Universidade Estadual de Campinas
Instituto de Computação

Julio César Mendoza Bobadilla

Improving Loss Functions and Feature Utilization for Self-Supervised Single-Image Depth Estimation from Monocular Videos

Melhorando Funções de Perda e Utilização de Características para Estimação Autossupervisionada de Profundidade de Imagem Única a Partir de Vídeos Monoculares

CAMPINAS
2022

Julio César Mendoza Bobadilla

Improving Loss Functions and Feature Utilization for Self-Supervised Single-Image Depth Estimation from Monocular Videos

Melhorando Funções de Perda e Utilização de Características para Estimação Autossupervisionada de Profundidade de Imagem Única a Partir de Vídeos Monoculares

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Hélio Pedrini

Este exemplar corresponde à versão final da Tese defendida por Julio César Mendoza Bobadilla e orientada pelo Prof. Dr. Hélio Pedrini.

CAMPINAS
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

M523i

Mendoza Bobadilla, Julio César, 1990-
Improving loss functions and feature utilization for self-supervised single-image depth estimation from monocular videos / Julio César Mendoza Bobadilla. – Campinas, SP : [s.n.], 2022.

Orientador: Hélio Pedrini.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Visão por computador. 2. Redes neurais (Computação). 3. Inteligência artificial. I. Pedrini, Hélio, 1963-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Melhorando funções de perda e utilização de características para estimação autossupervisionada de profundidade de imagem única a partir de vídeos monoculares
Palavras-chave em inglês:
Computer vision
Neural networks (Computer science)
Artificial intelligence
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Hélio Pedrini [Orientador]
Ronaldo Cristiano Prati
Guillermo Cámara Chávez
Romis Ribeiro de Faissol Attux
Allan da Silva Pinto
Data de defesa: 13-05-2022
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-5820-2615
- Currículo Lattes do autor: http://lattes.cnpq.br/4651224173763517

Universidade Estadual de Campinas
Instituto de Computação

Julio César Mendoza Bobadilla

Improving Loss Functions and Feature Utilization for Self-Supervised Single-Image Depth Estimation from Monocular Videos

Melhorando Funções de Perda e Utilização de Características para Estimação Autossupervisionada de Profundidade de Imagem Única a Partir de Vídeos Monoculares

Banca Examinadora:

- Prof. Dr. Hélio Pedrini
  IC/UNICAMP
- Prof. Dr. Guillermo Cámara Chávez
  DECOM/UFOP
- Prof. Dr. Ronaldo Cristiano Prati
  CMCC/UFABC
- Allan da Silva Pinto
  LNLS
- Romis Ribeiro de Faissol Attux
  FEEC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 13 de maio de 2022

Acknowledgments

First, I dedicate this accomplishment to my parents Benigna and Abel for their support and love throughout my life. I would like to thank all my family for their words of encouragement and support.

I am very grateful to my advisor, Professor Hélio Pedrini, for the all his advice, words of encouragement, continuous support, guidance, wisdom, patience, and insightful conversations during my Ph.D. studies. I would like to thank to all members of my committee, specially to Ph.D. Allan da Silva Pinto and Professor Romis Ribeiro de Faissol Attux, who participated in my qualification exam, also to Professor Guillermo Cámara Chávez and Professor Ronaldo Cristiano Prati. Their valuable feedback contributed greatly to this thesis.

I would like to thank to all my colleagues at the Laboratory of Visual Information (LIV), particularly to my nearby labmates Darwin, Jhosimar, Juan, John, Rodolfo, Jadisha, and Anderson for the valuable and nice conversations shared during our workdays at the lab, and for sharing their motivation for doing academic research. I am very grateful to Carla for the continuous encouragement and motivation, and for being supportive during these years.

I am grateful to the faculty and staff of the Institute of Computing at University of Campinas for providing a great, open, and kind environment for learning and doing research. I am thankful to Professor João Meidanis for providing access to computational resources that contributed greatly to the development of this thesis.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Finance Code 001.

This study was financed in part by The Brazilian National Council for Scientific and Technological Development (CNPq), grant #309330/2018-1.

Finally, I would like to acknowledge the São Paulo Research Foundation (FAPESP), project #2018/00031-7 for providing access to computational resources.

Resumo

Nesta tese, abordamos o problema de estimação de um mapa de profundidade denso a partir de uma única imagem de entrada. Focamos em abordagens autossupervisionadas que usam a reconstrução de vistas como uma tarefa auxiliar e usam vídeos monoculares para treinamento. Como a reconstrução das vistas depende de encontrar correspondências de pixels precisas entre as vistas em uma cena, um desafio importante é evitar que estimações de correspondências incorretas reduzam a eficácia da reconstrução de vistas baseada em perda para convergir em uma solução que tenha um desempenho adequado na estimação de profundidade. Estimações incorretas de correspondência de pixels podem ocorrer devido a vários motivos. Por exemplo, alguns pixels não têm correspondências de pixel verdadeiras, como pixels localizados em regiões com oclusão/desoclusão devido ao movimento da câmera ou do objeto. Outros pixels parecem ter várias correspondências, como pixels localizados em regiões homogêneas ou de pouca textura. Além disso, alguns pixels têm correspondência verdadeira em visualizações adjacentes com representações de características inconsistentes devido à reflexão e à refração que dificultam a correspondência. Para contornar esse desafio, desenvolvemos vários mecanismos para reduzir a influência de pixels com estimações de correspondência incorretas. Primeiramente, propusemos uma heurística baseada na consistência de profundidade para diminuir a influência dos pixels na função de perda. Além disso, desenvolvemos um mecanismo de atenuação de perda adaptativa para reduzir a influência de pixels com estimações de correspondências incorretas com base na incerteza aleatória. Por fim, formulamos uma função de perda de consistência adaptativa que penaliza a diferença de várias representações de características considerando apenas as correspondências com erro mínimo de reprojeção. Nossos resultados demonstram que as melhorias propostas para a função de perda podem aumentar a precisão do nosso modelo autossupervisionado de estimação de profundidade de imagem única. Outro desafio está relacionado à observação de que otimizar um modelo com reconstrução de vistas como tarefa auxiliar não implica que o modelo seja otimizado para a estimação de profundidade. Em resposta a esse desafio, desenvolvemos mecanismos para alavancar as representações de características aprendidas pelo modelo. Inicialmente, propusemos um mecanismo de compartilhamento de características que permite que o modelo de movimento da câmera aproveite as características profundas aprendidas pelo modelo por meio de conexões laterais. Além disso, a função de perda de consistência adaptativa leva em conta o mapa de coordenadas 3D, as características profundas e as representações de cores com reprojeção mínima. Por fim, desenvolvemos um método para realizar a autodestilação para fornecer um sinal de aprendizado adicional para treinamento. Esse método é o resultado da adaptação e avaliação de estratégias de aplicação de consistência para realizar a autodestilação por meio da consistência de predição. Nossos resultados mostram que as melhorias na forma como aproveitamos as representações de características e a autodestilação podem aumentar o desempenho da estimação autossupervisionada de profundidade de uma única imagem.

Abstract

In this thesis, we address the problem of estimating a dense depth map from a single input image. We focus on self-supervised approaches that use view reconstruction as an auxiliary task and use monocular videos for training. Since view reconstruction depends on finding accurate pixel correspondences among views of a scene, an important challenge is to prevent incorrect correspondence estimates from reducing the effectiveness of the view reconstruction-based loss to converge on a solution that performs well in depth estimation. Incorrect pixel correspondence estimates can occur due to a variety of reasons. For example, some pixels have no true pixel correspondences, such as pixels located in regions with occlusion/disocclusion due to camera or object motion. Other pixels appear to have multiple correspondences, such as the pixels located in homogeneous or low-textured regions. Moreover, some pixels have true corresponding ones in adjacent views with inconsistent feature representations due to reflection and refraction that make matching difficult. To address this challenge, we develop several mechanisms to diminish the influence of pixels with incorrect correspondence estimates. First, we propose a heuristic based on depth consistency to reduce the influence of pixels on the loss function. In addition, we formulate an adaptive loss attenuation mechanism to decrease the influence of pixels with incorrect correspondence estimates based on aleatoric uncertainty. Finally, we develop an adaptive consistency loss function that penalizes the difference of several feature representations considering only the correspondences with the minimum re-projection error. Our results demonstrate that the proposed improvements to the loss function can increase the accuracy of our self-supervised single image depth estimation model. Another challenge is related to the observation that optimizing a model with view reconstruction as auxiliary task does not imply that the model is optimized for depth estimation. In response to this challenge, we proposed mechanisms to leverage the feature representations learned by the model. First, we propose a feature sharing mechanism that allows the camera motion model to take advantage of the deep features learned by the depth model via lateral connections. In addition, the adaptive consistency loss leverages 3D coordinate map, deep features, and color representations on minimum re-projection. Finally, we develop a method to perform self-distillation to provide an additional learning signal for training. This method is the result of adapting and evaluating consistency enforcement strategies to perform self-distillation through prediction consistency. Our results show that improvements in how we leverage feature representations and self-distillation can increase performance in self-supervised single image depth estimation.

List of Figures

1.1 Input RGB image and its dense depth map prediction 14

2.1 Pixel coordinate mapping between a pair of frames. 22
2.2 Coarse-to-fine strategy using a convolutional encoder-decoder network. . . 24

3.1 Examples from the KITTI dataset. 28

4.1 Overview of our method 32
4.2 Thresholded and soft visibility masks 34
4.3 Feature sharing mechanism 36
4.4 Input images and corresponding depth maps generated with our method . 39

5.1 Overview of our method 42
5.2 Qualitative results of error weighting approach with uncertainty 48
5.3 Qualitative results 49

6.1 Overview of our method 54
6.2 Simplified views of the consistency enforcement strategies 56
6.3 Qualitative results 60

List of Tables

4.1 Ablation analysis 37
4.2 Results of depth estimation on the Eigen split of the KITTI dataset 38

5.1 Ablation study on the adaptive consistency loss 46
5.2 Using uncertainty to weigh the error contribution by pixel 47
5.3 Using uncertainty to weigh the error contribution by image 47
5.4 Ablation study of additional masks. We considered the Field-of-View masks (FOV), Auto mask (AM), Geometric mask(GM). 48
5.5 Results of depth estimation on the Eigen split of the KITTI dataset 49

6.1 Comparison of the baseline model and the variation of our method that uses the pseudo-labeling strategy. 59
6.2 Comparison of variants of our method with and without filtering strategies 59
6.3 Comparison of the representative consistency enforcement strategies 60
6.4 Comparison with the state-of-the-art on the Eigen split of the KITTI dataset 61

List of Abbreviations and Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
ADAM	Adaptive Moment Estimation
AR	Augmented Reality
CAD	Computer Aided Design
CNN	Convolutional Neural Network
DC	Depth Consistency
DNN	Deep Neural Network
DoF	Degrees-of-Freedom
DSSIM	Structure Dissimilarity
ELU	Exponential Linear Unit
EMA	Exponential Moving Average
FOV	Field of View
FS	Feature Sharing
GPU	Graphics Processing Unit
ICP	Iterative Closest Point
KD	Knowledge Distillation
LIDAR	Light Detection and Ranging
MRF	Markov Random Field
MSE	Mean Squared Error
PDF	Probability Distribution Function
RGB	Red-Green-Blue
RMSE	Root Mean Squared Error
SD	Self-Distillation
SIDE	Single-Image Depth Estimation
SSIM	Structure Similarity
SV	Soft Visibility
SWA	Stochastic Weighted Average
TV	Thresholded Visibility

Contents

1 Introduction **13**
- 1.1 Problem and Approach . . . 14
- 1.2 Challenges . . . 15
  - 1.2.1 Incorrect Pixel Correspondence Estimates . . . 15
  - 1.2.2 Divergence of View Reconstruction and Depth Estimation Performance . . . 16
- 1.3 Objectives . . . 16
- 1.4 Research Questions . . . 17
- 1.5 Contributions . . . 18
- 1.6 Publications . . . 18
- 1.7 Thesis Organization . . . 19

2 Background **20**
- 2.1 Learning SIDE from Monocular Videos . . . 20
  - 2.1.1 View Reconstruction Loss . . . 22
  - 2.1.2 Smoothness Regularization . . . 23
  - 2.1.3 Multi-Scale Architecture . . . 23
  - 2.1.4 Occlusion Handling . . . 23
  - 2.1.5 Multi-View Consistency . . . 24
- 2.2 Knowledge Distillation . . . 25
  - 2.2.1 Teacher-Student Structure . . . 25
  - 2.2.2 Applications . . . 26
  - 2.2.3 Self-Distillation . . . 26

3 Materials and Metrics **27**
- 3.1 Dataset . . . 27
- 3.2 Evaluation Metrics . . . 27
- 3.3 Hardware and Software Resources . . . 28

4 Self-Supervised Depth Estimation Based on Feature Sharing and Consistency Constraints **30**
- 4.1 Related Work . . . 30
  - 4.1.1 Geometric Constraints and Occlusion . . . 30
  - 4.1.2 Multi-Task Architecture . . . 31
- 4.2 Proposed Method . . . 31
  - 4.2.1 Overview . . . 32
  - 4.2.2 Depth Consistency and Occlusion . . . 33
  - 4.2.3 Depth Encoder Feature Sharing . . . 35
  - 4.2.4 Network Architecture . . . 35

4.3 Experiments . . . 36
4.3.1 Experimental Setup . . . 36
4.3.2 Depth Estimation . . . 37
4.4 Final Considerations . . . 39

5 Adaptive Self-Supervised Monocular Depth Estimation 40
5.1 Related Work . . . 40
5.1.1 Consistency Constraints . . . 40
5.1.2 Adaptive Losses based on Uncertainty . . . 41
5.2 Method . . . 41
5.2.1 Preliminaries . . . 41
5.2.2 Adaptive Consistency Loss . . . 42
5.2.3 Error Weighting Using Uncertainty . . . 43
5.2.4 Exploring Visibility Masks . . . 45
5.2.5 Implementation Details . . . 45
5.3 Experiments . . . 45
5.3.1 Experimental Setup . . . 45
5.3.2 Adaptive Consistency Loss . . . 46
5.3.3 Error Weighting Using Uncertainty . . . 46
5.3.4 Visibility Masks . . . 47
5.3.5 Comparison with the State of the Art . . . 47
5.4 Final Considerations . . . 49

6 Self-Distilled Self-Supervised Monocular Depth Estimation 51
6.1 Related Work . . . 51
6.1.1 Pseudo-Labeling Approaches for Self-Supervised Depth Estimation 52
6.1.2 Self-Distillation . . . 52
6.1.3 Consistency Regularization . . . 53
6.2 Proposed Method . . . 53
6.2.1 Self-Distillation via Prediction Consistency . . . 53
6.2.2 Filtering Pseudo-Labels . . . 54
6.2.3 Consistency Enforcement Strategies . . . 55
6.2.4 Additional Considerations . . . 56
6.3 Evaluation . . . 57
6.3.1 Experimental Setup . . . 58
6.3.2 Self-Distillation via Prediction Consistency . . . 58
6.3.3 Filtering Pseudo-Labels . . . 59
6.3.4 Consistency Enforcement Strategies . . . 59
6.3.5 State-of-the-Art Comparison . . . 59
6.4 Final Considerations . . . 60

7 Conclusions and Future Work 62

Bibliography 72

Chapter 1

Introduction

Humans have a remarkable capability to perceive the structure of the world. Given a single image, we can infer the several properties of the objects in the scene, such as their 3D shape, if they are rigid or deformable, if they are moving or static. Moreover, we can also infer relationships between the objects in the image, for example, if they are closer or distant among themselves or to the camera. Studies [36, 72, 79] suggest that we develop a structural understanding very early through visual experiences that consist mainly of observing and moving around our environment with very weak or no supervision.

Approaches to incorporate visual perception into computer systems have been widely studied in the literature. In recent years, deep learning methods have been successfully applied in many computer vision tasks, including tasks related to perception of 3D structure such as depth estimation. One of the key elements that enabled deep learning success is the availability of large labeled data sets. However, collecting large labeled data is expensive because it requires a huge amount of human effort. Moreover, the intuition that increasing the size of labeled data sets used for training is enough to enhance a model significantly does not always hold. Even in very large weakly labeled data sets of hundred of millions of examples performance increases at a log-linear rate than the size of data sets [24]. Thus, the size, quality and availability of labeled data sets are becoming a bottleneck for supervised methods.

In this thesis, we focus on single-image depth estimation (SIDE). The context of this problem is more adverse than other fundamental computer vision problems. Depth ground truth is mostly obtained with complex procedures that involve interpolating 3D Light Detection and Ranging (LIDAR) point clouds, using 3D fitted CAD models, data cleaning, and/or post-processing methods [41]. Moreover, data sets in this area [35, 74, 42, 53] are by far smaller when compared to the data sets available in other computer vision problems [6, 1, 65].

In self-supervised deep learning, the methods are trained on auxiliary or pretext tasks for which labels are obtained from the data itself, for free. Moreover, self-supervised methods are improving every year and becoming competitive with supervised learning methods. Several works aim to close the gap between supervised and self-supervised learning in various scenarios. For instance, in visual representation learning, Goyal et al. [24] showed that self-supervised pre-training can outperform or be competitive with supervised pre-training in several tasks. Moreover, when self-supervised methods cannot

surpass their supervised counter-pats, they can be complementary and improve aspects of model robustness such as robustness to adversarial examples and label corruption [27].

SIDE can be learned with a self-supervised approach. This approach consist of learning depth using view reconstruction as a pretext task. SIDE methods based on this approach rely on appearance and geometric consistency among nearby frames on videos, to reconstruct a reference frame with the intensities of another frame, and to use the reconstruction error as a supervisory signal. Thus, these methods can learn depth estimation without labeled data sets, and can take advantage of the vast amount and rich variability of video data available.

Advances on SIDE methods can impact a wide range of fields. Depth information is valuable for image and video manipulation applications such as image composition [37] and background replacement in videos [16]. The underlying relationships of blur and 3D geometry make depth information helpful for image deblurring [31]. In addition, depth can be used as a complementary source of information to enhance the performance of downstream tasks, for instance, dense depth maps can improve object detection methods [8], depth cues can help semantic segmentation [56], depth information is useful for video stabilization [44]. Moreover, several augmented reality (AR) applications require accurate depth estimation methods to obtain a representation of the 3D geometry. Efficient depth estimation methods are crucial in applications such as real-time navigation and shopping using mobile devices [80], AR assisted surgical procedures [5], AR assisted manufacturing processes [66], and potentially many other applications [50]. Furthermore, depth, as a representation of the 3D structure of the scene, is important for navigation in autonomous systems in real environments, for example, on autonomous driving [19].

1.1 Problem and Approach

The problem of estimating depth from a single image has been widely studied in computer vision [75]. It consists of predicting a dense depth map given a single input image. Figure 1.1 illustrates the input and output of SIDE methods.

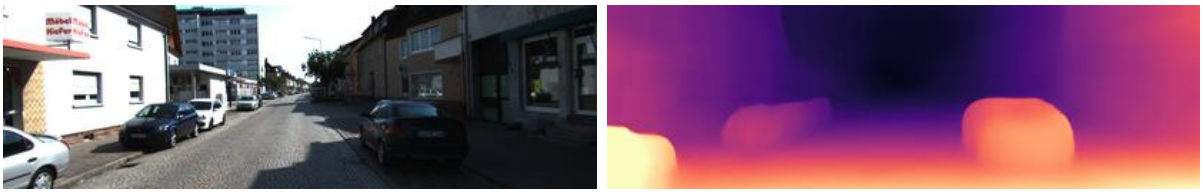


Figure 1.1: Input RGB image and its dense depth map prediction obtained with the method described in Chapter 6.

