for our future works. Indeed, the research community is going to pay growing attention to monocular applications as they require fewer sensors to be deployed. A binocular stereo setting has to be calibrated and synchronized, instead, a single camera is easier to deploy. We plan to further improve the distillation process by accounting for teacher and student confidence in the estimates. In this way, we expect to guide the learning process better and correct more effectively prediction inconsistencies. A second possibility is to detach student and teacher network. In our work, they have a single flow of the gradient in backpropagation, therefore, to obtain a more straightforward optimization procedure it could be interesting to redraw the network as a classic student and teacher networks for knowledge distillation. Third, we would like to improve the model training to a fully monocular setting. Conversely to the rectified stereo image pairs used in these works, can be achieved by using temporal reconstruction self-supervision in a video stream.

Bibliography

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012. 2, 12

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009. 2, 12

[3] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014. 4, 10, 13, 14, 16, 17, 23, 25, 27, 29, 50, 51, 52, 53, 63, 64, 65, 71, 79, 80, 83

[4] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *ECCV*. Springer, 2016. 4, 11, 17, 23, 25, 27, 31, 40, 50, 51, 52, 56, 64, 65, 67, 72, 79, 83, 84

[5] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017. 4, 6, 11, 13, 14, 17, 22, 23, 25, 27, 30, 31, 34, 38, 39, 40, 42, 43, 46, 50, 51, 52, 53, 56, 58, 64, 65, 67, 71, 72, 75, 76, 79, 80, 81, 82, 83, 84

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014. 4, 12, 20, 34, 35, 56

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017. 4, 5, 11, 12, 31

[8] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," *arXiv preprint*, 2017. 4, 5, 12

[9] Y. Zhong, Y. Dai, and H. Li, "Self-supervised learning for stereo matching with self-improving ability," *arXiv preprint arXiv:1709.00930*, 2017. 5, 11

[10] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Monocular depth estimation using multi-scale continuous crfs as sequential deep networks," *TPAMI*, 2018. 5, 10, 16, 17, 25, 27, 29, 50, 52, 71, 83, 84

[11] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *CVPR*, 2015. 5

[12] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015. 6

[13] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*, 2015. 6

[14] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *CVPR*, 2017. 6

[15] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012. 9

[16] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *TPAMI*, vol. 31, no. 5, pp. 824–840, 2009. 9, 63, 65

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012. 9, 23, 25, 41, 52, 78, 81, 83

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016. 9, 23, 32, 41, 63, 72, 78, 81

[19] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *CVPR*, 2018. 9, 32, 41, 43

[20] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019. 10

[21] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *TPAMI*, 2016. 10, 16, 17, 25, 27, 29, 50, 52, 60, 65, 71

[22] W. Zhuo, M. Salzmann, X. He, and M. Liu, "Indoor scene structure analysis for single image depth estimation," in *CVPR*, 2015. 10

[23] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3DV*, 2016. 10, 39, 40

[24] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *CVPR*, 2018. 10

[25] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille, "Towards unified depth and semantic prediction from a single image," in *CVPR*, 2015. 10, 13

[26] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *CVPR*, 2018. 10, 65

[27] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *CVPR*, 2018. 10

[28] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *ICCV*, 2017. 10

[29] S. Tulyakov, A. Ivanov, and F. Fleuret, "Practical deep stereo (pds): Toward applications-friendly deep stereo matching," *NIPS*, 2018. 10

[30] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *CVPR*, 2019. 10

[31] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. Di Stefano, "Real-time self-adaptive deep stereo," in *CVPR*, 2019. 10, 30, 50, 52

[32] A. Atapour-Abarghouei and T. P. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," in *CVPR*, 2018. 10, 11

[33] Y. Kuznietsov, J. Stückler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," *CVPR*, 2017. 11, 25, 27, 50, 52, 65

[34] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," *arXiv preprint arXiv:1803.03893*, 2018. 11, 58, 65, 67

[35] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017. 11, 17, 23, 25, 27, 31, 50, 51, 52, 64, 65, 71, 83

[36] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *CVPR*, 2018. 11, 17, 31, 65, 67, 71

[37] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *CVPR*, 2018. 11

[38] C. Godard, O. M. Aodha, and G. Brostow, "Digging into self-supervised monocular depth prediction," 2019. 11, 52, 53, 77, 80, 82, 83, 84

[39] H.-Y. Lai, Y.-H. Tsai, and W.-C. Chiu, "Bridging stereo matching and optical flow via spatiotemporal correspondence," in *CVPR*, 2019. 11

