DISS. ETH NO. 30130

LEARNING DEPTH FROM IMAGES

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES
(Dr. sc. ETH Zurich)

presented by
CE LIU
Master of Engineering in Computer Science and Technology
Harbin Institute of Technology

born on 01.09.1994

accepted on the recommendation of
Prof. Dr. Luc Van Gool, examiner
Prof. Dr. Andrea Vedaldi, co-examiner
Prof. Dr. Shubham Tulsiani, co-examiner
Prof. Dr. Radu Timofte, co-examiner

2024

ABSTRACT

Images have been extensively used in our daily life. Yet for many applications, it might be critical to infer the depth of each pixel. To this end, we study the problem of perceiving the depth from a single or stereo images.

Although there have been methods, especially the learning-based ones, achieving remarkable performance for depth perception, the deep neural networks might generalize poorly on unseen images, and produce wrong predictions. To address the above issues, in this thesis we advocate to exploit the invaluable invariances and priors in scenes by novel mathematical models.

To begin with, we investigate the conditional distribution of the depth map given a single image. Contrary to the existing methods, where per-pixel depth is assumed to be independent given the image, we introduce per-pixel covariance modeling that encodes its depth dependency with respect to all the scene points. Unfortunately, per-pixel depth covariance modeling leads to a computationally expensive continuous loss function, which we solve efficiently using the learned low-rank approximation of the overall covariance matrix. Notably, when tested on benchmark datasets, the model obtained by optimizing our loss function shows state-of-the-art results.

Then, we reveal the benefit of classical and well-founded variational constraints in the neural network design for the single-image dpeth prediction task. It is shown that imposing first-order variational constraints in the scene space together with popular encoder-decoder-based network architecture design provides excellent results. The imposed first-order variational constraints make the network aware of the depth gradient in the scene space, i.e., regularity. Our method at test time shows considerable improvements in depth prediction accuracy compared to the prior art and is accurate also at high-frequency regions in the scene space.

Next, we pursue efficient representations for the layouts, where the basic primitives, such as straight lines and vanishing points, can provide invaluable cues for depth. To make use of such an prior, we advocate to transform the primitives into the parameter space through

Hough transform. In addition, the line pooling module are proposed to select important primitives in parameter space. Our design improves the accuracy of off-the-shelf frameworks for monocular 3D object detection and depth prediction.

Finally, we turn to stereo images and introduce Stereo Risk, which formulates the scene disparity as an optimal solution to a continuous risk minimization problem. We demonstrate that L^1 minimization of the proposed continuous risk function enhances stereo-matching performance for deep networks, particularly for disparities with multimodal probability distributions. Furthermore, to enable the end-to-end network training of the non-differentiable L^1 risk optimization, we exploit the implicit function theorem, ensuring a fully differentiable network. A comprehensive analysis demonstrates our method's theoretical soundness and superior performance over the state-of-the-art methods across various benchmark datasets.

ZUSAMMENFASSUNG

Afbeeldingen worden veelvuldig gebruikt in ons dagelijks leven. Toch kan het voor veel toepassingen van cruciaal belang zijn om de diepte van elke pixel af te leiden. Daartoe bestuderen we het probleem van het waarnemen van de diepte vanuit een enkel of stereobeeld.

Hoewel er methoden zijn geweest, vooral de op leren gebaseerde, die opmerkelijke prestaties leverden op het gebied van diepteperceptie, zouden de diepe neurale netwerken slecht kunnen generaliseren op onzichtbare beelden en verkeerde voorspellingen kunnen opleveren. Om bovenstaande kwesties aan te pakken, pleiten we er in dit proefschrift voor om de onschatbare onveranderlijkheden en priors in scènes te exploiteren met behulp van nieuwe wiskundige modellen.