Self-supervised deep learning methods for SIDE rely on view reconstruction as pretext task. At training time, view reconstruction requires multiple views of an scene. These views can be obtained from monocular or stereo sequences. Assuming that we only consider a pair of view of a scene, we reconstruct one of the views, i.e., the target view, from the information contained in another view, that is, a source view. Reconstruction is possible because we can find pixel correspondences with consistent appearance on the

source and target views. In this thesis, we focus on methods that learn SIDE from monocular sequences. Although these method use monocular sequence for training, they only require a single image for inference.

View reconstruction is done by warping the source view into the target view using the pixel coordinate correspondences. We can obtain these correspondences using multi-view geometry relationships that require depth information of the target view and the relative camera motion between the source and target view. The depth maps as well as the relative camera motion can be learned with models, for instance, Convolutional Neural Networks (CNNs), using reconstruction error as a supervisory signal.

1.2 Challenges

In the previous section, we introduced the problem and the approach for self-supervised depth estimation. In this section, we will describe several challenges that emerge from characteristics of the problem and the approach.

1.2.1 Incorrect Pixel Correspondence Estimates

Incorrect correspondence estimates reduce the effectiveness of models trained with view reconstruction-based loss to converge on a solution that performs well on depth estimation. In this section, we describe key characteristics that are causes of incorrect pixel correspondence estimates.

Occlusion

View reconstruction performed either with depth maps or camera motion allows to warp the content of one view to another. The warping process consists of creating a mapping between all the pixels coordinates from the target view, the view to be reconstructed, to coordinates on a source view, the view where the pixel intensities come from. However, due to camera motion or moving objects, some pixels in the target view are occluded on the source view. Thus, depth estimates have high error rates on disoccluded regions in the target view because consistency cannot be enforced on these regions.

Moving Objects

Self-supervised methods that rely on view reconstruction as a supervisory signal and depend on a multi-view geometry relation that requires that the 3D structure of the scene be rigid among frames [97, 49, 2]. Thus, if this *rigid assumption* is violated, for instance, whether there are moving objects on the scene, the performance of models decreases. Recently, various works [7, 23, 62] have addressed this problem, showing promising results. Methods that depend on directly predicting the optical flow among frames do not have this limitation because they learn pixel displacements independently of the rigidity of the scene.

High-Frequency Structure Regions

Recent self-supervised dense depth estimation methods predict depth maps that capture the scene structure and motion reasonably. Most of these methods use pixel-wise or structure-aware loss functions that are able to learn low-frequency structures on the image, on the other hand, they perform poorly in regions with high-frequency structures.

Large Low-Textured or Texture-Less Regions

On the self-supervised context, depth estimation are ill-posed problems because there are various depth maps or flow fields that can satisfy geometric consistency. A common solution is to enforce continuity or smoothness on predictions penalizing large variations on regions with small intensity gradients [18]. However, even with this constraint, methods tend to fail in these regions [61]. Low-textured or texture-less regions do not occur only because of the material or the texture of the surface, but also because the regions can be overexposed.

Surface Reflection and Refraction

As self-supervised SIDE methods form structure and motion estimation rely on view reconstruction, appearance consistency among frames should be maintained. However, surfaces with highly specular reflection break this assumption. Moreover, transparent materials, which can also be specular depending on the illumination of the scene, could also be ignored because they can refract light.

1.2.2 Divergence of View Reconstruction and Depth Estimation Performance

Since self-supervised SIDE methods are not trained directly on depth estimation, but use view reconstruction as an auxiliary task, there is no guarantee that the convergence of a model to a good local optimum on view reconstruction implies that the model has converged to a proper solution for depth estimation. This observation is reinforced by a recent empirical evaluation [43] that suggests that networks trained using view reconstruction loss, after improving depth estimation to some extent, diverge from depth estimation error. Thus, the key challenge is to find approaches to improve the optimization objectives and reduce this divergence in training.

1.3 Objectives

The main goal of this thesis is to investigate approaches to address relevant challenges of self-supervised SIDE from monocular videos. To accomplish our aim, we dedicated our efforts to accomplish the following objectives:

- Evaluating and proposing approaches to handle regions without valid pixel correspondences in view reconstruction.

- Proposing an approach to leverage feature sharing between depth and camera motion networks.
- Proposing an approach to combine the consistency constraints on structure and feature representations used by depth models.
- Evaluating and proposing approaches to perform self-distillation via prediction consistency.
- Implementing a framework for the evaluation of self-supervised depth estimation approaches.

1.4 Research Questions

To achieve the objectives of our research, we formulated a set of research questions with the aim to bound the scope of the potential solutions and analysis.

How can we reduce the detrimental effect of pixels without correspondences on training neural networks for self-supervised SIDE?

The view-reconstruction approach to learn SIDE without labels require explicit mechanisms to reduce the detrimental effect of the corrupted gradients due to pixels without correspondences. In Chapter 4, we explore heuristics to filter out or to diminish the contribution of pixels without correspondences to the loss function. In Chapter 5, we explore an approach to diminish the error contribution of pixels that might not have correspondences using aleatoric uncertainty as weighting criterion.

How can we leverage the relationships of depth and camera motion feature representations?

We believe that similarities of depth and camera motion estimation tasks, such as their dependence of the geometric features, suggest that there might be complementary relationships between them. In Chapter 4, we propose an approach to share the feature representation learned by the depth network to the camera motion network.

How can we design a loss to enforce consistency between several feature representations learned by the neural networks?

When using frame sequences of three or more frames, we could have more than one valid correspondence by each pixel on the target view. In these cases, adaptive approaches, such as minimum re-projection to reconstruct the target view considering color features, are commonly used [22]. In Chapter 5, we extend minimum re-projection approach to enforce consistency between geometric representation and learned feature maps in addition to color information.

How can we provide additional learning signals to train neural networks for self-supervised SIDE?

The observation that the main optimization objectives used for training models with self-supervised SIDE methods are not fully correlated with depth performance suggest the necessity to investigate additional approaches to improve optimization on self-supervised SIDE methods. In Chapter 6, we propose an additional loss term that performs self-distillation via prediction consistency.

1.5 Contributions

The key contributions of this thesis are the following:

- A method that takes advantage of the availability of the feature presentations learned by the depth and camera motion models in the self-supervised depth estimation framework. Our method shares the features learned by depth estimation model to the camera motion model. Moreover, our method uses a heuristic based on a depth consistency constraint to diminish the error contribution of pixels without valid correspondences (see Chapter 4 for details).

- A method that addresses the presence of invalid correspondences by incorporating two improvements on the loss function: (1) the extension of minimum re-projection to enforce consistency between 3d-coordinate maps and deep features, and (2) the usage of aleatoric uncertainty to diminish the error contribution of pixels without valid correspondences (see Chapter 5 for details).

- A method that performs self-distillation via prediction consistency in self-supervised depth estimation. Our method incorporates per-pixel filtering strategies. Furthermore, we adapt and evaluate representative consistency enforcement strategies (see Chapter 6 for details).

- The development and public release of the source codes[1][2] that implement the methods presented in this thesis.

1.6 Publications

During the development of this thesis we published our results on international conferences dedicated to the communication of advances on the fields of computer vision, pattern recognition, and artificial intelligence.

- J. Mendoza, H. Pedrini. *Self-Supervised Depth Estimation Based on Feature Sharing and Consistency Constraints.* 15th International Conference on Computer Vision Theory and Applications (VISAPP). Valletta, Malta, pp. 134-141, February 27-29, 2020.

[1] `https://github.com/jmendozais/SDSSDepth`
[2] `https://github.com/jmendozais/DCFSGNet`

- J. Mendoza and H. Pedrini. *Adaptive Self-Supervised Depth Estimation in Monocular Videos.* 11th International Conference on Image and Graphics (ICIG). Haikou, China, pp. 687-699, August 06-08, 2021.
- J. Mendoza and H. Pedrini. *Self-Distilled Self-Supervised Depth Estimation in Monocular Videos.* 3rd International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI). Paris, France, pp. 423-434, June 1-3, 2022.

1.7 Thesis Organization

This thesis is organized as follows. In Chapter 2, we review the main concepts, approaches, and aspects of several works proposed in the literature for depth and optical flow estimation. In Chapter 3, we introduce the data sets and metrics. In Chapter 4, we propose methods to improve self-supervised depth estimation using feature sharing and consistency based occlusion detection. In Chapter 5, we develop two adaptive strategies enhance self-supervised depth estimation. In Chapter 6, we propose a self-distillation method based on prediction consistency. Finally, in Chapter 7, we present our concluding remarks.

Chapter 2

Background

Depth estimation is a longstanding task in computer vision. An early depth estimation strategy used edge information and perspective geometry to infer three-dimensional structure and, therefore, depth from a single image [63]. Learning-based approaches have also proposed for depth estimation. A method used for single-image depth estimation employed multi-scale Markov Random Field (MRF) model over relative and absolute features from the input image [69]. Furthermore, deep learning-based methods have also been successfully applied for depth estimation. Thus, a supervised approach proposed for depth estimation used a convolutional network to estimate a coarse depth map, and a second network to refine the results [14].

Another view of the brightness constancy assumption is as a source of supervision to warp or reconstruct one view to another using photometric error. Several methods for single-image depth estimation based on view reconstruction have been developed through stereo images for training. An early approach to learning single-image depth estimation used view reconstruction with the target and source views being the images in the stereo pair [18]. The view reconstruction process is required to know the relative position between the stereo pair and the intrinsic parameters of the camera. Thus, view reconstruction can be enforced between nearby views in the sequence frames of a monocular video [97]. In contrast with stereo methods, in which the relative position between cameras is already known, an additional model is used to estimate the relative position between the camera in different instants of time, expressed as an Euclidean transformation. Moreover, a recent approach proposed to learn the intrinsic parameters besides depth and the relative camera motion allowing to train the model with videos with unknown camera parameters [23].

In the following sections, we review relevant approaches and concepts, as well as describe aspects of several methods proposed in the literature for depth estimation.

2.1 Learning SIDE from Monocular Videos

The usage of view reconstruction as a main supervisory signal for depth estimation models was extensively explored in the literature. Most of these methods require to determine a mapping between pixel coordinates of a target and source views. A method for computing this mapping is through perspective projection and relative camera motion [97].

Perspective projection consists of transforming the coordinates of a point from the camera coordinates system to image plane coordinates. It requires to know the intrinsic parameters of the camera $\mathbf{K}$. Compactly, we can represent the perspective projection as follows:

$$h(x) = \pi(\mathbf{K}X) \tag{2.1}$$

where X is a target point in the camera coordinate system, $\mathbf{K}$ is a matrix that contains the camera intrinsic parameters, and π is a function that normalizes a homogeneous coordinate representation dividing their coordinate values by their z-coordinate, and $h(x)$ is the homogeneous coordinate of x, which is the target point in the pixel in the image plane. Similarly, the inverse perspective projection can be defined as follows:

$$X = \mathbf{D}(x)\mathbf{K}^{-1}h(x) \tag{2.2}$$

where $\mathbf{D}(x)$ is the depth value for the pixel x. This approach requires to back-project all pixels of the target image to the camera coordinate system. Thus, we need to estimate a dense depth map for the target image. Dense depth maps can be computed using a convolutional encoder-decoder network.

Relative camera motion is defined as the rotation and translation transforms that relate the coordinate systems that the camera had when the target $\mathbf{I_t}$ and the source views $\mathbf{I_s}$ were captured. Rotation and translation are Euclidean transformations, and can be expressed them as a single transformation matrix $\mathbf{T_{t\to s}} \in SE(3)$. Given a point of interest, we can use $\mathbf{T_{t\to s}}$ to map its coordinates from the target to the source camera coordinate systems as follows:

$$X_s = \mathbf{T_{t\to s}}X_t \tag{2.3}$$

where X_s and X_t are the coordinates of the point of interest on the source and target camera coordinate systems, respectively.

Given a pair of target and source views $(\mathbf{I_t}, \mathbf{I_s})$, we can estimate the Euclidean transformation $\mathbf{T_{t\to s}}$ between their camera coordinate system transformation using a convolutional network.

Using neural networks to estimate the depth map and the camera motion, and putting the explained geometric relationships together, we can estimate correspondences between the pixels in the target view $\mathbf{I_t}$ to the pixels in source view $\mathbf{I_s}$. Given a pixel x_t in the $\mathbf{I_t}$, its coordinate is back-projected to the camera coordinate system of the $\mathbf{I_t}$ using its depth value $\mathbf{D_t}(x_t)$, and the inverse of its intrinsic matrix $\mathbf{K}^{-1}$. Then, the relative motion transformation $\mathbf{T_{t\to s}}$ is applied to project the back-projected coordinate from the camera coordinate system of the $\mathbf{I_t}$ to the camera coordinate system of $\mathbf{I_s}$. Finally, transformed coordinates are projected into the source image x_s. We represent this mapping in the following equation:

$$h(x_s) = \pi(\mathbf{K}\mathbf{T_{t\to s}}\mathbf{D_t}(x_t)\mathbf{K}^{-1}h(x_t)) \tag{2.4}$$

where $h(x)$ is a function that maps a pixel x to its homogeneous representation and π is a function that normalizes homogeneous coordinates dividing their values by the last coordinate.

Figure 2.1 depicts the frame-to-frame pixel coordinate mapping graphically. The re-

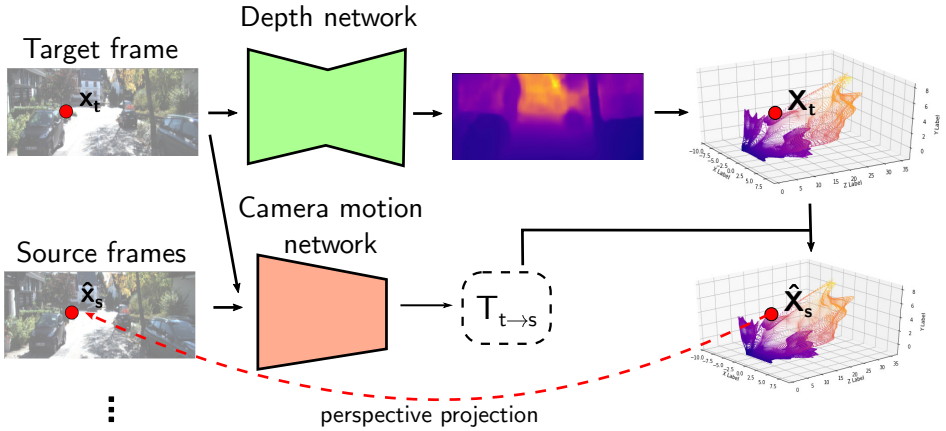


Figure 2.1: Pixel coordinate mapping. We show the pixel coordinate mapping between a source and a target frame using perspective projection and the relative camera motion $\mathbf{T_{t\to s}}$.

sulting coordinates can be floating points. Thus, bilinear interpolation is used to compute the pixel intensity values [97].

Once we know the projected coordinates and, therefore, the pixel intensities in the source image plane for each pixel in the target image, we can reconstruct the target frame. This process is also known as image warping. Let us define this operation as follows:

$$\hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}} = w_d(\mathbf{I_s}, \mathbf{D_t}, \mathbf{T_{t\to s}}) \tag{2.5}$$

where $\hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}}$ is the reconstructed image and w_d is the warping function.

2.1.1 View Reconstruction Loss

Depth and camera motion models are trained using the view reconstruction as a supervisory signal. View reconstruction is possible because we know that multiple views of the same scene have consistent properties. For instance, source and target source should be photometrically consistent, that is, the brightness information of both views should be consistent. Thus, the brightness difference between the target and the source view warped to the target view should be minimal. We show the view reconstruction objective in the following equation:

$$\mathcal{L}_{rec} = \sum_{\mathbf{I_s}\in\mathcal{I}_s} \rho(\mathbf{I_t}, \hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}}) \tag{2.6}$$

where $\mathcal{I}_s$ is a set of source views to the target view, and ρ is function that measure the brightness dissimilarity. We can consider the previous and next frames in the sequence of frames as source views, that is, $\mathcal{I}_s = \{\mathbf{I_{t-1}}, \mathbf{I_{t+1}}\}$, as in [97] used for depth estimation, or we can just consider the previous frame, that is, $\mathcal{I}_s = \{\mathbf{I_{t-1}}\}$.

The dissimilarity function could be any distance function that can be applied to a pair of images, such as L1 distance that was used earlier for depth estimation [18] or a combination of the L1 distance and the structure dissimilarity function (DSSIM) that has been widely used in recent works. This dissimilarity function [47], which is also known as

photometric consistency loss, is defined as follows:

$$\rho(\mathbf{I_t}, \hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}}) = \alpha_r \left(\frac{1 - \text{SSIM}\big(\mathbf{I_t}, \hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}}\big)}{2} \right) + (1 - \alpha_r) \big| \mathbf{I_t} - \hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}} \big| \tag{2.7}$$

where α_r is the trade-off parameter to combine the $L1$ and DSSIM and SSIM is the structure similarity function [83].

2.1.2 Smoothness Regularization

Moreover, photometric error is not-informative in homogeneous regions since in these regions multiple depth values can produce equally good reconstructions [18, 34]. Most methods in the literature address this problem using a *smoothness constraint*, which enforces continuity on predictions. Several smoothness constraints have been explored in the literature. A common issue about the smoothness constraint is that we cannot enforce it in edge regions, where depth variations are natural. A heuristic to overcome this issue is to reduce the loss inversely proportional to the image gradient in order to preserve edges. Equation 2.8 shows an instance of an edge-preserving depth smoothness loss used in [21, 92].

$$\mathcal{L}_{ds} = \sum_{x \in \Omega(\mathbf{I_t})} |\nabla \mathbf{D_t}(x)| \big(\exp\big(|\nabla \mathbf{I_t}(x)| \big) \big)^{\intercal} \tag{2.8}$$

where $\Omega(\mathbf{I_t})$ is the set of pixel coordinates of $\mathbf{I_t}$, and $\mathbf{D_t}$ is the corresponding depth map.

2.1.3 Multi-Scale Architecture

In this work, we focus on methods that use convolutional encoder-decoder networks as depth models. An advantage of these networks is that they allow to implement the coarse-to-fine strategies used by early works in the literature of depth estimation [14, 30]. Figure 2.2 shows a multi-scale convolutional encoder-decoder for depth estimation.

Thus, we can use view reconstruction and the smoothness constraint at various scales as a supervisory signal. Then, depth network can be trained with loss functions, shown in Equation 2.9.

$$\mathcal{L}_{total} = \sum_{i \in S} \mathcal{L}_{rec}^{(i)} + \lambda_{ds} \mathcal{L}_{ds}^{(i)} \tag{2.9}$$

where S is the set of desired scales.

2.1.4 Occlusion Handling

Consistency cannot be enforced in occluded regions. Several methods deal with occluded regions by detecting and excluding them from the error computation. An approach to detecting occlusion is to use thresholding to categorize the pixels with larger inconsistencies as occluded.