[40] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu, "Adadepth: Unsupervised content congruent adaptation for depth estimation," in *CVPR*, 2018. 11, 25, 27, 50, 52, 53, 56, 65, 83

[41] C. Zheng, T.-J. Cham, and J. Cai, "T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," in *ECCV*, 2018. 11

[42] S. Zhao, H. Fu, M. Gong, and D. Tao, "Geometry-aware symmetric domain adaptation for monocular depth estimation," in *CVPR*, 2019. 11

[43] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016. 12

[44] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *NIPS*, 2016. 12

[45] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arxiv*, 2016. 12

[46] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*, 2017. 12

[47] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos, "The cramer distance as a solution to biased wasserstein gradients," 2018. [Online]. Available: ICLR 12

[48] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2813–2821. 12, 35, 40

[49] A. Siarohin, S. Lathuilière, E. Sangineto, and N. Sebe, "Appearance and pose-conditioned human image generation using deformable gans," *T-PAMI*, 2019. 12

[50] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," in *NIPS*, 2019. 12

[51] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *NIPS Workshop on Deep Learning*, 2015. 12

[52] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *CVPR*, 2017. 12

[53] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *ICLR*, 2017. 12

[54] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *CVPR*, 2016. 12

[55] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation." in *ICCV*, 2017. 12

[56] P. Luo, Z. Zhu, Z. Liu, X. Wang, X. Tang *et al.*, "Face model compression by distilling knowledge from neurons." in *AAAI*, 2016. 12

[57] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection cnns by self attention distillation," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 12

[58] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *CVPR*, 2014. 16, 29, 71

[59] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *NIPS*, 2006. 17, 25, 27, 29, 50, 52, 63

[60] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *CVPR*, 2016. 17, 18, 30, 32, 72

[61] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *CVPR*, 2016. 17, 30, 50, 52, 67, 72

[62] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *CVPR*, 2018. 17, 30, 31, 65, 72

[63] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015. 22, 40

[64] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/ 23, 42

[65] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe, "Unsupervised adversarial depth estimation using cycled generative networks," in *3DV*, 2018. 30, 41, 42, 56, 71, 75

[66] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *IJRR*, 2013. 32, 63, 72

[67] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *NIPS*, 2015. 33, 75

[68] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *CVPR*, 2018. 37

[69] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015. 40

[70] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. 46

[71] H. Jiang, G. Larsson, M. Maire Greg Shakhnarovich, and E. Learned-Miller, "Self-supervised relative depth learning for urban scene understanding," in *ECCV*, 2018. 50, 52, 83

[72] Y. Gan, X. Xu, W. Sun, and L. Lin, "Monocular depth estimation with affinity, vertical pooling, and label enhancement," in *ECCV*, September 2018. 50, 52, 65, 67, 83, 84

[73] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, "Learning monocular depth by distilling cross-domain stereo networks," in *ECCV*, 2018. 50, 52, 65, 67, 83, 84

[74] Y. Zou, Z. Luo, and J.-B. Huang, "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency," 2018. 52, 65, 67, 83

[75] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *The European Conference on Computer Vision (ECCV)*, September 2018. 50, 74, 81, 82, 83, 84

[76] A. Behl, O. H. Jafari, S. K. Mustikovela, H. A. Alhaija, C. Rother, and A. Geiger, "Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios?" in *CVPR*, 2017. 56

[77] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, and M. Sitti, "A non-rigid map fusion-based rgb-depth slam method for endoscopic capsule robots," *arXiv preprint arXiv:1705.05444*, 2017. 56

[78] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation," in *CVPR*, 2017. 60

[79] V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *NIPS*, 2011. 60

[80] K. Ristovski, V. Radosavljevic, S. Vucetic, and Z. Obradovic, "Continuous conditional random fields for efficient regression in large fully connected graphs." in *AAAI*, 2013. 60

[81] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015. 64

[82] A. Adams, J. Baek, and M. A. Davis, "Fast high-dimensional filtering using the permutohedral lattice," in *Computer Graphics Forum*, vol. 29, no. 2, 2010, pp. 753–762. 68

[83] R. Girshick, "Fast r-cnn," in *ICCV*, 2015. 71

[84] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Semi-convolutional operators for instance segmentation," in *ECCV*, 2018. 71

[85] X. Nie, J. Feng, J. Xing, and S. Yan, "Pose partition networks for multi-person pose estimation," in *ECCV*, 2018. 71

[86] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "A comprehensive analysis of deep regression," *TPAMI*, 2019. 79