Om te beginnen onderzoeken we de voorwaardelijke verdeling van de dieptekaart gegeven een enkel beeld. In tegenstelling tot de bestaande methoden, waarbij wordt aangenomen dat de diepte per pixel onafhankelijk is gezien het beeld, introduceren we covariantiemodellering per pixel die de diepteafhankelijkheid codeert met betrekking tot alle scènepunten. Helaas, covariantie per pixeldiepte modellering leidt tot een computationeel dure continue verliesfunctie, die we efficiënt oplossen met behulp van de geleerde lage benadering van de algehele covariantiematrix. Met name wanneer het wordt getest op benchmarkdatasets, vertoont het model dat is verkregen door het optimaliseren van onze verliesfunctie state-of-the-art resultaten.

Vervolgens onthullen we het voordeel van klassieke en goed onderbouwde variatiebeperkingen in het neurale netwerkontwerp voor de dpeth-voorspellingstaak met één afbeelding. Er wordt aangetoond dat het opleggen van eerste-orde variatiebeperkingen in de scèneruimte, samen met het populaire op encoder-decoder gebaseerde netwerkarchitectuurontwerp, uitstekende resultaten oplevert. De opgelegde variatiebeperkingen van de eerste orde maken het netwerk bewust van de dieptegradiënt in de scèneruimte, dat wil zeggen de regelmaat. Onze methode tijdens de test laat aanzienlijke verbeteringen zien in de nauwkeurigheid van de dieptevoorspelling in vergelijking met de stand van de techniek en is ook nauwkeurig in hoogfrequente gebieden in de scèneruimte.

Vervolgens streven we naar efficiënte representaties voor de lay-outs, waarbij de basisprimitieven, zoals rechte lijnen en verdwijnpunten, waardevolle aanwijzingen voor diepte kunnen bieden. Om van een dergelijke prior gebruik te maken, pleiten wij ervoor om de primitieven via Hough-transformatie naar de parameterruimte te transformeren. Bovendien wordt voorgesteld dat de lijnpoolingmodule belangrijke primitieven in de parameterruimte selecteert. Ons ontwerp verbetert de nauwkeurigheid van kant-en-klare raamwerken voor monoculaire 3D-objectdetectie en dieptevoorspelling.

Ten slotte kijken we naar stereobeelden en introduceren we Stereo Risk, dat de ongelijkheid in scènes formuleert als een optimale oplossing voor een voortdurend risicominimalisatieprobleem. We laten zien dat L^1-minimalisatie van de voorgestelde continue risicofunctie de prestaties van stereomatching voor diepe netwerken verbetert, met name voor verschillen met multimodale waarschijnlijkheidsverdelingen. Om de end-to-end netwerktraining van de niet-differentieerbare L^1 risico-optimalisatie mogelijk te maken, maken we bovendien gebruik van de impliciete functiestelling, waardoor een volledig differentieerbaar netwerk wordt gegarandeerd. Een uitgebreide analyse toont de theoretische deugdelijkheid en superieure prestaties van onze methode aan ten opzichte van de modernste methoden in verschillende benchmarkdatasets.

PUBLICATIONS

The following publications are included as a whole or in parts in this thesis:

- Ce Liu et al. "Single Image Depth Prediction Made Better: A Multivariate Gaussian Take." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17346–17356.
- Ce Liu et al. "VA-DepthNet: A Variational Approach to Single Image Depth Prediction." In: *The Eleventh International Conference on Learning Representations*. 2022.
- Ce Liu et al. "Deep line encoding for monocular 3d object detection and depth prediction." In: *32nd British Machine Vision Conference*. BMVA Press. 2021, p. 354.
- Ce Liu et al. "Stereo Risk: A Continuous Modeling Approach to Stereo Matching." In: *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

Furthermore, the following publications were part of my PhD research, but are not covered in this thesis:

- Guolei Sun et al. "Indiscernible Object Counting in Underwater Scenes." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 13791–13801.
- Yawei Li et al. "Lsdir: A large scale dataset for image restoration." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 1775–1787.

ACKNOWLEDGMENTS

I have spent a wonderful and educational time in pursuing my PhD. It would not be possible without the help and guidance of many brilliant people.

I would like to thank my supervisor Prof. Dr. Luc Van Gool, and co-supervisor Prof. Dr. Radu Timofte. Because they gave me the opportunity to pursue my PhD in the Computer Vision Lab, and the freedom to choose research topics that interested me. The environment of the lab helped me to stay focused and motivated.