Similarly, another approach to enforcing consistency is to penalize as the difference between the forward and the inverse of the backward flow fields, while the threshold can be set proportional to the magnitude of the flows [98]. A similar approach to verifying

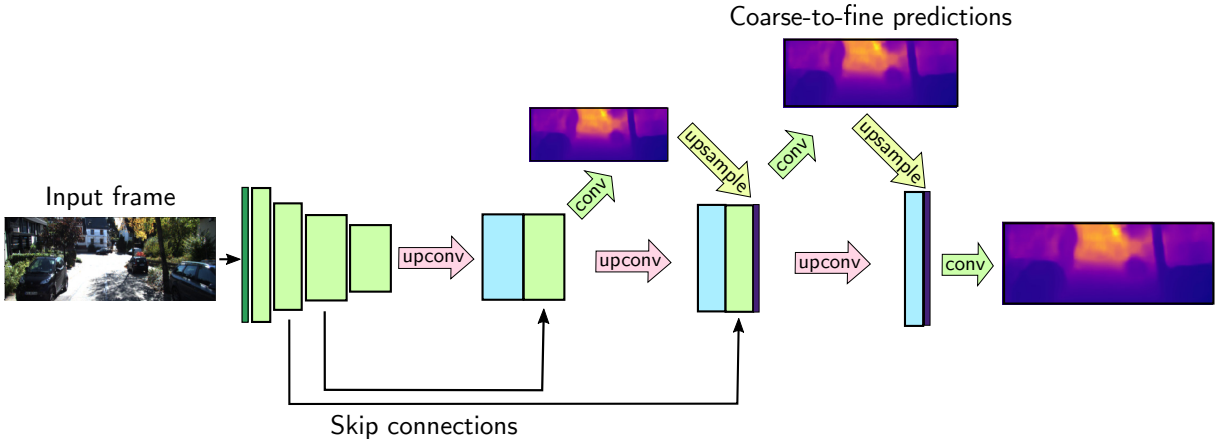


Figure 2.2: Coarse-to-fine strategy using a convolutional encoder-decoder network. The feature maps at each level of the decoder have two uses. Fist, they are used to predict a depth map using a convolution and an activation function. The activation limits the output to valid depth values. Second, they are used to generate part of the features for the next level of the decoder using a convolution followed by an upscale operation. The features of the next level of the decoder are the concatenation of the up-scaled feature maps, the up-scaled depth maps, and the feature maps of the skip connections.

if a pixel in the target view is occluded consists of checking if there are no pixels on the source view that will land in its neighborhood, using the predicted flow [48]. Another approach consists of using an additional model to estimate a mask, which excludes regions in the target image that cannot be explained in the warping process, probably, due to occlusions [97].

View reconstruction-based depth estimation methods that consider a target with several source views can leverage that regions in the target view that are occluded in one source view may not be occluded in another source view. Thus, when reconstructing a region that is occluded in one source view, for instance, a frame before the target frame, another view where this region is not occluded will reconstruct it with less error, for instance, a frame after the target frame, which can be accomplished using a pixel-wise min operator over the photometric error maps [22, 7]. Equation 2.10 shows this error function, also known as *minimum re-projection loss*.

$$\mathcal{L}_{rec} = \min_{\mathbf{I_s} \in \mathcal{I}_s} \Big(\rho\big(\mathbf{I_t}, w(\mathbf{I_s}, \mathbf{D_t}, \mathbf{T_{t \to s}})\big) \Big) \tag{2.10}$$

Moreover, depth information can be used to detect occluded regions. A strategy for determining if a pixel in the target view is occluded in a source view is to verify that, when its translated to the coordinate system of the source view, it is behind another pixel [23].

2.1.5 Multi-View Consistency

Nearby frames should have consistent structure representations since they correspond to the same scene. An approach to enforcing consistency on predictions is to penalize the difference between the depth maps of the target and the source views on the same coordinate system. For example, a method estimated the depth value for each pixel of the

target view in the image coordinate system of the source view using the camera motion transformation, in order to penalize their difference with depth values according to the source view [96].

A similar approach is to enforce consistency not only with the depth values but with the 3D coordinates of the target and source views [9]. When transforming depth or 3D coordinates to the source coordinate system, the point may lie in floating-point locations in the source view. Then, the source view values can be obtained using interpolation. An alternative strategy when using 3D coordinates is to compute the difference using the Iterative Closest Point (ICP) algorithm [49].

Consistency can be enforced not only on depth or 3D coordinates but also the feature maps. One approach is to warp depth feature maps of the source view to the target view, in a similar way as view reconstruction is done, to penalize its difference with the feature maps of the target view [58, 59]. Another approach leverages structure, appearance, and feature consistency to improve depth and camera motion estimates through a differentiable Levenberg-Marquardt algorithm [71, 76]. Additionally, while most of the methods enforce consistency between adjacent views, considering consistency between not adjacent views in the neighborhood can boost the performance [96].

2.2 Knowledge Distillation

Knowledge distillation (KD) is a technique created to transfer information from one model to another. Early iterations of KD methods focused on transferring information from a powerful model that has certain limitations for deployment to a model that does not have these limitations. The powerful model could be a single large neural network or an ensemble. These models have limitations to be deployed on devices with low computational resources or applications with strict time constraints. Thus, knowledge distillation can be used to transfer information from these powerful and cumbersome models to smaller models that could meet the time and resource constraints with a minimal performance reduction. Furthermore, later iterations of KD methods have shown that transferring information from source to target model with similar characteristics also benefits the target model.

2.2.1 Teacher-Student Structure

The teacher-student structure is a common characteristic of methods that aim transfer information from a source model or ensemble, which takes the role of the teacher, to a target model, which takes the role of the student. Teacher-student structure has been widely used in KD methods.

Moreover, the teacher-student structure has been extensively used in semi-supervised learning. In semi-supervised learning the teacher network is trained with labeled data and predicts pseudo-labels for the unlabeled data. Then, the student learns with the labeled data and the unlabeled (pseudo-labeled) data.

2.2.2 Applications

In classification problems, KD is performed by transferring the information about how likely an input example belongs to the incorrect classes, in addition to the correct class. This information, also known as *dark knowledge*, provides insights about how the source model tends to generalize [29]. Dark knowledge is transferred by minimizing the difference between the logits or smoothed probabilities obtained from the teacher and student networks.

In regression problems, the output of the networks have similar characteristics as the ground-truth. In SIDE, for instance, the output could be a two-dimensional matrix with continuous values that is similar to the dense depth ground truth. Thus, we do not have access to any dark knowledge. However, it has been shown that with additional considerations [68, 94] student network can still benefit from the information provided by the teacher network.

2.2.3 Self-Distillation

There are challenging scenarios in which the KD can not be applied effectively, for instance, when the computational resources available are not good enough to train a large model, when training a large teacher model is too challenging, or when the method cannot efficiently transfer information from the teacher to the student models. One approach to deal with these scenarios is self-distillation. Self-distillation (SD) is a technique to transfer information from the target model to itself without using external models. Self-distillation has other advantages such as it does not require teacher model selection.

There are several methods to perform self-distillation [82]. One approach is the sequential SD, which consists of training models in multiple sequential stages, where the student model trained in a previous stage becomes a teacher to train the current student model. In SD based on data transformation the student learns consistent representations for input examples that are exposed to transformations. Another approach, SD via deep supervision, is applied to student models with multiple output branches at different layers. Based on the observation that deeper branches can provide useful information to shallower layers, this strategy uses the deeper branches as teacher and the shallower branches as student.

Chapter 3

Materials and Metrics

In this chapter, we describe the data sets and the evaluation metrics used in the methods that will present in Chapters 4, 5 and 6. In addition, we describe the hardware and software resources used to implement the methods.

3.1 Dataset

KITTI benchmark [20] is one of the most used data sets for the evaluation of depth estimation methods. It was created with the purpose reduce the bias and complement available benchmarks with real-world data. It is composed of video sequences with 93 thousand images acquired through high-quality RGB cameras captured by driving on rural areas and highways of a city. For depth estimation, each image has a sparse depth ground-truth provided by a Velodyne LIDAR scanner. The LIDAR scanner and RGB cameras capture information of the scene at 10 frames per second.

As several of depth estimation methods available in the literature, we plan to use the Eigen split [14] for evaluation. It contains 40K images for training, 4K images for validation, and 687 images for testing. For optical flow estimation, the benchmark provides 200 training and 200 testing scenes [53]. Figure 3.1 shows samples from the KITTI benchmark with depth/disparity and optical flow annotations.

3.2 Evaluation Metrics

We used the error metrics defined by Eigen et al. [14], which are extensively used in the evaluation protocols of other methods in the literature: Absolute Relative Difference, Squared Relative Difference, Root Mean Squared Error (RMSE), and Log RMSE. These metrics can be expressed as follows:

$$E_{\text{Abs Rel}} = \frac{1}{|D|} \sum_{d \in D} \frac{|d - d^*|}{y^*} \tag{3.1}$$

$$E_{\text{Sq Rel}} = \frac{1}{|D|} \sum_{d \in D} \frac{||d - d^*||^2}{y^*} \tag{3.2}$$

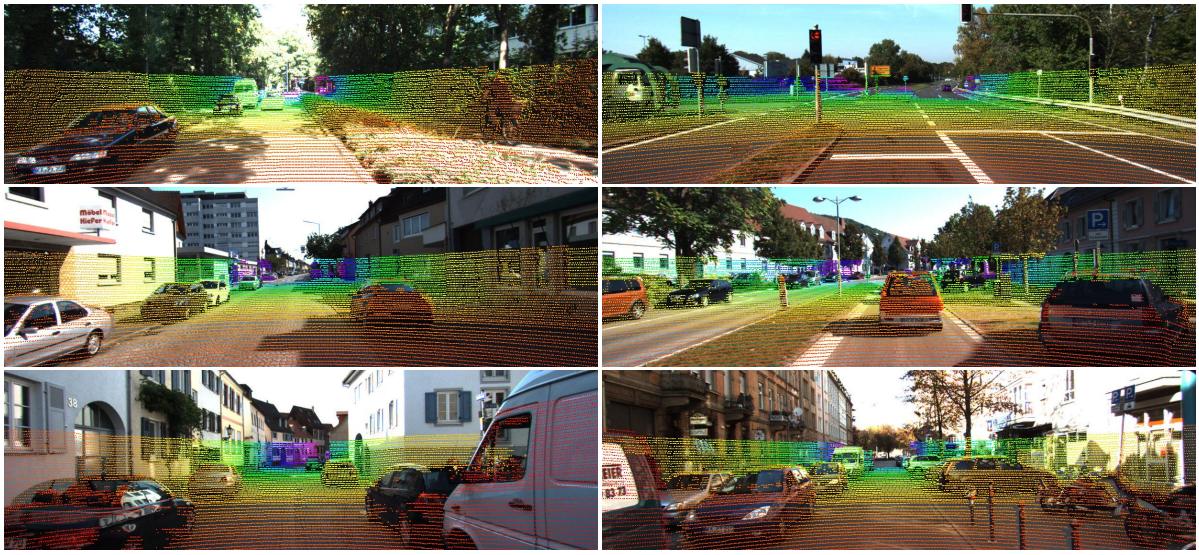


Figure 3.1: Examples from the KITTI dataset. Each example displays an RGB image and the sparse LIDAR measurements captured from a scene.

$$E_{\text{RMSE}} = \sqrt{\frac{1}{|D|}\sum_{d\in D}||d - d^*||^2} \tag{3.3}$$

$$E_{\text{Log RMSE}} = \sqrt{\frac{1}{|D|}\sum_{d\in D}||\log d - \log d^*||^2} \tag{3.4}$$

where d is a predicted depth value, d^* is the ground-truth depth value for d, and D represents the sets of values on the predicted depth map.

Moreover, we used *thresholded accuracy metric*, which is the proportion of depth values with a ratio of the predicted to ground-truth value in the interval $< \frac{1}{\delta}, \delta >$. Similar to previous works, we computed the proportion for the intervals defined by δ values equal to 1.25, 1.25^2 and 1.25^3.

$$E_\delta = \frac{1}{|D|}\sum_{d\in D}\left[\max\left(\frac{d}{d^*}, \frac{d^*}{d}\right) < \delta\right] \tag{3.5}$$

where [.] is the Iverson bracket operator.

3.3 Hardware and Software Resources

The experiments required to conduct in this thesis demand high computational cost due to the image processing, computer vision and deep learning algorithms that we need to execute. Most of this computational burden can be executed effectively using GPU hardware. We use the equipment available in the Laboratory of Visual Informatics of the Institute of the Computing at UNICAMP, which have several computers with GeForce GTX 1080 Ti and a TITAN V GPU cards with 11 and 12 GB of memory, respectively.

Python programming language has been used to implement our framework due to

its availability of open-source libraries for image processing, computer vision, and implementing deep neural networks. We have used several libraries such as NumPy for linear algebra operations in CPU; OpenCV, PIL, and Scikit-Image for computer vision and image processing operations; and TensorFlow and PyTorch for deep learning.

Chapter 4

Self-Supervised Depth Estimation Based on Feature Sharing and Consistency Constraints

In this chapter, we propose a self-supervised method for depth and camera motion estimation in monocular videos. Inspired by multi-task learning literature, where various methods have been proposed that take advantage of the task similarity and share representations between tasks, we propose to share the representations of depth network to camera motion network. Specifically, we use the feature maps of each layer in the encoder part of the depth network as input to the camera motion network by projecting and summing them to their task-specific feature maps.

Moreover, we investigate a constraint that decreases the error contribution of regions with inconsistent projected depth values. Thus, the model does not lose supervision in the early stages of training, where depth estimates are more prone to have inconsistencies.

This text is organized as follows. In Section 4.1, we review some relevant methods related to the topic under investigation. In Section 4.2, we present the proposed self-supervised depth estimation methodology. In Section 4.3, we describe and evaluate the experimental results. In Section 4.4, we conclude the chapter with some final remarks. Our code is available at `https://github.com/jmendozais/DCFSGNet`.

4.1 Related Work

In this section, we briefly review some relevant approaches available in the literature related to the topics explored in our work.

4.1.1 Geometric Constraints and Occlusion

The usage of geometric constraints to deal with occlusion have been widely explored in the literature. An approach proposed by Yin and Shi [92] penalized optical flow inconsistencies between the flow predictions obtained from depth and camera motion estimates as well as an optical flow network prediction, in the forward and backward direction.

Luo et al. [48] proposed to penalize depth inconsistency by projecting depth maps between adjacent frames using the respective camera motion transformation, besides optical flow consistency Similarly, Zhou et al. [96] penalized depth inconsistencies not only between adjacent frames but between each pair of frames in the neighborhood.

Mahjourian et al. [49] and Chen et al. [9] penalized the difference between the predicted depth maps back-projected to the same reference three-dimensional coordinate system. Moreover, they also penalized inconsistencies of the optical flow prediction obtained from the depth and camera motion estimates and the flow predicted with another network. In addition to prediction error, depth inconsistencies occur in regions that are not explainable because occlusion.

Other methods use geometric constraints to ignore occluded or disoccluded regions on the reconstruction loss [23, 48, 98]. However, we observed that geometric inconsistent regions are common in earlier stages of training, and the model loss the supervisory signal to those regions if they are completely ignored. In contrast, we decrease the error contribution on those regions instead of removing them.

4.1.2 Multi-Task Architecture

Representation sharing is an important aspect of multi-task learning because it has advantages such as the reduction of over-fitting and the reduction of processing time. However, determining a proper degree of representation sharing is not trivial.

Several approaches have been proposed to train various similar tasks simultaneously with some degree of representation sharing. For instance, in convolutional neural networks, this representation sharing can be tuned by the number of layers shared by various networks trained to learn different tasks. Thus, networks with a few shared layers will share less information than networks with more shared layers.

Misra et al. [54] shared information by computing the representation at each level of the network as a linear combination of the representations obtained from the last level. Similarly, Ruder et al. [64] computed a half representation as a linear combination of the output of previous layers, keeping the other half as a task specific representation. Liu et al. [46] used a network to learn global representations and task-specific networks that use attention modules to learn task-specific representation from global representations.

In this work, we show that sharing depth network representation to the camera motion network can improve our model performance.

4.2 Proposed Method

We summarize our method in Figure 4.1. In this section, we give an overview of the formulation used for depth and camera motion estimation. Then, we describe our geometric consistency constraint and feature sharing mechanism. Finally, we present architecture considerations of our neural network.

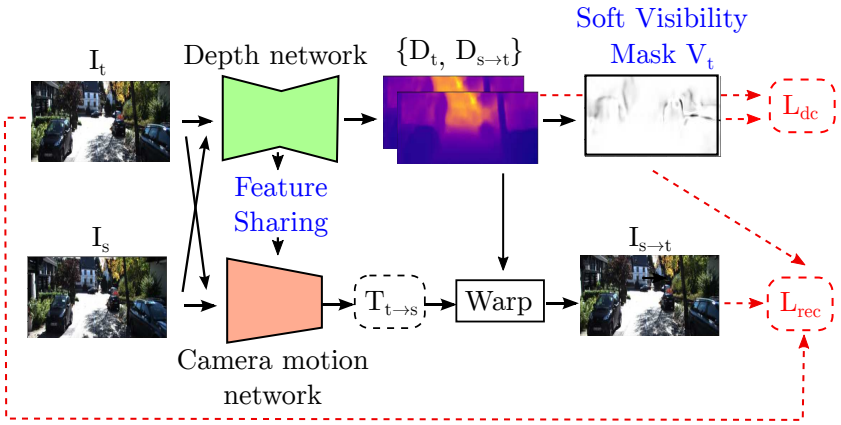


Figure 4.1: Overview of our method. The depth network is used to predict the depth maps for the source I_s and target I_t images. The camera motion network predicts the Euclidean transformation between the target and source camera coordinate systems $T_{t\to s}$. A soft-visibility mask is computed based on the target depth map and the projected source depth map $D_{s\to t}$. Feature maps of the depth network are shared with the camera motion network. Depth consistency and reconstruction loss terms are computed considering the soft visibility mask V_t.

4.2.1 Overview

The core idea of self-supervised depth estimation is that, given two views of the same scene, we can reconstruct one of the views, that is, the target view $\mathbf{I_t}$, from the other view, that is, the source view $\mathbf{I_s}$. Thus, reconstruction error is used to guide the learning of the model.

Reconstruction is done through perspective projection and the relative camera motion between a pair of views. Perspective projection requires to have the intrinsic parameters of the camera $\mathbf{K}$, and the depth values for each pixel in the target image. We obtain depth values using a convolutional encoder-decoder network D that learns to estimate a dense depth map $\mathbf{D}$ for an input image $\mathbf{I}$.

The relative camera motion is represented by an Euclidean transformation $\mathbf{T}_{t\to s} \in SE(3)$ between the coordinate systems that the camera had when the target view $\mathbf{I_t}$ and the source view $\mathbf{I_s}$ were captured. Given a pair of views $\left(\mathbf{I_t}, \mathbf{I_s}\right)$, we estimate its motion transformation $\mathbf{T}_{t\to s}$ using a convolutional network M. We reconstruct the target frame by projecting each pixel coordinate from the target view to the source view. Given a pixel x_t in the target frame, its coordinate is back-projected to the camera coordinate system of the target view using the inverse of its intrinsic matrix $\mathbf{K^{-1}}$. Then, the relative motion transformation $\mathbf{T_{t\to s}}$ is applied to project the coordinates form the coordinate system of the target view to the coordinate system of the source view. Finally, coordinates are projected to pixel coordinates on the source view. Equation 4.1 shows this mapping as follows:

$$h(x_s) = \pi\left(\mathbf{K}\mathbf{T}_{\mathbf{t}\to\mathbf{s}}\mathbf{D}_{\mathbf{t}}(x_t)\mathbf{K}^{-1}h(x_t)\right) \tag{4.1}$$

where $h(x)$ is the homogeneous representation of the pixel x, and π is a function that normalizes homogeneous coordinates dividing their values by the last coordinate. The resulting coordinates can be floating points. Thus, bilinear interpolation is used to compute the

pixel intensity values [97]. Using these pixel coordinates and intensity correspondences, we reconstruct the target frame as $\mathbf{I_{s\to t}}(x_t) = \mathbf{I_s}(x_s)$.

Then, we use the reconstruction error for training. Equation 4.2 expresses the reconstruction loss term as follows:

$$\mathcal{L}_{rec} = \sum_{\mathbf{I_s}\in\{\mathbf{I_{t-1}},\mathbf{I_{t+1}}\}} \mathbf{M_{t\to s}}\rho\big(\mathbf{I_t}(x_t), \mathbf{I_{s\to t}}(x_t)\big) \tag{4.2}$$

We consider the two adjacent frames of the target as source frames. ρ is a dissimilarity function. In addition, we use the principled mask $\mathbf{M_{t\to s}}$ proposed by Mahjourian et al. [49] to ignore pixels that became not visible because of the camera motion. As several works of the literature, we use a photometric consistency loss (See Section 2.1.1 for details). However, photometric consistency loss is not-informative in homogeneous regions since in these regions multiple depth assignments can produce equally good reconstructions [18]. This problem can be addressed by enforcing continuity on depth maps. We use the edge-preserving local smoothness term described in Section 2.1.2.

In addition, the depth network is designed to predict depth maps at multiple scales to address the gradient locality problem [97, 18]. Thus, we train the model with following loss function:

$$\mathcal{L} = \sum_{i\in S} \mathcal{L}_{rec}^{(i)} + \lambda_{ds}\mathcal{L}_{ds}^{(i)} \tag{4.3}$$

where S is the set of desired scales.

The described considerations are used for our baseline method. Both the depth network D and the camera motion network M are trained jointly in an end-to-end manner.

4.2.2 Depth Consistency and Occlusion

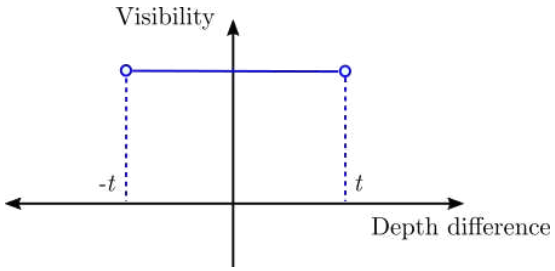
Depth map and camera motion predictions determine implicitly a flow field that contains the displacement of each pixel coordinate from the target frame to the source frame. This flow field allows us to warp not only the source frame appearance I_t but also its dense depth map D_t. For instance, we can warp the depth map of the source frame D_s to the target frame.

Then, the depth maps predicted for the target frame D_t should be consistent with the warped depth map $D_{s->t}$. Depth consistency can also be enforced in the inverse direction, that is, in the forward and backward direction.

$$\mathcal{L}_{dc} = \sum_{x\in\Omega(\mathbf{I_t})} \big|\mathbf{D_t}(x) - \mathbf{D_{s\to t}}(x)\big| \tag{4.4}$$

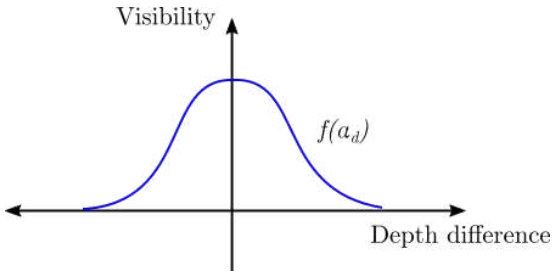
Depth consistency does not hold for all pixels in the image because of the occlusions and disocclusions produced by camera motion or by moving objects in the scene. Some works use this prior to create a visibility mask that hide or decrease the error contribution of pixels that have large inconsistencies.

Inconsistency at each pixel in the target image can be measured as absolute value of normalized difference between the predicted depth value on the target image, and the



(a) thresholded visibility function

(b) thresholded visibility mask



(c) soft visibility function

(d) soft visibility mask

Figure 4.2: Thresholded and soft visibility masks. In the first row, we show (a) thresholded visibility as a function of the normalized depth difference and (b) its thresholded visibility mask. On the second row, we show (a) soft visibility as a function and (d) its soft visibility mask.

depth value of the source depth map projected to the target camera coordinate system. As defined in Equation 4.5, we can compute a visibility mask by thresholding the inconsistencies along the target image with a threshold t obtained empirically. Equation 4.5 depicts this relation.

$$\mathbf{V_t}(x) = \left[\left|\frac{\mathbf{D_t}(x) - \mathbf{D_{s\to t}}(x)}{\mathbf{D_t}(x)}\right| < t\right] \tag{4.5}$$

where [.] is the Iverson bracket operator.

However, the networks do not produce accurate predictions on training, and a binary visibility mask do not explicitly handle the inconsistency variability. Thus, instead of ignoring several regions in the reconstruction loss, we propose to reduce the error contribution of inconsistent regions mapping normalized per-pixel depth differences to visibility values using a Gaussian function. This idea is used to built our soft-visibility mask as follows:

$$\mathbf{V_t}(x) = e^{-\alpha_d\left(\frac{\mathbf{D_t}(x) - \mathbf{D_{s\to t}}(x)}{\mathbf{D_t}(x)}\right)^2} \tag{4.6}$$

where α_d controls the smoothness degree of the visibility mask. Figure 4.2 shows the inconsistency-visibility and visibility masks obtained with thresholding or with a Gaussian on the inconsistencies.

We apply the visibility mask to the depth consistency and reconstruction loss terms as follows:

$$\mathcal{L}_{dc} = \sum_{x\in\Omega(\mathbf{I_t})} \mathbf{V_t}(x)\left|\mathbf{D_t}(x) - \mathbf{D_{s\to t}}(x)\right| \tag{4.7}$$

$$\mathcal{L}_{rec} = \sum_{\mathbf{I_s} \in \mathcal{I}_s} \mathbf{V_t} \mathbf{M_{t \to s}} \rho(\mathbf{I_t}, \mathbf{I_{s \to t}}) \tag{4.8}$$

where $\mathcal{I}_s$ is a set of source views to the target view, and ρ is dissimilarity function. We consider the previous and next frames in the sequence of frames as source views, this is $\mathcal{I}_s = \{\mathbf{I_{t-1}}, \mathbf{I_{t+1}}\}$.

Putting the loss terms together, the final loss function is the following:

$$\mathcal{L} = \sum_{i \in \mathcal{S}} \mathcal{L}_{rec}^{(i)} + \lambda_{ds} \mathcal{L}_{ds}^{(i)} + \lambda_{dc} \mathcal{L}_{dc}^{(i)} \tag{4.9}$$

where $\mathcal{S}$ is the set of scales, λ_{ds} is the weight factor for the depth smoothness term, and λ_{dc} is the weight factor for the depth consistency term.

4.2.3 Depth Encoder Feature Sharing

It has been shown that using a network with some degree of representation sharing can be better than using separate networks [54, 13, 46], mainly because individual tasks can be reinforced with the representation of other tasks and also because feature sharing allows representations to avoid over-fitting in individual tasks, but to be useful in other tasks.

In our context, where estimation of depth and camera motion operates simultaneously with the same input data and where tasks are complementary because the geometric formulation provides supervision to both networks with the same loss function, this motivates us to believe that representation sharing can improve the model performance.

Figure 4.3 illustrates our feature sharing mechanism. We propose to share the feature maps of the depth encoder with the camera motion network. This allows the camera motion network to leverage the depth features to improve the pose estimation. Moreover, better pose estimates can potentially improve the reconstruction and, as a consequence, depth estimation.

Equation 4.10 summarizes our proposal. Given a target frame $\mathbf{I_t}$ and its source frames $\mathcal{I}_s = \{\mathbf{I_{t-1}}, \mathbf{I_{t+1}}\}$, our depth and camera motion networks produces feature maps from these frames at each layer of the network. Thus, $\mathbf{F}_{D,\mathbf{t}}^{(l)}$, $\mathbf{F}_{D,\mathbf{t-1}}^{(l)}$, and $\mathbf{F}_{D,\mathbf{t+1}}^{(l)}$ are the features at the layer l of the depth network on the target and source frames, respectively, $\mathbf{F}_{M}^{(l)}$ are the features of the camera motion network at the layer l. In addition, we apply a non-linear transformation f over the concatenated depth representations $\mathbf{F}_{D}^{(l)}$. For simplicity, we set f to be convolution layer with 1×1 filters. We set the output of f to have the same amount of feature maps of the camera motion network in the same layer level. Finally, we sum the transformed features to the camera motion features as follows:

$$\mathbf{F}_{MD}^{(l)} = \mathbf{F}_{M}^{(l)} + f\big([\mathbf{F}_{D,\mathbf{t}}^{(l)} : \mathbf{F}_{D,\mathbf{t-1}}^{(l)} : \mathbf{F}_{D,\mathbf{t+1}}^{(l)}]\big) \tag{4.10}$$

4.2.4 Network Architecture

Finally, we briefly describe the network architecture. Our depth encoder-decoder network is based on the DepthNet [92]. Its encoder network is based on the ResNet50.

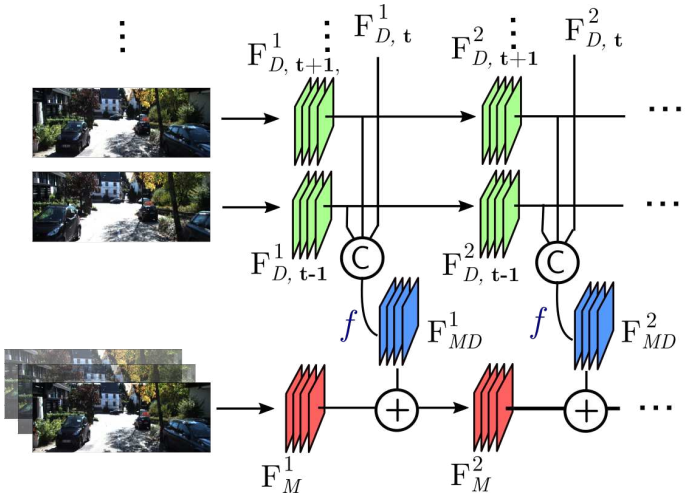


Figure 4.3: Feature sharing mechanism. The feature maps in the depth and camera motion network are shown with green and red colors, respectively. "C" represents the concatenation operation. f is a function that transforms the concatenated depth features. "+" represents the element-wise sum operation.

Its decoder network is composed of deconvolutional layers that up-sample the bottleneck representation in order to upscale the feature maps to the input resolution.

The encoder network has skip connections with the decoder network. In addition, we use dropout after the last two layers of the encoder and the first two of the decoder network to reduce over-fitting. In addition, we use bilinear interpolation for up-sampling instead of nearest-neighbor interpolation to produce more accurate depth maps.

Our camera motion network predicts the relative motion between two input frames. The relative camera motion has a 6-DoF representation, that is, the rotation angles and the translation vectors. We use the architecture proposed by Zhou et al. [97].

4.3 Experiments

In this section, we describe and evaluate the experimental results achieved with the preliminary implementation of our method.

4.3.1 Experimental Setup

We describe here the parameters of our model and the optimization method used in the learning process, the dataset used to train and evaluate the models and, finally, the metrics used to assess the model performance.

Parameter Setup

We used a trade-off parameter $\alpha_r = 0.85$ in the reconstruction loss term. The weights of the depth smoothness λ_{ds} and depth consistency terms λ_{dc} are 0.5 and 0.31, respectively. We employed a smoothness parameter of the visibility map $\alpha_d = 2$. We used a threshold $t = 0.3$ for the alternative version of our method with the thresholded visibility mask. We

applied Adam optimization with parameter $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We chose a batch size of 4.

The input images are re-scaled to 128×416 pixels. Furthermore, we apply random scaling, cropping, and various color perturbations to the input images in the data augmentation stage to reduce over-fitting. Depth and camera motion networks are trained from scratch.

Dataset

We used the KITTI benchmark (Section 3.1), composed of video sequences acquired by RGB cameras, and with sparse depth ground-truth provided by Velodyne LIDAR scanner. As several works available in the literature, we used the Eigen split [14] for evaluation. It contains 40K images for training, 4K images for validation, and 687 images for testing.

4.3.2 Depth Estimation

In this section, we present our experiments. First, we perform ablative experiments to analyze the impact of each contribution on the performance of our model. Then, we compare our results with other self-supervised depth estimation methods categorized into three groups: methods that assume a static scene, methods that explicitly model moving objects on the scene, and methods that perform parameter or output fine-tuning at test time.

Ablation Study

Table 4.1 shows the performance of variants of our model. It is possible to observe that the addition of depth consistency and either a hard or soft visibility masks is the major source of improvement. Moreover, we can see that the soft visibility map is slightly better than the thresholded visibility mask. It is also possible to observe that the complete model obtains better results than just considering depth consistency and visibility mask.

Table 4.1: Ablation analysis. We compare the performance of several variants our method. First, we present the results of our baseline model. Then, we show that the addition of the depth consistency term (DC) and the thresholded visibility mask (TV) improved the performance of the baseline. Furthermore, our results show that the use of a soft visibility mask (SV) improves almost all the evaluation metrics. Finally, we present the results achieved with our final model through depth consistency, soft visibility mask and feature sharing (FS). The best result achieved for each metric is highlighted in bold.

Method	**↓ Lower is better**				**↑ Higher is better**		
	Abs Rel	**Sq Rel**	**RMSE**	**LRMSE**	$\delta = 1.25$	$\delta = 1.25^2$	$\delta = 1.25^3$
Baseline	0.150	1.266	5.864	0.232	0.803	0.932	0.973
Ours w/ DC & TV, w/o FS	0.141	1.061	5.679	0.222	0.809	0.936	0.976
Ours w/ DC & SV, w/o FS	0.141	**1.029**	5.536	0.219	0.811	0.939	0.977
Ours	**0.138**	1.030	**5.394**	**0.216**	**0.820**	**0.941**	**0.977**

State-of-the-Art Comparison

In Table 4.2, we compare our method with the state-of-the-art methods. We split the competing methods into three groups. In the first group, we consider methods with similar settings to our method. Our method obtains better results in almost all metrics. In the second group, our method is compared with methods that explicitly address moving objects. Our method obtained competitive results. Finally, our method is compared with methods that explicitly address moving objects and perform test-time fine-tuning. Competing methods obtained better results than our method. Test-time fine-tuning is important to obtain better results, but these gains come at the cost of higher inference times [7, 9].

Table 4.2: Results of depth estimation on the Eigen split of the KITTI dataset. We compare our results against several methods of the literature. Methods are categorized into three groups: methods that assume rigid scenes, methods that explicitly model moving objects, and methods that perform fine-tuning on the last layer of the network besides considering moving objects. (*) indicates newly results obtained from an official repository. Column "M" indicates whether the method has address moving objects explicitly. Column "F" indicates whether the method performs test-time fine-tuning. The best result achieved for each metric is highlighted in bold.

Method	**↓ Lower is better**				**↑ Higher is better**				
	Abs Rel	**Sq Rel**	**RMSE**	**LRMSE**	$\delta = 1.25$	$\delta = 1.25^2$	$\delta = 1.25^3$	**M.**	**F.**
Zhou et al. [97]*	0.183	1.595	6.709	0.270	0.734	0.902	0.959		
Mahjourian et al. [49]	0.163	1.240	6.220	0.250	0.762	0.916	0.967		
Wang et al. [81]	0.151	1.257	5.583	0.228	0.810	0.936	0.974		
Yin and Shi [92]*	0.149	1.060	5.567	0.226	0.796	0.935	0.975		
Zou et al. [98]	0.150	1.124	5.507	0.223	0.806	0.933	0.973		
Almalioglu et al. [2]	0.150	1.141	5.448	0.216	0.808	0.939	0.975		
Zhou et al. [96] "LR"	0.143	1.104	**5.370**	0.219	**0.824**	0.937	0.975		
Ours	**0.138**	**1.030**	5.394	**0.216**	0.820	**0.941**	**0.977**		
Luo et al. [48]	0.141	1.029	5.350	0.216	0.816	0.941	0.976	✓	
Ranjan et al. [62]	0.140	1.070	5.326	0.217	0.826	0.941	0.975	✓	
Ours	0.138	1.030	5.394	0.216	0.820	0.941	0.977		
Casser et al. [7] "M"	0.141	1.026	5.290	0.215	0.816	0.945	**0.979**	✓	
Gordon et al. [23]	**0.128**	**0.959**	**5.230**	**0.212**	**0.845**	**0.947**	0.976	✓	
Ours	0.138	1.030	5.394	0.216	0.820	0.941	0.977		
Casser et al. [7] "M+R"	0.109	0.825	4.750	0.187	0.874	**0.958**	**0.983**	✓	✓
Chen et al. [9]	**0.099**	**0.796**	**4.743**	**0.186**	**0.884**	0.955	0.979	✓	✓

Figure 4.4 shows that depth maps predicted through our method can capture the structure of the scene. In addition, the last two images show that, when the scene is not rigid, our method is more prone to errors.

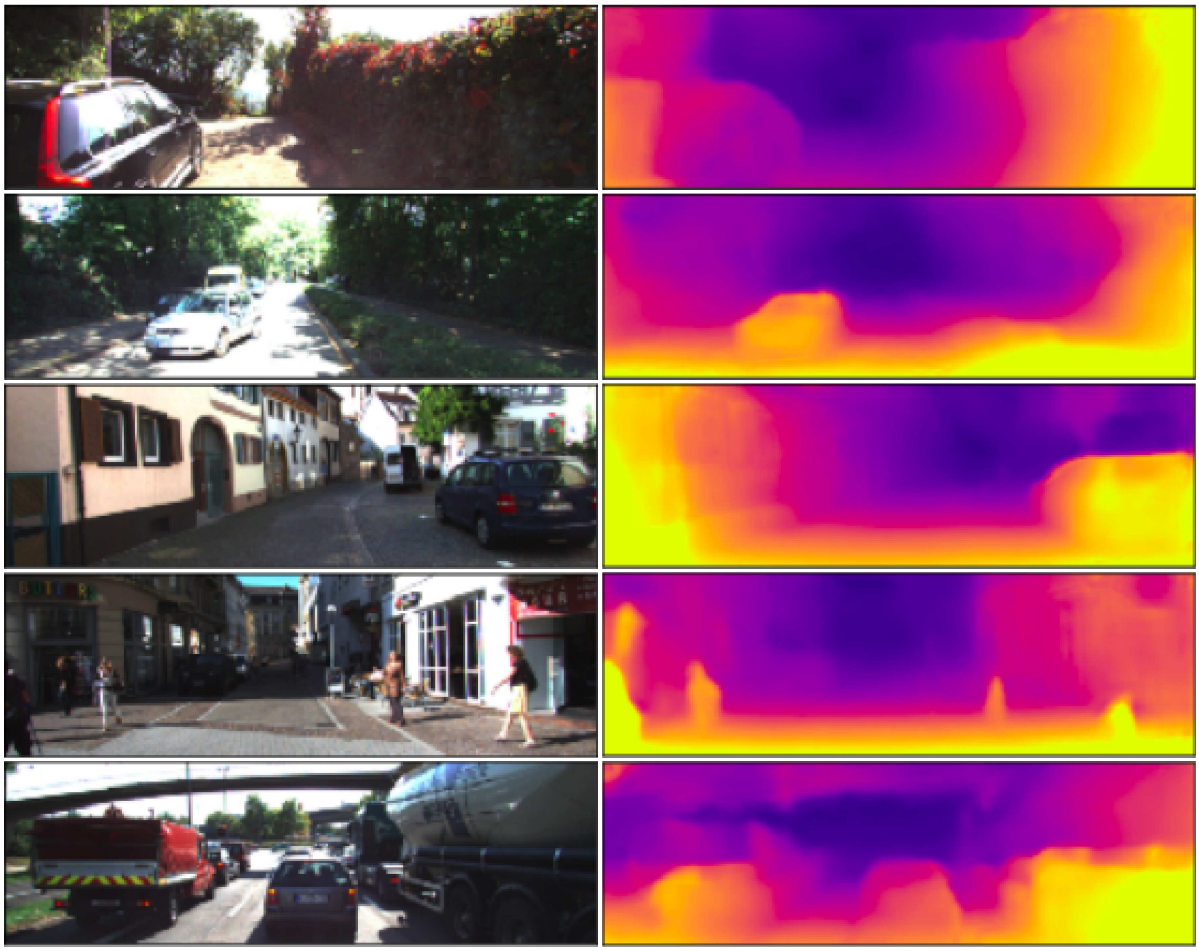


Figure 4.4: Input images and corresponding depth maps generated with our method. Images were sampled from the KITTI dataset.