I am eternally thankful to Prof. Dr. Suryansh Kumar, who gave me incredible encouragement and support. When going though tough times, his suggestions helped me to find solutions and make progress. He contributed a lot to improving the quality of my works.

My gratitude also goes towards Prof. Dr. Shuhang Gu. We have countless discussions in the past years. My PhD would not have been the same without his insightful comments and kind help.

I also express my gratitude to Prof. Dr. Andrea Vedaldi and Prof. Dr. Shubham Tulsiani for their time on reviewing my thesis and being co-examiners of my PhD examination.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

Images are likely the most popular and effective media format in our daily life. It's known that images are usually intuitive, containing rich information, and can be stored and retrieved easily on the Internet [10, 47, 45, 183]. Furthermore, nowadays there are various accessible softwares to design and edit images [12, 21, 1, 177], and even cheap cameras in mobile phones can capture high-quality pictures [41, 263]. Hence, images are used not only by human beings for recording and illustration, but also by robots and autonomous cars for perceiving the environment [64, 22].

In this thesis, we focus on the images taken by ordinary cameras, such as mobile phones. The formation of images involves both the geometric and radiometric processes [162]. Starting from the light source, the light travels in the world following the Fermat's principle [19]. When interacting with the objects' surface, it can be absorbed, transmitted, or reflected, depending on the reflectance properties of the surface. In the end, the light passes through the lenses of the camera and projects onto the array of sensors.

The intricate interactions between the light and the objects can leave the final image with a wealth of information about the surfaces, such as texture and so on. As human being, we can further understand the semantic meaning of the scene, and even give a rough estimate of the 3D structure [235, 190, 199]. However, it is well known that the imaging process is non-reversible [63]. There is information lost during the procedure. One of the major concerns is that it is usually impossible to recover the depth of each pixel from only a single image [77, 238, 98], because there are many possible 3D positions for a point to produce the same pixel after projection.

In many real-world applications, however, the depth of pixels plays an important role. One example is by projecting the 3D points from the image onto different cameras we can syntheses novel views [236, 24]. Another one is that it is easier to find safe paths for robots if the 3D positions of obstacles are known [11, 179]. One straightforward solution to perceive depth is to resort to special sensors such as LiDAR

(a) Image (b) Depth

Figure 1.1: An illustration of single-image depth prediction. (a) The test image with multiple objects and complex lighting. (b) The predicted depth map from deep networks [143]. Lighter color indicates smaller depth values.

[94, 184] or Kinect [263, 209, 74]. However, there are many scenarios where only images are available. Therefore a natural question arises: how to recover the depth from the images?

The above question has been extensively studied in the last fifty years, and various solutions have been proposed. In general it is ill-posed and requires to provide additional images captured under different viewpoints [77, 81], lighting conditions [87, 237], or focus [213, 201]. With more observed images, we can construct more mathematical constraints about the depth of pixels and obtain predictions that are more reliable [77]. Although feasible, the methods usually make strict assumptions about the scenes and the imaging process. In practical applications, there are often cases where it is difficult to find sufficient and reliable mathematical constraints to infer the depth.

More recently, with the explosion of big data [46, 267, 56] and the availability of powerful computing resources [40, 39], deep learning techniques have drawn more and more attention [80, 68, 120, 113, 202]. In the field of computer vision, the Convolutional Neural Networks (CNN) [69, 119], Long Short-Term Memory (LSTM) networks [82, 225, 83], and Vision Transformers (ViT) [51, 75] have improved the state-of-the-art performance of various tasks by a significant margin. The core of deep networks is to learn intricate structures of raw inputs by the composition of non-linear transformations, which usually includes millions of learnable parameters and can represent a wide variety of

functions [120]. Moreover, comparing to hand-crafted features, the learned ones are optimized on large-scale training set in an end-to-end manner and therefore more suitable for the specific task [196, 84].