4.4 Final Considerations

We proposed a self-supervised method for monocular depth estimation that relies on (i) a depth consistency constraint, (ii) a soft visibility map that reduces the error contribution in depth inconsistent regions, and (iii) sharing features from the depth to the camera motion networks.

We showed that the soft visibility mask and feature sharing mechanism can improve the performance of our baseline model. Our method achieves competitive results even with methods that explicitly model moving objects in the scene.

Chapter 5

Adaptive Self-Supervised Monocular Depth Estimation

One of the main challenges of self-supervised approaches based on reconstruction is that some pixels in frame cannot be explained from other frames because of occlusion, specular reflection, textureless regions among other reasons. Several approaches deal with these challenges excluding or attenuating the influence of pixels based on priors or adaptive approaches that leverage the availability of multiple frames neighboring a target frame to explain their pixels.

We develop and evaluate two adaptive strategies to improve the robustness of self-supervised depth estimation approaches with pixels that violate the assumptions of view reconstruction. Initially, we develop an adaptive consistency loss that extends the usage of minimum re-projection to enforce consistency on 3D structure and feature maps, in addition to the photometric consistency. Moreover, we evaluate the usage of uncertainty as loss attenuation mechanism, where the uncertainty is learned by modeling predictions as Laplacian, smooth-L1 or Cauchy probability distributions. Finally, we improve our model with a composite visibility mask. Our code is available at `https://github.com/jmendozais/SDSSDepth`.

5.1 Related Work

In this section, we briefly review some relevant methods available in the literature related to the topics addressed in our work.

5.1.1 Consistency Constraints

The availability of a correspondence between the pixels on the source and target views allows supervision by enforcing consistency on representations, in addition to the pixel intensities. For example, we can enforce consistency between forward and backward optical flows [92, 48], predicted and projected depth maps [21, 48], 3D coordinates [49], and feature maps [71, 93]. However, we cannot enforce consistency in the entire image because some regions do not have valid correspondences, for example, occluded regions produced by the motion of the camera or objects, or regions with specular reflection where the color

is inconsistent with the structure of the scene, and also due to multiple correspondences for single pixels at homogeneous regions do not provide supervision.

Techniques that exclude or attenuate the error contribution of these regions have been proposed in the literature. For example, learning an explainability mask [97], excluding pixels that are projected out of the field of view [49], excluding pixels with high inconsistencies on optical flows or depth maps [92], excluding stationary pixels [22], excluding occluded pixels using to geometric cues [23], attenuating the error using similar criteria [51].

Another approach leverages the availability of correspondences from multiple source frames [22] or estimated from different models [9], considering only the correspondences with minimum photometric error.

Our strategy extends the minimum re-projection error on other consistency constraints in addition to photometric consistency.

5.1.2 Adaptive Losses based on Uncertainty

The importance of quantifying the uncertainty on predictions has motivated research endeavors in several problems on computer vision, such as robust regression [4], representation learning [84], object detection [26], image de-raining [91], optical flow [32] and depth estimation [40, 78, 90, 60].

Researchers have explored approaches that leverage uncertainty information for depth estimation, for instance, a method that leverages existing uncertainty estimation techniques [78] and an approach that predicts the uncertainty using a neural network [40, 60, 90]. A recent work explored approaches to estimate epistemic uncertainty and aleatoric uncertainty on an unsupervised monocular setting [60].

In this work, we explore several probability functions to predict aleatoric uncertainty to improve depth estimation.

5.2 Method

Figure 5.1 illustrates the main components of our method. In this section, we provide an overview of our baseline system. Moreover, we introduce two adaptive strategies to improve the robustness of our approach. Finally, we explore additional constraints.

5.2.1 Preliminaries

Approaches that use view reconstruction as main supervisory signal require to find correspondences between pixel coordinates on frames that represent views of the same scene. These correspondences can be computed using multi-view geometry.

Given a pixel coordinate x_t in a target frame $\mathbf{I_t}$, we can obtain its coordinate x_s in a source frame $\mathbf{I_s}$ by back-projecting x_t to the camera coordinate system of the $\mathbf{I_t}$ using its depth value $\mathbf{D_t}(x_t)$, and the inverse of its intrinsic matrix $\mathbf{K}^{-1}$. Then, the relative motion transformation $\mathbf{T_{t\to s}}$ is applied to project the coordinates form the coordinate system of the $\mathbf{I_t}$ to the coordinate system of $\mathbf{I_s}$. Finally, the coordinates are projected onto the

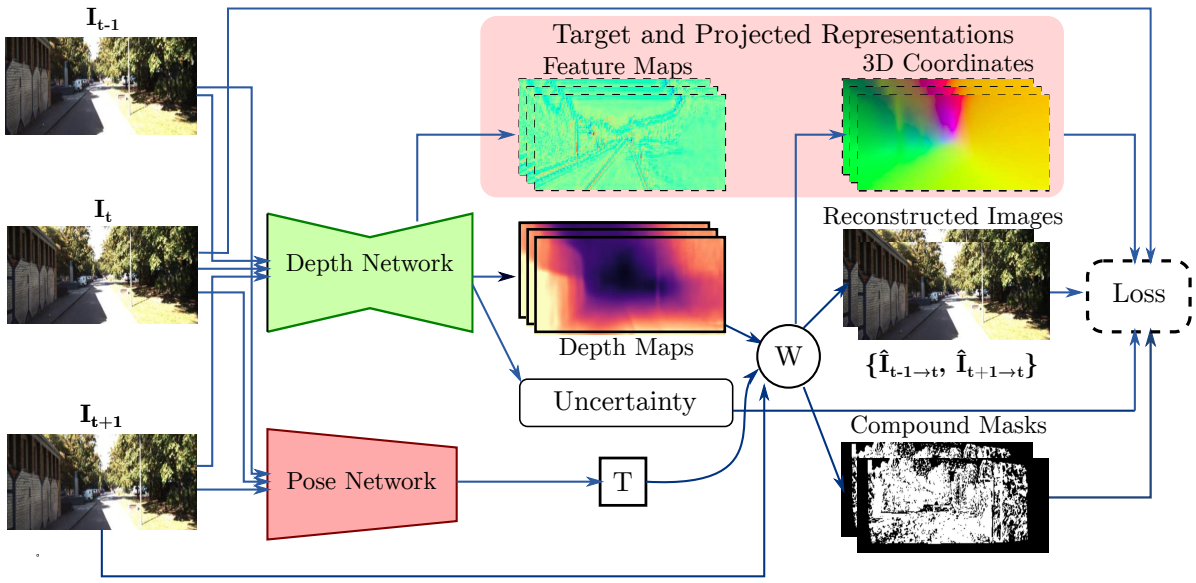


Figure 5.1: Overview of our method. The depth network is used to predict the depth maps for the target $\mathbf{I_t}$ and source images $\mathbf{I_s} \in \{\mathbf{I_{t-1}}, \mathbf{I_{t+1}}\}$. The pose network predicts the Euclidean transformation between the target and source camera coordinate systems $\mathbf{T_{t\to s}}$.

image plane in the source frame. We express this correspondence in Equation 5.1. We refer the reader to [97] for a detailed explanation.

$$x_s \sim \mathbf{K}\mathbf{T}_{\mathbf{t}\to\mathbf{s}}\mathbf{D_t}(x_t)\mathbf{K}^{-1}x_t \tag{5.1}$$

Once we know the projected coordinates and, therefore, the pixel intensities in the source image plane for each pixel x_t in the target image, we reconstruct the target frame $\hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}}(x_t) = \mathbf{I_s}(x_s)$. This process is known as image warping. This approach requires the dense depth map $\mathbf{D_t}$ of the target image, which we aim to reconstruct, the Euclidean transformation $\mathbf{T_{t\to s}}$, and camera intrinsics $\mathbf{K}$.

Our model predicts the depth maps and the Euclidean transformation using convolutional neural networks and assumes that the camera intrinsics are given. The networks are trained using the reconstruction error as supervisory signal.

5.2.2 Adaptive Consistency Loss

Consistency could be enforced on representations of the scene such as 3D structure and feature maps. We propose an *adaptive consistency loss* that, in addition to the photometric consistency, also considers 3D structure and feature consistency constraints. This idea leverages the robustness of the min-reprojection error to pixels with high reconstruction error that could potentially be outliers. The adaptive consistency loss is defined as

follows:

$$\mathcal{L}_{ac} = \sum_{x_t \in \mathbf{I_t}} \min_{\mathbf{I_s}} \Bigg(\mathbf{M_o}(x_t) \Big(\rho_{pc}\big(\mathbf{I_t}(x_t), \hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}}(x_s)\big) + \lambda_{sc}\rho_{sc}\big(\mathbf{C_{t\to s}}(x_t), \hat{\mathbf{C}}_{\mathbf{s}\to\mathbf{t}}(x_s)\big) + \lambda_{fc}\rho_{fc}\big(\mathbf{F_t}(x_t), \hat{\mathbf{F}}_{\mathbf{s}\to\mathbf{t}}(x_s)\big)\Big)\Bigg) \tag{5.2}$$

where ρ_{pc} measures the photometric consistency between pixels on the original $\mathbf{I_t}$ and reconstructed images $\hat{\mathbf{I}}_{\mathbf{s}\to\mathbf{t}}$, ρ_{sc} measures the structure consistency between the 3D-coordinates of the target image projected to the camera coordinate system of the source image $\mathbf{C_{t\to s}}$, and the 3D-coordinates of the source image warped to the target frame $\hat{\mathbf{C}}_{\mathbf{s}\to\mathbf{t}}$, and ρ_{fc} measure the feature dissimilarity between the feature vectors for all pixels, and obtained from the target $\mathbf{F_t}$, and the source feature maps warped to the target frame $\hat{\mathbf{F}}_{\mathbf{s}\to\mathbf{t}}$. The feature maps are extracted from the decoder part of the depth network. $\mathbf{M_o}$ is a visibility mask that excludes pixels that lie out the field-of-view on the source frame [49].

Our photometric error function ρ_{pc} is a combination of an $L1$ distance and the structure similarity index metric (SSIM) [83], with a trade-off parameter α. This function is shown in Equation 5.3.

$$\rho_{pc}(p,q) = \alpha\frac{1-\text{SSIM}(p,q)}{2} + (1-\alpha)||p-q||_1 \tag{5.3}$$

where p and q represent the colors of two corresponding pixels.

Our structure error function ρ_{sc} is the average of a normalized absolute difference of 3D-coordinates as follows:

$$\rho_{sc}(x,y) = \frac{1}{3}\sum_{i=1}^{3}\frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{5.4}$$

where x and y represent the 3D-coordinates of two corresponding pixels, and i represents a dimension of a 3D-coordinate.

Our feature dissimilarity function ρ_{sc} measures the squared L_2 distance of the L_2 normalized feature vectors $\hat{f}_s = f_s/||f_s||_2$ and $\hat{f}_t = f_t/||f_t||_2$, with $f_s = \hat{\mathbf{F}}_{\mathbf{s}\to\mathbf{t}}(x_t)$ and $f_t = \mathbf{F_{t\to s}}(x_t)$.

$$\rho_{fc}(f_s, f_t) = ||\hat{f}_s - \hat{f}_t||_2^2 \tag{5.5}$$

The total loss is the sum of the adaptive consistency loss and depth smoothness loss term [21] for the defined output scales $\mathcal{S}$.

$$\mathcal{L}_{total} = \sum_{i\in\mathcal{S}} \mathcal{L}_{ac}^{(i)} + \mathcal{L}_{ds}^{(i)} \tag{5.6}$$

5.2.3 Error Weighting Using Uncertainty

The adaptive consistency loss can handle cases in which at least one the source images can provide the information to reconstruct each pixel. However, several cases might break this condition, for instance, homogeneous regions and regions with specular reflection.

Therefore, we aim to find other mechanisms to handle pixels with large error on these cases.

An approach is to allow the model to learn the uncertainty about the depth estimates, and leverage this information to attenuate the effect of pixels with large errors on the overall error. We can do that by placing a probability distribution function over the outputs of the model. The predicted depth values $\mathbf{D_t}(x_t)$ are modeled as corrupted with additive random noise sampled from a PDF with a scale parameter σ_{x_t} that is predicted by depth depth network. σ_{x_t} quantifies the uncertainty of the model on the predictions. The model is trained to minimize the negative log-likelihood.

First, we assume that noise comes from a Laplacian distribution, then the error function is the negative log-likelihood of this distribution. Equation 5.7 shows the error function.

$$\rho_{Laplacian}(p_t, p_s) = \frac{|\rho_{pc}(p_t, p_s)|}{\sigma_{x_t}} + \log(2\sigma_{x_t}) \tag{5.7}$$

where $p_t = \mathbf{I_t}(\mathbf{x_t})$, $p_s = \hat{\mathbf{I}}_{\mathbf{s}\rightarrow\mathbf{t}}(x_s)$, ρ_{pc} is the photometric error function, and σ_{x_t} is the predicted uncertainty for the pixel x_t.

We can observe that the first term in Equation 5.7 attenuates the error when the uncertainty is high. Then, the second term discourages the model to predict high uncertainty values for all pixels. Thus, in order to minimize the function, the model is encouraged to predict high uncertainty values for pixels with large errors, attenuating the influence of large error in the overall error.

In order to explore the space of probability functions, we also evaluate our approach on the smooth-L1 functions and the Cauchy functions [4]. We define the probability distribution associated to the smooth-L1 function using the family of probability distributions defined in [4]. Equation 5.8 shows the negative log-likelihood associated to the smooth-L1 function.

$$\rho_{smooth\text{-}L1}(p_t, p_s) = \sqrt{\left(\frac{\rho_{pc}(p_t, p_s)}{\sigma_{x_t}}\right)^2 + 1} - 1 + \log(Z(1)) \tag{5.8}$$

where $Z(1)$ is a normalization factor for smooth-L1 function. We refer the reader to [4] for a detailed explanation.

Finally, Equation 5.9 shows the negative log-likelihood associated with the Cauchy distribution.

$$\rho_{Cauchy}(p_t, p_s) = \log\left(\frac{1}{2}\left(\frac{\rho_{pc}(p_t, p_s)}{\sigma_{x_t}}\right)^2 + 1\right) + \log(\sqrt{2}\pi\sigma_{x_t}) \tag{5.9}$$

Similarly, we propose to attenuate the error contribution in the scale of the images. This is a single uncertainty σ_t is predicted by each image. In the training process, the uncertainty is optimized to match to the distribution of errors for all the pixels of each image.

5.2.4 Exploring Visibility Masks

We combine several strategies to filter out pixels that are likely to be outliers. We mask the pixels on the target image that lie out of the field-of-view on the source image, also known as principled mask [92], the pixels that belong to homogeneous regions and do not change their appearance, even when the camera is moving [22], and the target pixels that are occluded in the source view [23]. The resulting composite mask is applied to our adaptive consistency loss at each scale.

5.2.5 Implementation Details

The depth network is a convolutional encoder-decoder network with skip connections. We used a ResNet18 as backbone for the encoder part of the depth network. The decoder network is composed of deconvolutional layers that up-sample the bottleneck representation in order to upscale the feature maps to the input resolution. For uncertainty estimation, we add a channel on the output of the depth network. In order to predict uncertainty pixel values, the extra channel is used as uncertainty map. On the other hand, when we aim to predict a single uncertainty value for image, we use spatial average pooling over the uncertainty map.

The motion network predicts the relative motion between two input frames. The relative camera motion has a 6-DoF representation that corresponds to 3 rotation angles and the translation vector. The motion network is composed of the first five layers of the ResNet18 architecture, followed by a spatial average pooling and four 1×1 convolutional layers.

5.3 Experiments

In this section, we show the experiments conducted to evaluate each component of the proposed system separately, as well the complete system with the proposed components.

5.3.1 Experimental Setup

Next, we describe the dataset used to train and evaluate the models, as well as the parameters of our model and the optimization method used to train the proposed method.

Dataset

We use the KITTI benchmark [20], described in Section 3.1. We used the Eigen split [14] with 45,023 images for training and 687 for testing. Moreover, we partitioned the training set on 40441 for training, 4,582 for validation. For result evaluation, we used the standard metrics [14].

Training

Our networks are trained using ADAM optimization algorithm with a learning rate of $2e-5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. We used the batch size of 12 snippets. Each

snippet is a 3-frame sequence. The frames are resized to a resolution of 416×128 pixels.

5.3.2 Adaptive Consistency Loss

In Table 5.1, we show the performance of the baseline model improved by considering the spatial and feature consistency loss terms individually, as well as combined using average and minimum re-projection. In the first row, we present the results of our baseline model that only considers the photometric consistency and depth smoothness loss terms. In the following rows, we compare the performance of the model including structure and feature consistency terms individually and jointly by using average or minimum re-projection.

As other works in the literature [21, 48, 49, 71, 93], we show that including structure and feature consistency terms is beneficial. The results indicate that our implementations of structure and feature consistency can improve the performance of the model individually, in most of the metrics. Furthermore, our results indicate that both terms are complementary and, together, can improve the performance with average and minimum re-projection losses. We obtained better results with minimum re-projection error.

Table 5.1: Ablation study on the adaptive consistency loss. We evaluate the performance of structure and feature consistency terms with average re-projection, and the adaptive consistency loss, which uses minimum re-projection error.

				↓ **Lower is better**				↑ **Higher is better**		
Avg.	**Min.**	**SC**	**FC**	**Abs Rel**	**Sq Rel**	**RMSE**	**Log RMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
				0.1116	0.8905	4.7177	0.1840	0.8717	0.9564	0.9817
✓		✓		0.1113	1.0024	4.6312	0.1807	0.8797	0.9595	0.9823
✓			✓	0.1104	<u>0.8747</u>	4.6005	0.1800	0.8785	0.9587	0.9825
✓		✓	✓	<u>0.1096</u>	1.0134	<u>4.5476</u>	<u>0.1776</u>	**0.8838**	<u>0.9611</u>	<u>0.9828</u>
	✓	✓	✓	**0.1059**	**0.7520**	**4.4537**	**0.1737**	<u>0.8834</u>	**0.9620**	**0.9848**

5.3.3 Error Weighting Using Uncertainty

We evaluate the usage of uncertainty to weigh the error contribution when the uncertainty values are predicted by pixel and by image.

Error weighting by pixel

In Table 5.2, we show that predicting uncertainty to weight the error contribution by pixel improves the performance of the baseline model using smooth-L1 probability function. However, the variants of the model that use the Laplacian and Cauchy distribution degrade the results.

We observe that the model predicts incorrect depth values on regions where the pixel intensities vary. This variation occurs because the predictive uncertainty is formulated on the photo-metric consistency term (Equation 5.7).

Table 5.2: Using uncertainty to weigh the error contribution by pixel.

Method	↓ Lower is better				↑ Higher is better		
	Abs Rel	**Sq Rel**	**RMSE**	**Log RMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline-L1	0.1894	4.1497	5.9739	0.2433	**0.8111**	0.9228	0.9599
Laplacian	0.1987	4.4033	6.0675	0.2481	0.8034	0.9216	0.9593
Smooth-L1	**0.1810**	**3.0795**	**5.6726**	**0.2386**	0.8027	**0.9245**	**0.9634**
Cauchy	0.1968	3.2513	5.9439	0.2540	0.7836	0.9147	0.9565

Error weighting by image

In Table 5.3, we show the effect of predictive uncertainty by image to weight the error contribution of images using the Laplacian, Smooth-L1, and Cauchy probability functions. The first row shows our baseline, which use an L1 distance between pixel intensities to measure photometric consistency and depth smoothness.

Our results indicate that the Smooth-L1 function improves the performance of the baseline and outperforms the approaches that assume other distributions. However, using uncertainties predicted through Laplacian and Cauchy functions does not improve the performance. Qualitative results are illustrated in Figure 5.2.

Table 5.3: Using uncertainty to weigh the error contribution by image.

Method	↓ Lower is better				↑ Higher is better		
	Abs Rel	**Sq Rel**	**RMSE**	**Log RMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline-L1	0.1894	4.1497	5.9739	0.2433	0.8111	0.9228	0.9599
Laplacian	0.1928	4.4074	5.9921	0.2472	**0.8153**	0.9234	0.9598
Smooth-L1	**0.1561**	**1.3712**	**5.3931**	**0.2239**	0.8018	**0.9286**	**0.9683**
Cauchy	0.1976	3.3892	6.0628	0.2530	0.7846	0.9160	0.9600

5.3.4 Visibility Masks

We performed ablation studies with visibility masks to filter out inconsistent pixels. We used model trained with the adaptive consistency loss as baseline. In Table 5.4, we show that every mask improves the error metrics, as well as the thresholded accuracy metrics. Moreover, the model trained with all mask formulations achieved better results. Qualitative results are illustrated in Figure 5.3.

5.3.5 Comparison with the State of the Art

In Table 5.5, we show that our method achieved competitive results when compared to state-of-the-art methods. Moreover, our approach is compatible and it could be improved with advanced strategies such as inference-time refinement [9, 7], joint depth and optical flow estimation [9], and effective architecture designs [25].

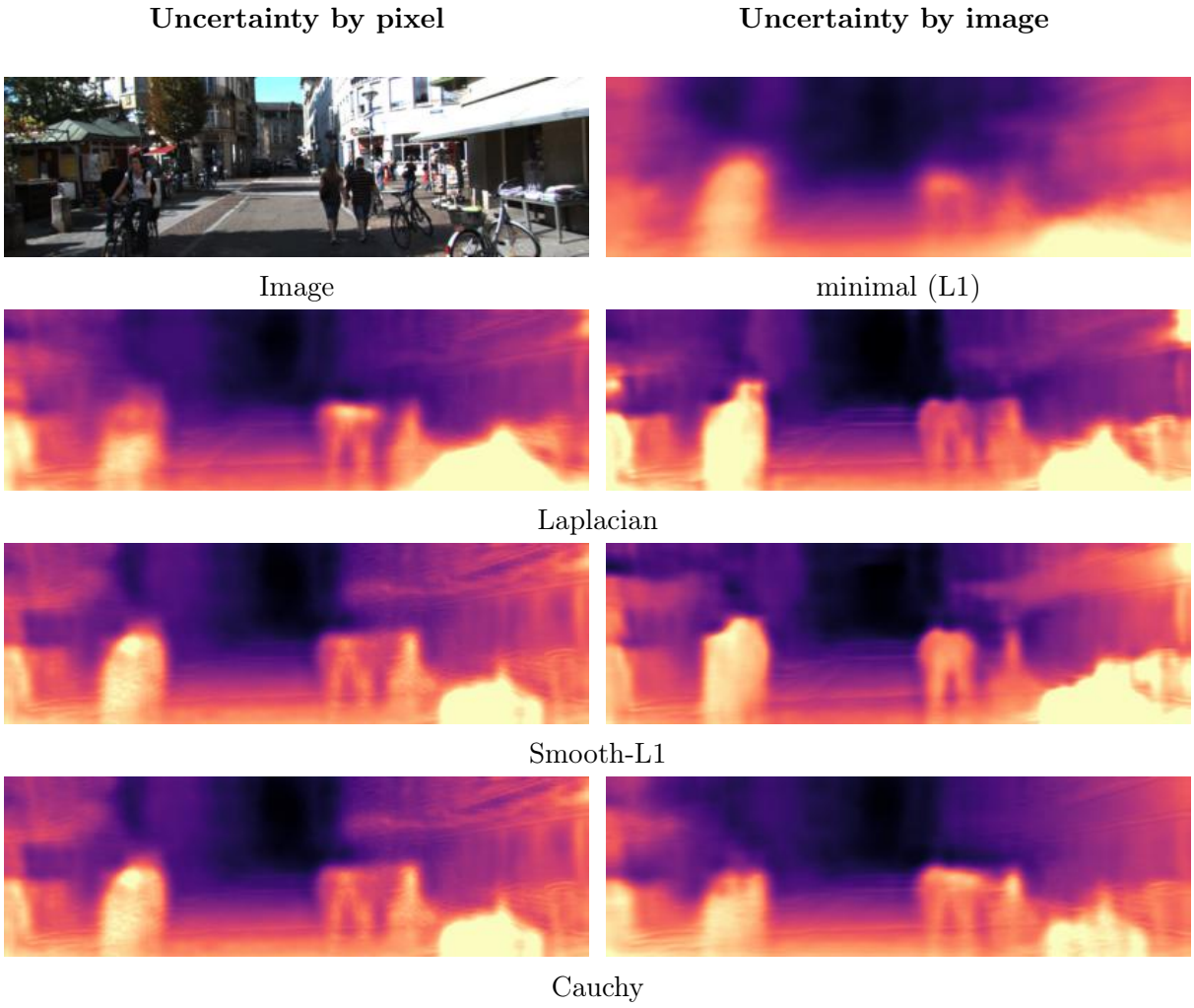


Figure 5.2: Qualitative results of error weighting approach with uncertainty. The first row shows a target image and its depth maps predicted with the minimal model. The remaining rows compare the results for the error weighting approaches for the PDF associated to Laplacian, Smooth-L1 and Cauchy functions. For each row, we present the result of method considering an uncertainty value by pixel on the left and and the result considering an uncertainty value by image respectively on the right.

Table 5.4: Ablation study of additional masks. We considered the Field-of-View masks (FOV), Auto mask (AM), Geometric mask(GM).

			↓ Lower is better				↑ Higher is better		
FOV	**AM**	**GM**	**Abs Rel**	**Sq Rel**	**RMSE**	**Log RMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
✓			<u>0.1059</u>	**0.7520**	4.4537	0.1737	0.8834	0.9620	**0.9848**
✓	✓		0.1063	0.8071	4.5570	0.1779	0.8829	0.9612	0.9831
✓		✓	0.1073	0.9355	**4.4135**	<u>0.1734</u>	<u>0.8877</u>	**0.9629**	<u>0.9840</u>
✓	✓	✓	**0.1015**	<u>0.7692</u>	<u>4.4297</u>	**0.1719**	**0.8890**	<u>0.9622</u>	0.9839

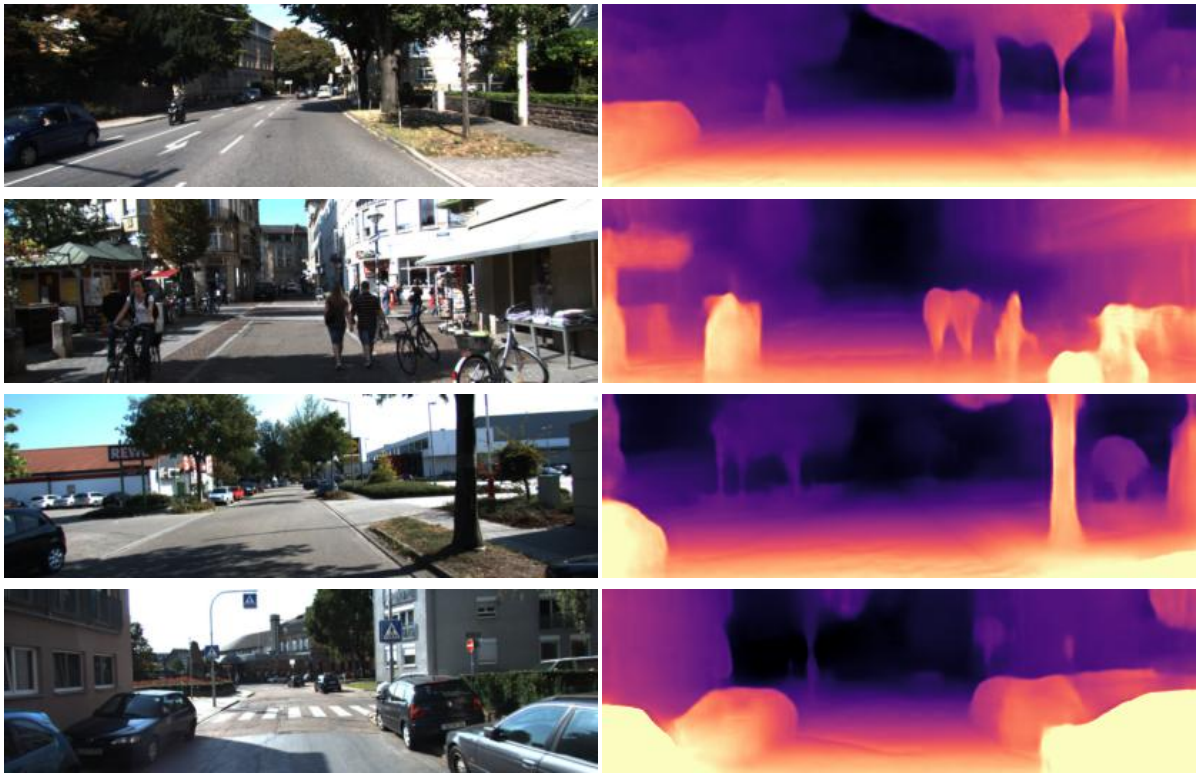


Figure 5.3: Qualitative results. Depth prediction using our final model.

Table 5.5: Results of depth estimation on the Eigen split of the KITTI dataset. We compared our results against several methods of the literature. In order to allow a fair comparison, we report the results of competitive methods trained with a resolution of 416×128 pixels. (*) indicates newly results obtained from an official repository. (-ref.) indicates that the online refinement component is disabled.

Method	**↓ Lower is better**				**↑ Higher is better**		
	Abs Rel	**Sq Rel**	**RMSE**	**Log RMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou et al. [97]*	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Mahjourian et al. [49]	0.163	1.240	6.220	0.250	0.762	0.916	0.967
Ying et al. [92]*	0.149	1.060	5.567	0.226	0.796	0.935	0.975
Casser et al. [7] (-ref.)	0.141	<u>1.026</u>	5.290	0.215	0.816	0.945	<u>0.979</u>
Chen et al. [9] (-ref.)	0.135	1.070	5.230	<u>0.210</u>	0.841	0.948	**0.980**
Gordon et al. [23]	<u>0.129</u>	**0.959**	5.230	0.213	0.840	0.945	0.976
Ours	0.131	1.037	<u>5.173</u>	**0.204**	<u>0.846</u>	<u>0.952</u>	**0.980**
Godard et al. [22]	**0.128**	1.087	**5.171**	**0.204**	**0.855**	**0.953**	0.978

5.4 Final Considerations

In this work, we show that minimum re-projection can be used to jointly enforce consistency on photometric, 3D structure, and feature representations of frames. This approach reduces the influence of pixels without valid correspondences on other consistency constraints, in addition to photometric consistency.

Moreover, our results suggest that the error weighting approaches based on predictive

uncertainty at pixel and image levels can be beneficial when the model is minimal, when the model does not implement additional strategies to handle invalid correspondences and when the outputs are assumed to follow the probability distribution derived from the smooth-L1 function. Further exploration could be done to leverage uncertainty to improve the performance of self-supervised depth estimation methods that consider several priors to handle invalid correspondences.

Chapter 6

Self-Distilled Self-Supervised Monocular Depth Estimation

Several works have shown that self-supervised depth estimation can be benefited from learning additional auxiliary tasks, for example, self-distillation. Self-distillation methods aim to improve a model performance by distilling knowledge from the model itself. An interesting approach to perform self-distillation consists of extracting information from distorted versions of the input data [87]. This is accomplished by enforcing consistency between predictions from distorted versions of the same input.

In this chapter, we propose a self-distillation approach via prediction consistency to improve self-supervised depth estimation from monocular videos. Since enforcing consistency between predictions that are unreliable cannot provide useful knowledge, we propose a strategy to filter out unreliable predictions.

Moreover, the idea of enforcing consistency between predictions has been widely explored in self-distillation [55, 17, 88, 95, 39] and semi-supervised learning [67, 77, 33, 3, 85, 73]. In order to explore the space of consistency enforcement strategies, we adapt and evaluate representative approaches on the self-supervised depth estimation task.

In summary, the main contributions of our solution are the following: (i) the proposition of a multi-scale self-distillation method based on prediction consistency, (ii) the design of an approach to filter unreliable per-pixel predictions on the pseudo-labels used in self-distillation, and (iii) the exploration and adaptation of several consistency enforcement strategies for self-distillation.

To validate our method, we show a detailed evaluation and a comparison against state-of-the-art methods on the KITTI benchmark. Our code is available at `https://github.com/jmendozais/SDSSDepth`.

6.1 Related Work

In this section, we briefly review some relevant approaches available in the literature related to the topics explored in our work.

6.1.1 Pseudo-Labeling Approaches for Self-Supervised Depth Estimation

Many self-supervised methods trained from stereo images [78, 86] or monocular sequences [78, 11, 10, 38, 45] rely on pseudo-labels to provide additional supervision for training their depth networks. These methods can use state-of-the-art classical stereo matching algorithms [78], external deep learning methods [11, 10], or their own predictions [86] to obtain pseudo-labels.

Since the quality of the pseudo-labels is not always guaranteed, methods filter out unreliable per-pixel predictions based on external confidence estimates [78, 11, 10] or uncertainty estimates that are a result of the method itself [86].

Additionally, some methods leverage multi-scale predictions for creating pseudo-labels by using the predictions at the highest-resolution as pseudo-labels to supervise predictions at lower resolutions [89] or by selecting, per pixel, the prediction with the lowest reconstruction error among the multi-scale predictions [57].

We focus on methods that use their own prediction as pseudo-labels. For example, Kaushik et al. [38] augmented a self-supervised method by performing a second forward pass with strongly perturbed inputs. The predictions from the second pass are supervised with predictions of the first pass. Liu et al. [45] proposed to leverage the observation depth maps predicted from day-time images are more accurate than predictions from night-time images. They used predictions from day-time images as pseudo-labels and train a specialized network with night-time images synthesized using a conditional generative model.

6.1.2 Self-Distillation

These methods let the target model leverage information from itself to improve its performance. An approach is to transfer knowledge from an instance of the model, previously trained, via predictions [55, 17, 88, 95, 39] and/or features to a new instance of the model. This procedure could be repeated iteratively. Self-distillation has a regularization effect on neural networks. It was shown that, at earlier iterations, self-distillation reduces overfitting and increases test accuracy, however, after too many iterations, the test accuracy declines and the model underfits [55].

Self-distillation has been extensively explored, mainly in image classification problems. An approach performs distillation by training instances of a model sequentially such that a model trained on a previous iteration is used as a teacher for the model trained in the current iteration [17]. Similarly, Yang et al. [88] proposed to train a model in a single training generation imitating multiple training generations using a cyclic learning rate scheduler and using the snapshots obtained at the end of the previous learning rate cycle as a teacher.

Our work explores the idea of leveraging multiple snapshots in a single training generation on the self-supervised depth estimation problem.

6.1.3 Consistency Regularization

Enforcing consistency between predictions obtained from perturbed views of input examples is one of the main principles behind consistency regularization approaches on deep semi-supervised works.

An early method [67] used this principle doing several forward passes on perturbed versions of the input data. Furthermore, other methods showed that the usage of advanced and strong data augmentation perturbations [85] or a combination of a weak and strong data augmentation perturbation in a teacher-student training scheme [73] can be helpful to improve the resulting models.

Existing works showed that average models, i.e., models whose weights are the average of the model being trained at different training steps, can be more accurate [77, 33, 3]. Average models can be used, as teachers, to obtain more accurate pseudo-labels [77, 3]. Moreover, the use of cyclic learning rate schedulers can improve the quality of the models that are averaged and the resulting model at accuracy and generalization [33], as well as it can be adapted to the consistency regularization framework [3].

Similarly to the method developed by Athiwaratkun et al. [3], our method uses a cyclic cosine annealing learning rate schedule to obtain a better teacher model.

6.2 Proposed Method

In this section, we present a method to perform self-distillation via prediction consistency. First, we describe the core idea of our method. Then, we introduce a mechanism to filter-out unreliable per-pixel depth prediction. Finally, we detail several prediction consistency enforcement strategies.

6.2.1 Self-Distillation via Prediction Consistency

The core idea of self-distillation based on prediction consistency is to provide additional supervision to the model by enforcing consistency between the depth map predictions obtained from different perturbed views of an input image.

Our self-distillation approach applies two different data augmentation perturbations to an input snippet. To use less computational resources, we use snippets of two frames $\mathcal{I} = \{\mathbf{I_t}, \mathbf{I_{t+1}}\}$. The model predicts the depth maps for all images in the input snippet. Since we need to apply two data augmentation perturbations, we have two depth maps for each frame in the snippet. Then, we enforce consistency between predictions by minimizing the difference between the predicted depth maps for each frame.

There are several approaches to enforce consistency between prediction. The simplest variation of our method use the pseudo-label approach. It considers one of depth maps as pseudo-label $\mathbf{D^{(pl)}}$, which implies that gradients are not back-propagated through it, and the other depth map as prediction $\mathbf{D^{(pred)}}$. In Section 6.2.3, we improve our method considering other consistency enforcement strategies.

Moreover, we enforce prediction consistency using the mean squared error (MSE) as difference measure. In addition, we filter the unreliable depth values on the pseudo-label

using a composite mask. Equation 6.1 shows the self-distillation loss term for a snippet $\mathcal{I}$.

$$\mathcal{L}_{sd} = \frac{1}{|\mathcal{I}|} \sum_{\mathbf{I_k} \in \mathcal{I}} \frac{1}{|\mathbf{M_k^{(c)}}|} \sum_{x \in \Omega(\mathbf{I_k})} \mathbf{M_k^{(c)}}(x) \left(\mathbf{D_k^{(pl)}}(x) - \mathbf{D_k^{(pred)}}(x) \right)^2 \tag{6.1}$$

where $\mathbf{I_k}$ is a frame in the snippet and $\mathbf{M_k^{(c)}}$ is its composite mask, x is a pixel coordinate, $\Omega(\mathbf{I_k})$ is the set of pixel coordinates, and $\mathbf{D_k^{(pl)}}$ and $\mathbf{D_k^{(pred)}}$ represent the pseudo-label and predicted depth maps, respectively.

Since our model predicts the depth maps at multiple scales, we compute the self-distillation loss for each scale. We assume that the pseudo-label at the finest scale is more accurate than the pseudo-labels at coarser scales. Thus, we only use the finest pseudo-label. We upscale the predictions to the finest scale to match the pseudo-label scale. Finally, we compute the self-distillation loss for each scale. Figure 6.1 depicts our self-distillation approach.