For depth prediction, the deep learning techniques have also been widely used. Popular attempts include applying deep networks for single-image depth prediction [55, 224, 33], stereo matching [257, 103, 259], and multi-view depth prediction [251, 252]. Specifically, the parameters of networks are optimized to predict depth or disparities that are close to the ground truth. The networks can discover discriminative features from intensities of pixels, and have ranked first in the leaderboards of various depth perception tasks [64, 206, 200]. An illustration of single-image depth prediction (SIDP) by the deep network is shown in Fig. 1.1.

However, the deep learning technique is still not perfect. For human being it is difficult to understand what the networks have learned [258, 13, 170] because the functions are represented by the composition of hundreds of transformations with millions of parameters. Moreover, the networks might generalize poorly on unseen images [172, 99] and produce wrong predictions especially when the number of training images are insufficient.

To take the perception of depth further, in this thesis we advocate that a more precise mathematical modeling of depth or disparity is beneficial. Because the formulation not only explains the properties of depth to the human being, but also facilitates the depth perception by encoding our priors. We could design network structures and loss functions under the guidance of the formulation. More specifically, the thesis includes four parts: Firstly, we observe the depth at different pixels often show correlation, hence we propose to use the multivariate Gaussian to model the distribution of depth. Secondly, to further exploit the correlation at neighboring pixels, we regularize the single-view depth prediction by variational constraints. Thirdly, we show the basic primitives, like straight lines and vanishing points, provide invaluable cues for depth, and we advocate to model the primitives efficiently in parameter space. Finally, we turn attention to stereo images and formulate the continuous disparity as the optimal solution of minimizing L^1 risk, which alleviates the disturbance from outliers. In the next sections, we briefly introduce each part.

1.1 MULTIVARIATE GAUSSIAN FOR DEPTH

In the pursuit of perfect depth estimation, most existing state-of-the-art learning techniques predict a single scalar depth value per pixel. Yet, it is well-known that the trained model has accuracy limits and can predict imprecise depth [101, 102]. Therefore, it's important to be mindful of the expected depth variations in the model's prediction at test time. Accordingly, we introduce an approach that performs continuous modeling of per-pixel depth, where we can predict and reason about the per-pixel depth and its distribution.

Existing methods in this direction model depth per pixel independently. It is clearly unreasonable, however, to assume absolute democracy among each pixel, especially for very closeby scene points. Therefore, it is natural to think of modeling this problem in a way where where depth at a particular pixel can help infer, refine, and constrain the distribution of depth value for other image pixels. Nevertheless, it has yet to be known a prior the neighboring relation among pixels in the scene space to define the depth covariance among them. We do that here by defining a very general covariance matrix of dimension `number of pixels` $\times$ `number of pixels`, *i.e.*, depth prediction at a given pixel is assumed to be dependent on all other pixels' depth. Unfortunately, per-pixel depth covariance modeling leads to a computationally expensive continuous loss function. To efficiently optimize the proposed formulation, we parameterize the covariance matrix, assuming that it lies in a rather low-dimensional manifold so that it can be learned using a simple neural network.

For training our deep network, we utilize the negative log likelihood as the loss function. Notably, when tested on benchmark datasets, the SIDP model obtained by optimizing our loss function shows state-of-the-art results.

1.2 VARIATIONAL CONSTRAINT

An image of a general scene—indoor or outdoor, has a lot of spatial regularity. While state-of-the-art deep neural network methods for SIDP learn the scene depth from images in a supervised setting, they often overlook the invaluable invariances and priors in the rigid scene space. In this part, we resort to the physics of variation in the neural network design for better generalization of the SIDP network, which by

the way, keeps the essence of affine invariance. We show that imposing first-order variational constraints in the scene space together with popular encoder-decoder-based network architecture design provides excellent results for the supervised SIDP task. The imposed first-order variational constraint makes the network aware of the depth gradient in the scene space, *i.e.*, regularity. As we demonstrate later in the thesis, such an idea boosts the network's depth accuracy while preserving the high-frequency and low-frequency scene information.