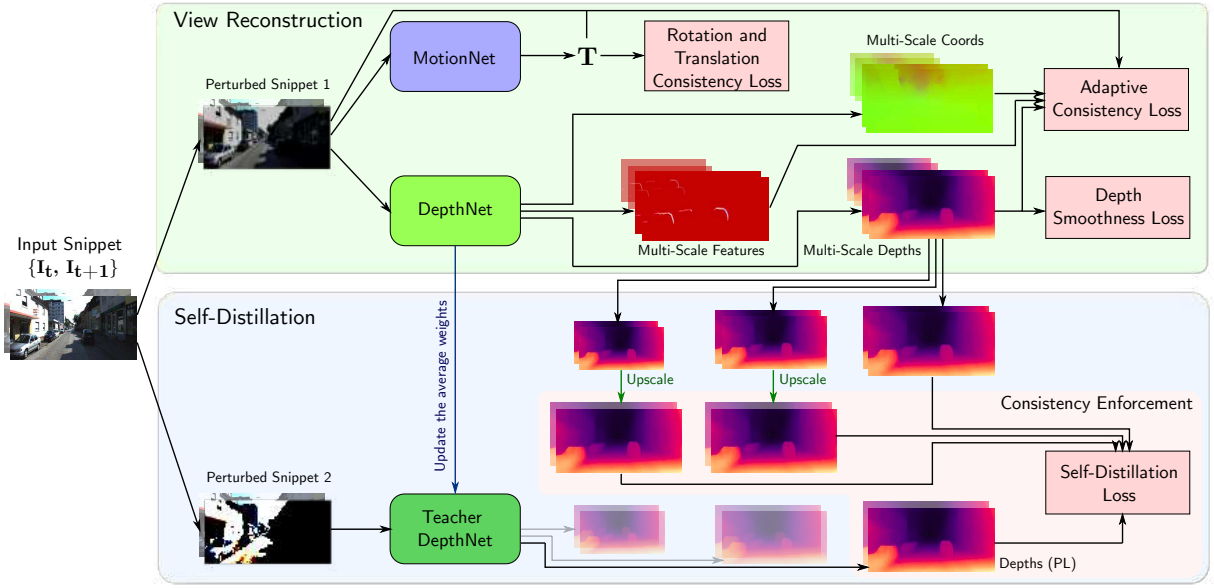


Figure 6.1: Overview of our method. The self-distillation component leverages the multi-scale predictions obtained from the view reconstruction component. The predictions are upscaled to the finest resolution. More accurate predictions are obtained from the teacher model. The teacher predictions at the finest resolution are used as pseudo-labels to improve the predictions obtained from view reconstruction.

6.2.2 Filtering Pseudo-Labels

We noticed empirically that unreliable depth prediction produces very large differences between pseudo-labels and predictions. These very large differences make training unstable and do not allow the model to converge randomly. We address this problem by excluding pixels with very large differences using a threshold value. In this section, we present two schemes to determine the threshold.

In the first scheme, we compute the threshold as a percentile P on the pseudo-label and prediction differences for all pixels in a batch of snippets. Then, we create valid mask

considering as valid all the pixels with differences smaller than the threshold, as shown in Equation 6.2.

$$\mathbf{M}^{(p)}(x) = \left[\left(\mathbf{D}^{(\mathbf{pl})}(x) - \mathbf{D}^{(\mathbf{pred})}(x)\right)^2 < P\right] \tag{6.2}$$

where [.] denotes the Iverson bracket operator. The final mask is obtained combining the latter mask with the compound mask. The final mask could be expressed as $\mathbf{M} = \mathbf{M}^{(p)} \odot \mathbf{M}^{(c)}$, where $\odot$ represents the element-wise product. Finally, we replace $\mathbf{M}^{(c)}$ with $\mathbf{M}$ in Equation 6.1.

We believe that the idea of using a threshold obtained from the distribution of differences by batch might be detrimental because we do not take into consideration that batches with reliable predictions should have thresholds that exclude less pixels than the threshold used on batches with more unreliable predictions.

In the second scheme, we address this limitation by approximating a global threshold $P^{(\text{EMA})}$ using the exponential moving average (EMA) of the percentile values for each batch during training. Another advantage of using a moving average is that we take into consideration that the distribution of depth differences change during training. This means that, when the depth differences become smaller during training, the threshold changes by increasing the weight of the percentiles from latter batches on the average. Equation 6.3 shows our global threshold approximation.

$$P_t^{(\text{EMA})} = P_{t-1}^{(\text{EMA})} \cdot \beta + P_t \cdot (1 - \beta) \tag{6.3}$$

where P_t is the threshold computed from the batch at the t training iteration, $P_t^{(\text{EMA})}$ is threshold obtained using the EMA at the t training iteration, and β controls the influence of the previous moving average percentile and the current percentile into the computation of the current threshold . Similarly to the first scheme, we compute a valid mask $\mathbf{M}^{(\text{EMA})}$using $P^{(\text{EMA})}$, we combine this mask with the compound mask $\mathbf{M} = \mathbf{M}^{(\text{EMA})} \odot \mathbf{M}^{(c)}$ and, finally, we use $\mathbf{M}$ instead of $\mathbf{M}^{(c)}$ in Equation 6.1.

6.2.3 Consistency Enforcement Strategies

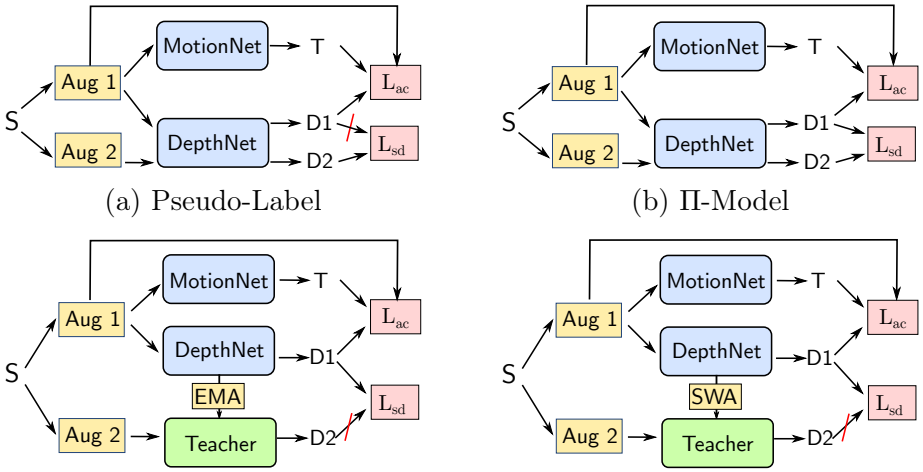
In previous sections, we described a pseudo-label strategy to enforce consistency between depth predictions. Here, we describe representative consistency enforcement strategies adapted to our self-distillation approach.

Figure 6.2 depicts these consistency enforcement strategies. We named each strategy similarly to the methods that introduced the key idea in the semi-supervised learning literature [67, 77, 33, 3].

Similarly to the pseudo-label strategy, variants of our method that use the strategies described in this section also adopt the second scheme described in Section 6.2.2 to filter out unreliable per-pixel predictions before computing the predictions difference.

Π-Model

Similarly to the pseudo-label approach, this strategy consists of enforcing consistency between depth predictions from two perturbed views of the same input. In contrast with the pseudo-label approach, the gradients are back-propagated through both predictions.



(a) Pseudo-Label (b) Π-Model

(c) Mean Teacher (d) Stochastic Weight Averaging

Figure 6.2: Simplified views of the consistency enforcement strategies. S denotes the the input snippet, *Aug 1* and *Aug 2* denote two perturbed views of the input snippet, T denotes the camera motion transformation, $D1$ and $D2$ denote depth maps predictions, L_{ac} denotes the adaptive consistency loss, L_{sd} denotes the self-distillation loss, and red lines — mark connections where the gradients are not back-propagated

Mean Teacher

Instead of using the same depth network to generate the pseudo-labels and the predictions, we can introduce a teacher network that can potentially predict more accurate pseudo-labels, and provide better supervisory signal to the model currently being trained, the student network. In this approach, the teacher depth network weights are the EMA of the depth network weights in equally spaced training iterations.

Stochastic Weight Averaging

Similarly to the mean teacher strategy, we set the teacher depth network weights as the EMA of the depth network weights. In contrast, the training process is split into several cycles. At each cycle, the learning rate decreases and the teacher depth network is updated with the weights of depth network at the end of the last epoch of each training cycle, where the learning rate reaches its lowest value.

In the first generation of the training process, we use the student network to predict pseudo-label. Once the model has converged to a proper local optimum, we use its weights to initialize the teacher network. Then, in the following cycles, the training process mimics multiple training generations using a cyclic cosine annealing learning rate. At the end of each cycle, when the learning rate reaches its lowest value, and model likely converged to a good local optimum, we update the weights the teacher network using EMA with the student network weights.

6.2.4 Additional Considerations

Final Loss. The overall loss is a weighted sum of our self-distillation loss $\mathcal{L}_{sd}$, adaptive consistency loss $\mathcal{L}_{ac}$ [52], depth smoothness loss $\mathcal{L}_{ds}$, translation consistency loss $\mathcal{L}_{tc}$, and

rotation consistency loss $\mathcal{L}_{rc}$. The rotation and translation consistency losses are similar to the cyclic consistency loss defined in [23]. In contrast, our translation consistency loss just considers camera motion. Equation 6.4 shows our final loss.

$$\mathcal{L} = \sum_{i \in \mathcal{S}} \frac{1}{2^i} \left(\mathcal{L}_{ac}^{(i)} + \lambda_{ds} \mathcal{L}_{ds}^{(i)} + \lambda_{sd} \mathcal{L}_{sd}^{(i)} \right) + \lambda_{rc} \mathcal{L}_{rc} + \lambda_{tc} \mathcal{L}_{tc} \tag{6.4}$$

where $\mathcal{S}$ is the set of scales and λ_{sd}, λ_{ds} λ_{rc}, λ_{tc} is the weight of the self-distillation, depth smoothness, rotation consistency, and translation consistency loss terms, respectively.

Network Architecture. We use similar depth and motion network architectures to those used in the method described in Chapter 5. For the depth network, we use a convolutional encoder-decoder network with skip connections. For the encoder, we use a ResNet18. For the decoder, we stack convolutional layers and up-sampling layers. Convolutional layers use the ELU activation function [12] for intermediate layers and sigmoid activation function for output layers.

In contrast to the depth network used in Chapter 5, in which the outputs are used as disparity prediction, here we use the outputs as depth prediction. We noticed that this change reduces artifacts with very high depth values on the outputs.

For the motion network, we also use a ResNet18 backbone for feature extraction with a modification to allow multi-frame inputs, that is, snippets. For the head of the motion network, we use 4 convolutional layers. In contrast with the motion network used in Chapter 5, we do not use global average pooling after feature extraction. We let the head of the motion network leverage the spatial information learned by the feature extractor. Finally, we use global average pooling in the last layer of the head part of the motion network.

6.3 Evaluation

The goal of our experimental evaluation is to answer the following questions: (1) Can a model trained with self-supervised loss based on view reconstruction further improve when a multi-scale self-distillation via prediction consistency loss term is considered?, (2) do our mechanisms to exclude the influence of unreliable per-pixel predictions enhance a model trained with self-supervised and self-distillations loss terms?, and (3) what are the most effective consistency enforcement strategies to use on our multi-scale self-distillation via prediction consistency loss term?

To answer the first question, we compare the performance of a competitive baseline with a variant of this baseline that includes our multi-scale self-distillation based on the prediction consistency method in Section 6.3.2. We answer the second question by comparing the performance of the simplest variant of our self-distillation approach with two variants that implement our per-pixel filtering schemes in Section 6.3.3. We answer the third question by comparing the performance of the variants considering the different consistency enforcement strategies in Section 6.3.4. Finally, we show qualitative results and perform a comparison with state-of-the-art methods evaluated in similar settings.

6.3.1 Experimental Setup

We describe here the dataset used to train and evaluate the models, as well as the parameters of our model and the optimization method used to train the main variants of the proposed method.

Dataset

We use the KITTI benchmark, described in Section 3.1. We used the Eigen split [14] with 45023 images for training and 687 for testing. Moreover, to search for hyper-parameters, we partitioned the training set on 40441 for training, 4582 for validation. For result evaluation, we used the standard metrics.

Training

Our networks are trained using ADAM optimization algorithm using $\beta_1 = 0.9$, $\beta_2 = 0.999$. We used the batch size of 4 snippets. We use 2-frame snippets unless otherwise specified. We resize the frames to resolutions of 416×128 pixels unless otherwise specified.

The training process has multiple stages. In the first stage, we train our models with the self-distillation loss disabled. We use a learning rate of $1e-4$ during 15 epochs, then it is reduced to $1e-5$ during 10 additional epochs.

We train all models that include a self-distillation term in a second stage. In this stage, models are trained with a learning rate of $1e-5$ during 10 epochs. Finally, we train the variants of our model that use teacher networks with average weights in a third stage.

For the mean teacher model, this stage lasts 10 epochs and the weights are updated every 1e3 iterations. For the SWA model, training is done using a cyclical cosine learning rate schedule with an upper bound of $1e-4$, a lower bound of $1e-5$, and using 4 cycles of 6 epochs each.

6.3.2 Self-Distillation via Prediction Consistency

We compared the performance of the simplest variant of our self-distillation method with the baseline in Table 6.1. Results show that our baseline has a competitive performance since it is trained with 2-frame snippets and obtains a performance similar to widely used baseline [22] that uses 3-frame snippets for training. The results show a consistent improvement when the self-distillation loss is used. The model trained with the self-distillation loss outperforms the baseline at all error metrics and almost all accuracy metrics.

When searching for the optimal weight λ_{sd} for the self-distillation term, we noticed that large λ_{sd} values allow to obtain good results. However, due to large depth differences, the model diverges on some executions. Due to this instability, we use a smaller $\lambda_{sd} = 1e2$. This observation motivated us to explore approaches to filter unreliable predictions.

Table 6.1: Comparison of the baseline model and the variation of our method that uses the pseudo-labeling strategy.

Method	↓ Lower is better				↑ Higher is better		
	Abs Rel	**Sq Rel**	**RMSE**	**LRMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	0.128	1.005	5.152	0.204	**0.848**	**0.951**	0.979
PL	**0.126**	**0.907**	**5.068**	**0.202**	0.847	**0.951**	**0.980**

6.3.3 Filtering Pseudo-Labels

In Table 6.2, we show that our two filtering strategies outperform that variation of our method does not use any additional filtering approach other than the composite mask in the majority of error and accuracy metrics. Moreover, the results show that the approach that uses the EMA of the percentiles to estimate the threshold is better than using only the percentile of each batch.

Table 6.2: Comparison of variants of our method with and without filtering strategies. P denotes that we filtered pseudo-labels using a percentile by batch as thresholds, and $P^{(\text{EMA})}$ denotes that we filtered pseudo-labels using a threshold that is the EMA computed from percentiles of the batches during training iterations.

Method	↓ Lower is better				↑ Higher is better		
	Abs Rel	**Sq Rel**	**RMSE**	**LRMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
PL (w/o filtering)	**0.126**	0.907	5.068	**0.202**	**0.847**	0.951	**0.980**
PL + P	**0.126**	0.911	5.033	0.203	**0.847**	**0.952**	**0.980**
PL + $P^{(\text{EMA})}$	**0.126**	**0.904**	**5.024**	**0.202**	**0.847**	**0.952**	**0.980**

6.3.4 Consistency Enforcement Strategies

In Table 6.3, our results indicate that, regardless the consistency enforcement strategy, self-distillation via prediction consistency can improve the performance of our baseline model. Moreover, the results show that the variant that uses SWA strategy outperforms the other consistency enforcement strategies in most of the error and accuracy metrics. This variant is used as our final model.

We show qualitative results in Figure 6.3. We can observe that the predicted depth maps are sharp on salient objects of the image. In addition, the bottom-right image shows that our final model does not predict a consistent depth map for a thin object with a variable background.

6.3.5 State-of-the-Art Comparison

In Table 6.4, we show a quantitative comparison with state-of-the-art methods. Our method outperforms methods that explicitly address moving objects such as [7, 9, 23].

Table 6.3: Comparison of the representative consistency enforcement strategies. PL denotes the pseudo-label, Π M denotes the Π-Model, MT denotes the mean teacher, and SWA denotes the stochastic weight averaging strategy.

Method	↓ Lower is better				↑ Higher is better		
	Abs Rel	**Sq Rel**	**RMSE**	**LRMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline	0.128	1.005	5.152	0.204	**0.848**	0.951	0.979
PL + $P^{(\text{EMA})}$	0.126	0.904	**5.024**	0.202	0.847	**0.952**	0.980
Π-M + $P^{(\text{EMA})}$	0.126	0.902	5.041	0.202	0.847	**0.952**	0.980
MT + $P^{(\text{EMA})}$	0.126	0.898	5.061	**0.201**	0.846	**0.952**	**0.981**
SWA + $P^{(\text{EMA})}$	**0.125**	**0.881**	5.056	0.202	**0.848**	**0.952**	0.980

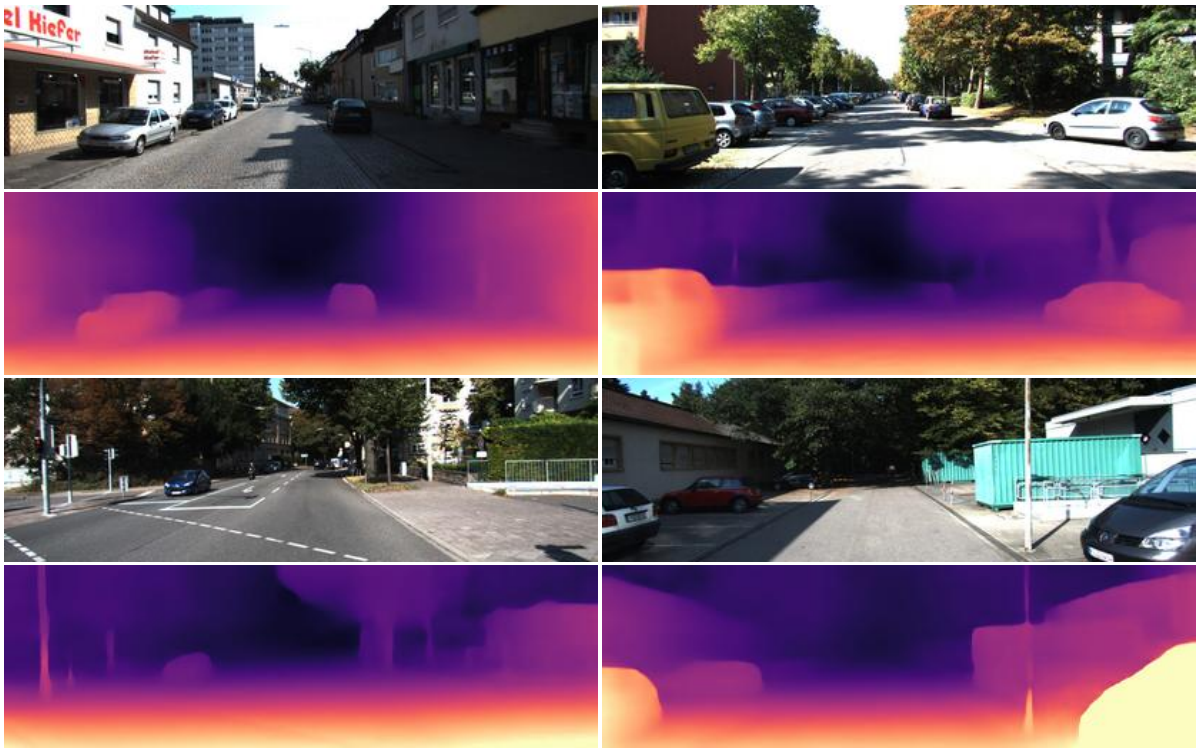


Figure 6.3: Qualitative results. Depths maps obtained using our final model.

The results show that our method achieves competitive performance when compared to state-of-the-art methods.

6.4 Final Considerations

We showed that to take full advantage of self-distillation in self-supervised depth estimation from monocular videos, we need to consider additional strategies. One strategy was filtering unreliable per-pixel predictions with threshold value.

Moreover, we demonstrated that choosing a proper consistency enforcement strategy in self-distillation is important. Our results suggest that the features of SWA consistency enforcement strategy, such as (i) enforcing teacher quality and (ii) enforcing difference be-

Table 6.4: Comparison with the state-of-the-art on the Eigen split of the KITTI dataset. We compared our results against several methods of the literature. To allow a fair comparison, we report the results of competitive methods trained with a resolution of 416×128 pixels. N.F. denotes the number of frames in the input snippet (*) indicates newly results obtained from an official repository. (-ref.) indicates that the online refinement component is disabled.

Method	**N.F.**	**↓ Lower is better**				**↑ Higher is better**		
		Abs Rel	**Sq Rel**	**RMSE**	**LRMSE**	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Gordon et al. [23]	2	0.129	0.959	5.230	0.213	0.840	0.945	0.976
Our method	2	**0.125**	**0.881**	**5.056**	0.202	**0.848**	**0.952**	**0.980**
Zhou et al. [97]*	3	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Mahjourian et al. [49]	3	0.163	1.240	6.220	0.250	0.762	0.916	0.967
Casser et al. [7] (-ref)	3	0.141	1.026	5.290	0.215	0.816	0.945	0.979
Chen et al. [9] (-ref)	3	0.135	1.070	5.230	0.210	0.841	0.948	0.980
Godard et al. [22]	3	0.128	1.087	5.171	0.204	0.855	0.953	0.978
Our method	3	0.123	**0.906**	5.083	0.200	0.856	0.953	0.980
Fang [15]	3	**0.116**	-	**4.850**	**0.192**	**0.871**	**0.959**	**0.982**

tween teacher and student network weights, are important to obtain larger improvements.

The improvements obtained with the variations of our method are consistent with the findings of recent works that also included self-distillation as a term of their loss function [45, 38]. In addition, the mechanisms implemented in our method are fully compatible with these works [45, 38], and could be easily combined.

Finally, we believe that our findings could provide useful insights to leverage self-distillation in methods that use stereo sequences as input, as well as semi-supervised and supervised methods.

Chapter 7

Conclusions and Future Work

In this thesis, we addressed the problem of single image depth estimation. We focus on self-supervised approaches that learn to predict dense depth maps using monocular sequences for training. First, we introduced the single image depth estimation problem, its challenges, and the main concepts related to the methods that we presented in this document.

Then, we proposed a method that shares the representations learned by from the depth model to provide additional cues to the camera motion model using lateral connections. Moreover, in this method we introduced a mechanism to deal the invalid correspondences, an heuristic that showed that diminishing the influence of invalid correspondences found our model can improve the the performance of our depth estimation model.

Later, we presented two adaptive strategies to deal with invalid correspondences. The first strategy shows that extending the minimum re-projection loss to 3D coordinates and deep feature representation can increase the performance on depth estimation. The second strategy shows an based aleatoric uncertainty can useful to diminish the influence of invalid correspondences on learning, however, its effect might not be complementary and its effect is not perceived when other strategies such as minimum re-projection or our composite visibility mask are used.

Moreover, we showed that self-distillation can provide an additional learning signal for self-supervised depth estimation. We showed that a per-pixel filtering strategy can help to deal with unreliable predictions. Finally, we demonstrated that choosing a proper consistency enforcement strategy in self-distillation is another important dimension to take the most of self-distillation.

During the development of this thesis, we focused on answering the research questions shown in the introductory chapter. We highlight reflections about the answers of the research questions in the next paragraphs.

Handling invalid correspondences. Results reported in Chapters 4 and 5 provided insights on approaches to mask out or to diminish the influence of pixels that have invalid correspondences. We observed that modeling explicitly the uncertainty of the model on the validity of pixel correspondences is critical to design effective methods. We noticed that mechanisms to diminish or mask out the influence of invalid pixel correspondences with heuristics based on geometric consistency or minimum re-projection, or with learning mechanisms such as loss attenuation based on aleatoric uncertainty can enhance self-

supervised single image depth estimation models.

Leveraging the relationships of features learned by depth and camera motion networks. Results reported in Chapter 4 demonstrated that strategies to take advantage of the feature representation obtained from neural networks in the self-supervised SIDE approach based on view reconstruction are valuable. We showed that deep feature representation learned by the depth network can be useful for the camera motion network to improve the view reconstruction and, as a consequence, the depth estimation performance.

Designing a loss to enforce consistency between feature representations learned by the neural networks. Results reported in Chapter 5 suggested that mechanisms considered in the design of photometric loss can also be helpful to enhance consistency losses based on other features. We showed that enforcing consistency between deep feature representations and 3D coordinate maps in addition to photometric information is useful when minimum re-projection is employed in the view reconstruction loss.

Additional learning signals to train networks for self-supervised SIDE. In Chapter 6, we proposed a self-distillation method to provide an additional learning signal to the self-supervised approach for SIDE. Similar to the learning signal provided by the view reconstruction auxiliary task, in which *consistency* of adjacent views is assumed, the consistency of prediction of perturbed input images in our self-distillation method provides the learning signal. Similar mechanisms used in view reconstruction loss, such as filtering and multi-scale processing, inspired the design of our self-distillation loss.

We believe that the proposition and evaluation of the mechanisms incorporated in the methods proposed in this thesis are an important contribution to the research community. Furthermore, we believe that future efforts related to the topics explored in this thesis might be of interest to the research community.

A future research direction, which is a direct extension of self-distillation via predictions consistency, is feature distillation. In the context of our problem, we could perform feature distillation by enforcing consistency between deep features obtained from the depth network.

We found several ideas in the feature distillation literature that we believe might lead to further improvements. A research direction would be to investigate transformations in the teacher features in order to leverage beneficial features and suppress adverse features from the teacher network, such as the approach proposed by Heo et al. [28], which uses a variation of the ReLU activation as a transformation.

Another future research direction would be to investigate improvements in the dissimilarity functions used to enforce consistency between teacher and student features. A comparison of simpler dissimilarity functions, for instance, L1, L2 and cosine similarity with more complex methods, such as methods that use adversarial learning to enforce similarity between deep features using discriminators [70], might be of interest to the research community.

Bibliography

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8M: A Large-Scale Video Classification Benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Y. Almalioglu, M. R. U. Saputra, P. P. de Gusmao, A. Markham, and N. Trigoni. GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation With Generative Adversarial Networks. In *International Conference on Robotics and Automation*, pages 5474–5480. IEEE, 2019.

[3] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson. There Are Many Consistent Explanations of Unlabeled Data: Why You Should Average. *International Conference on Learning Representations*, 2019.

[4] J. T. Barron. A General and Adaptive Robust Loss Function. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

[5] S. Bernhardt, S. A. Nicolau, L. Soler, and C. Doignon. The status of Augmented Reality in Laparoscopic Surgery as of 2016. *Medical Image Analysis*, 37:66–90, 2017.

[6] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. A Short Note on the Kinetics-700 Human Action Dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[7] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning From Monocular Videos. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 8001–8008, 2019.

[8] X. Chai, F. Gao, C. Qi, Y. Pan, Y. Xu, and Y. Zhao. Obstacle Avoidance for a Hexapod Robot in Unknown Environment. *Science China Technological Sciences*, 60 (6):818–831, 2017.

[9] Y. Chen, C. Schmid, and C. Sminchisescu. Self-Supervised Learning With Geometric Constraints in Monocular Video: Connecting Flow, Depth, and Camera. In *IEEE International Conference on Computer Vision*, pages 7063–7072, 2019.

[10] J. Cho, D. Min, Y. Kim, and K. Sohn. Deep Monocular Depth Estimation Leveraging a Large-scale Outdoor Stereo Dataset. *Expert Systems with Applications*, 178:114877, 2021.

[11] H. Choi, H. Lee, S. Kim, S. Kim, S. Kim, K. Sohn, and D. Min. Adaptive Confidence Thresholding for Monocular Depth Estimation. In *IEEE/CVF International Conference on Computer Vision*, pages 12808–12818, 2021.

[12] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELU). *arXiv preprint arXiv:1511.07289*, 2015.

[13] C. Doersch and A. Zisserman. Multi-Task Self-Supervised Visual Learning. In *IEEE International Conference on Computer Vision*, pages 2051–2060, 2017.

[14] D. Eigen, C. Puhrsch, and R. Fergus. Depth Map Prediction From a Single Image Using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.

[15] Z. Fang, X. Chen, Y. Chen, and L. V. Gool. Towards Good Practice for CNN-based Monocular Depth Estimation. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1091–1100, 2020.

[16] E. J. Fernandez-Sanchez, L. Rubio, J. Diaz, and E. Ros. Background Subtraction Model Based on Color and Depth Cues. *Machine Vision and Applications*, 25(5): 1211–1225, 2014.

[17] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar. Born Again Neural Networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[18] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[19] A. Geiger, P. Lenz, and R. Urtasun. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[21] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised Monocular Depth Estimation With Left-Right Consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[22] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into Self-Supervised Monocular Depth Prediction. In *International Conference on Computer Vision*, October 2019.

[23] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth From Videos in the Wild: Unsupervised Monocular Depth Learning From Unknown Cameras. *arXiv preprint arXiv:1904.04998*, 2019.

[24] P. Goyal, D. Mahajan, A. Gupta, and I. Misra. Scaling and Benchmarking Self-Supervised Visual Representation Learning. *arXiv preprint arXiv:1905.01235*, 2019.

[25] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon. 3D Packing for Self-Supervised Monocular Depth Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020.

[26] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang. Bounding Box Regression With Uncertainty for Accurate Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2888–2897, 2019.

[27] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.

[28] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi. A Comprehensive Overhaul of Feature Distillation. In *IEEE/CVF International Conference on Computer Vision*, pages 1921–1930, 2019.

[29] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[30] B. K. Horn and B. G. Schunck. Determining Optical Flow. In *Techniques and Applications of Image Understanding*, volume 281, pages 319–331. International Society for Optics and Photonics, 1981.

[31] Z. Hu, L. Xu, and M.-H. Yang. Joint Depth Estimation and Camera Shake Removal from Single Blurry Image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2893–2900, 2014.

[32] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox. Uncertainty Estimates and Multi-hypotheses Networks for Optical Flow. In *European Conference on Computer Vision*, pages 652–667, 2018.

[33] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging Weights Leads to Wider Optima and Better Generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[34] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016.

[35] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim. Complex Urban Dataset With Multi-Level Sensors From Highly Diverse Urban Environments. *The International Journal of Robotics Research*, 38(6):642–657, 2019.

[36] S. P. Johnson. How Infants Learn About the Visual World. *Cognitive Science*, 34 (7):1158–1184, 2010.

[37] K. Karsch, K. Sunkavalli, S. Hadap, N. Carr, H. Jin, R. Fonte, M. Sittig, and D. Forsyth. Automatic Scene Inference for 3D Object Compositing. *ACM Transactions on Graphics*, 33(3):32, 2014.

[38] V. Kaushik, K. Jindgar, and B. Lall. ADAADepth: Adapting Data Augmentation and Attention for Self-Supervised Monocular Depth Estimation. *arXiv preprint arXiv:2103.00853*, 2021.

[39] K. Kim, B. Ji, D. Yoon, and S. Hwang. Self-Knowledge Distillation with Progressive Refinement of Targets. In *IEEE/CVF International Conference on Computer Vision*, pages 6567–6576, 2021.

[40] M. Klodt and A. Vedaldi. Supervising the New with the Old: Learning SFM from SFM. In *European Conference on Computer Vision*, pages 698–713, 2018.

[41] C.-É. N. Laflamme, F. Pomerleau, and P. Giguère. Driving Datasets Literature Review. *arXiv preprint arXiv:1910.11968*, 2019.

[42] Z. Li and N. Snavely. MegaDepth: Learning Single-View Depth Prediction From Internet Photos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[43] Z. Li, N. Drenkow, H. Ding, A. S. Ding, A. Lu, F. X. Creighton, R. H. Taylor, and M. Unberath. On the Sins of Image Synthesis Loss for Self-supervised Depth Estimation. *arXiv preprint arXiv:2109.06163*, 2021.

[44] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving Warps for 3D Video Stabilization. *ACM Transactions on Graphics*, 28(3):1–9, 2009.

[45] L. Liu, X. Song, M. Wang, Y. Liu, and L. Zhang. Self-supervised Monocular Depth Estimation for All Day Images using Domain Separation. In *IEEE/CVF International Conference on Computer Vision*, pages 12737–12746, 2021.

[46] S. Liu, E. Johns, and A. J. Davison. End-To-End Multi-Task Learning With Attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[47] A. Loza, L. Mihaylova, N. Canagarajah, and D. Bull. Structural Similarity-based Object Tracking in Video Sequences. In *9th International Conference on Information Fusion*, pages 1–6. IEEE, 2006.

[48] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every Pixel Counts++: Joint Learning of Geometry and Motion With 3D Holistic Understanding. *arXiv preprint arXiv:1810.06125*, 2018.

[49] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised Learning of Depth and Ego-Motion From Monocular Video Using 3D Geometric Constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018.

[50] M. Mekni and A. Lemieux. Augmented Reality: Applications, Challenges and Future Trends. *Applied Computational Science*, 20:205–214, 2014.

[51] J. Mendoza and H. Pedrini. Self-Supervised Depth Estimation Based on Feature Sharing and Consistency Constraints. In *15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 134–141, Valletta, Malta, Feb. 2020.

[52] J. Mendoza and H. Pedrini. Adaptive Self-supervised Depth Estimation in Monocular Videos. In *International Conference on Image and Graphics*, pages 687–699. Springer, 2021.

[53] M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.

[54] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-Stitch Networks for Multi-Task Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[55] H. Mobahi, M. Farajtabar, and P. Bartlett. Self-Distillation Amplifies Regularization in Hilbert Space. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3351–3361. Curran Associates, Inc., 2020.

[56] S.-J. Park, K.-S. Hong, and S. Lee. RDFNnet: RGB-D Multi-level Residual Feature Fusion for Indoor Semantic Segmentation. In *IEEE International Conference on Computer Vision*, pages 4980–4989, 2017.

[57] R. Peng, R. Wang, Y. Lai, L. Tang, and Y. Cai. Excavating the Potential Capacity of Self-Supervised Monocular Depth Estimation. In *IEEE International Conference on Computer Vision*, 2021.

[58] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe. Unsupervised Adversarial Depth Estimation Using Cycled Generative Networks. In *International Conference on 3D Vision (3DV)*, pages 587–595. IEEE, 2018.

[59] A. Pilzer, S. Lathuiliere, N. Sebe, and E. Ricci. Refine and Distill: Exploiting Cycle-inconsistency and Knowledge Distillation for Unsupervised Monocular Depth Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9768–9777, 2019.

[60] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia. On the Uncertainty of Self-supervised Monocular Depth Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020.

[61] V. Prasad and B. Bhowmick. SfMLearner++: Learning Monocular Depth & Ego-Motion Using Meaningful Geometric Constraints. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2087–2096. IEEE, 2019.

[62] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12240–12249, 2019.

[63] L. G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[64] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard. Latent Multi-Task Architecture Learning. *arXiv preprint arXiv:1705.08142*, 2017.

[65] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[66] C. K. Sahu, C. Young, and R. Rai. Artificial Intelligence (AI) in Augmented Reality (AR)-Assisted Manufacturing Applications: A Review. *International Journal of Production Research*, 59(16):4903–4959, 2021.

[67] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with Stochastic Transformations and Perturbations for Deep Semi-supervised Learning. *Advances in Neural Information Processing Systems*, 29:1163–1171, 2016.

[68] M. R. U. Saputra, P. P. De Gusmao, Y. Almalioglu, A. Markham, and N. Trigoni. Distilling Knowledge from a Deep Pose Regressor Network. In *IEEE/CVF International Conference on Computer Vision*, pages 263–272, 2019.

[69] A. Saxena, S. H. Chung, and A. Y. Ng. Learning Depth From Single Monocular Images. In *Advances in Neural Information Processing Systems*, pages 1161–1168, 2006.

[70] Z. Shen, Z. He, and X. Xue. MEAL: Multi-model Ensemble via Adversarial Learning. In *Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 4886–4893, 2019.

[71] Y. Shi, J. Zhu, Y. Fang, K. Lien, and J. Gu. Self-Supervised Learning of Depth and Ego-Motion With Differentiable Bundle Adjustment. *arXiv preprint arXiv:1909.13163*, 2019.

[72] L. Smith and M. Gasser. The Development of Embodied Cognition: Six Lessons From Babies. *Artificial Life*, 11(1-2):13–29, 2005.

[73] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. *Advances in Neural Information Processing Systems*, 33, 2020.

[74] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, and B. Caine. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. *arXiv*, pages arXiv–1912, 2019.

[75] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

[76] C. Tang and P. Tan. BA-Net: Dense Bundle Adjustment Network. *arXiv preprint arXiv:1806.04807*, 2018.

[77] A. Tarvainen and H. Valpola. Mean Teachers are Better Role Models: Weight-averaged Consistency Targets Improve Semi-supervised Deep Learning Results. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[78] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano. Unsupervised Domain Adaptation for Depth Prediction from Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2396–2409, 2019.

[79] S. Tsutsui, D. Zhi, M. A. Reza, D. Crandall, and C. Yu. Active Object Manipulation Facilitates Visual Object Learning: An Egocentric Vision Study. *arXiv preprint arXiv:1906.01415*, 2019.

[80] J. Valentin, A. Kowdle, J. T. Barron, N. Wadhwa, M. Dzitsiuk, M. Schoenberg, V. Verma, A. Csaszar, E. Turner, and I. Dryanovski. Depth from Motion for Smartphone AR. *ACM Transactions on Graphics*, 37(6):1–19, 2018.

[81] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning Depth From Monocular Videos Using Direct Methods. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018.

[82] L. Wang and K.-J. Yoon. Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and new Outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[83] 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*, 13(4):600–612, 2004.

[84] O. Wiles, A. Sophia Koepke, and A. Zisserman. Self-supervised Learning of Class Embeddings from Video. In *IEEE/CVF International Conference on Computer Vision Workshops*, pages 1–8, 2019.

[85] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le. Unsupervised Data Augmentation for Consistency Training. *Advances in Neural Information Processing Systems*, 33, 2020.

[86] H. Xu, Z. Zhou, Y. Wang, W. Kang, B. Sun, H. Li, and Y. Qiao. Digging into Uncertainty in Self-supervised Multi-view Stereo. In *IEEE/CVF International Conference on Computer Vision*, pages 6078–6087, 2021.

[87] T.-B. Xu and C.-L. Liu. Data-Distortion Guided Self-Distillation for Deep Neural Networks. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 5565–5572, 2019.

[88] C. Yang, L. Xie, C. Su, and A. L. Yuille. Snapshot Distillation: Teacher-Student Optimization in one Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019.

[89] J. Yang, J. M. Alvarez, and M. Liu. Self-supervised Learning of Depth Inference for Multi-view Stereo. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7526–7534, 2021.

[90] N. Yang, L. von Stumberg, R. Wang, and D. Cremers. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. *arXiv preprint arXiv:2003.01060*, 2020.

[91] R. Yasarla and V. M. Patel. Uncertainty Guided Multi-scale Residual Learning-using a Cycle Spinning CNN for Single Image De-raining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8405–8414, 2019.

[92] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

[93] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry With Deep Feature Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018.

[94] Y. Zhang, L. Chen, Y. Liu, W. Zheng, and J.-H. Yong. Explicit Knowledge Distillation for 3D Hand Pose Estimation from Monocular RGB. In *British Machine Vision Conference*, 2020.

[95] Z. Zhang and M. R. Sabuncu. Self-Distillation as Instance-Specific Label Smoothing. *arXiv preprint arXiv:2006.05065*, 2020.

[96] L. Zhou, J. Ye, M. Abello, S. Wang, and M. Kaess. Unsupervised Learning of Monocular Depth Estimation With Bundle Adjustment, Super-Resolution and Clip Loss. *arXiv preprint arXiv:1812.03368*, 2018.

[97] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised Learning of Depth and Ego-Motion From Video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[98] Y. Zou, Z. Luo, and J.-B. Huang. DF-Net: Unsupervised Joint Learning of Depth and Flow Using Cross-Task Consistency. In *European Conference on Computer Vision*, pages 36–53, 2018.