1.3 LINE PRIORS FOR DEPTH

In man-made environments, especially in autonomous driving scenarios, the basic primitives, like straight lines and vanishing points, provide invaluable cues for depth. Because their angle or position indicates the 3D layout of the whole scene. To explicitly represent the semantics (*e.g.*, guard rail, horizontal line, *etc.*) and algebraic parameters of lines, we perform deep Hough transform [53, 138] on the feature map of deep networks. The voting for a line is obtained by aggregating the features along the line, which encodes the semantic information from the entire line. In addition, the angle and position are indicated by the voting location in parameter space. For efficiency, we further propose the line pooling module to select important lines in parameter space. We apply our design to off-the-shelf frameworks for monocular 3D object detection and depth prediction in autonomous driving scenarios. The improvements demonstrate the effectiveness of our design.

1.4 STEREO RISK

In this part, we consider the situation where a rectified stereo image pair is available. Accordingly, the depth perception problem boils down to estimating the per-pixel displacement from left to right images, popularly known as a disparity map [200].

One challenge is how to produce continuous disparity values, given only a limited number of candidate pixels to match. Recent works either predict a real-valued offset by neural networks, or take the expectation value of the categorical distribution of matching similarity.

We introduce a radically different perspective by framing it as a search problem of finding the minimum risk of disparity values. Specifically, the risk is defined by averaging the prediction error with respect to all possible values of the ground-truth disparity. At the time of making the prediction, the ground truth is unavailable, which is therefore approximated by the disparity hypotheses with a categorical distribution. We search for a disparity value as our prediction that achieves minimal overall risk involved with it.

Moreover, we demonstrate that the commonly used disparity expectation is a special case of L^2 error function within the proposed risk formulation, which is sensitive to multi-modal distribution and may result in the over-smooth solution. In contrast, we advocate the use of the L^1 error function during risk minimization. We have extensively evaluated the proposed method on a variety of stereo matching datasets. Our approach demonstrates superior performance compared to many state-of-the-art methods on benchmarks.

2

MULTIVARIATE GAUSSIAN FOR DEPTH

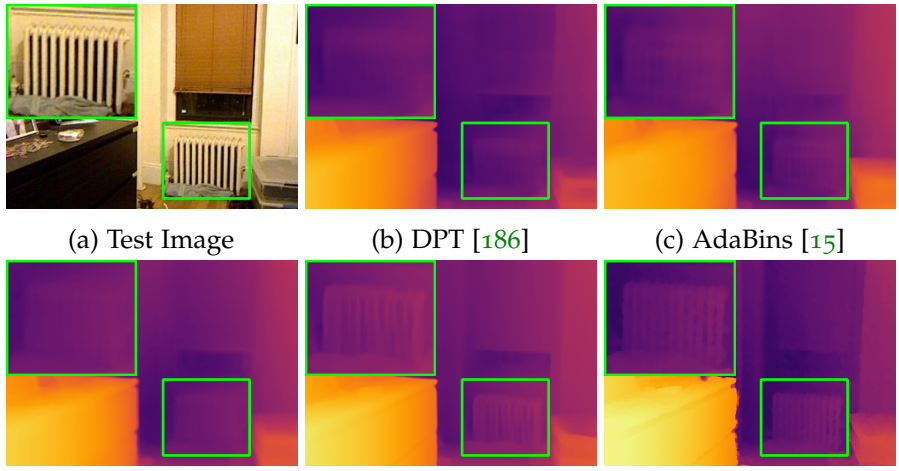
This chapter is based on our paper: Ce Liu et al. "Single Image Depth Prediction Made Better: A Multivariate Gaussian Take." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17346–17356.

2.1 INTRODUCTION

Recovering the depth of a scene using images is critical to several applications in computer vision [3, 59, 114, 115, 100]. It is well founded that precise estimation of scene depth from images is likely only under multi-view settings [223] —which is indeed a correct statement and hard to contend[1]. But what if we could effectively learn scene depth using images and their ground-truth depth values, and be able to predict the scene depth using just a single image at test time? With the current advancements in deep learning techniques, this seems quite possible empirically and has also led to excellent results for the single image depth prediction (SIDP) task [145, 186]. Despite critical geometric arguments against SIDP, practitioners still pursue this problem not only for a scientific thrill but mainly because there are several real-world applications in which SIDP can be extremely beneficial. For instance, in medical [150], augmented and virtual reality [86, 192], gaming [73], novel view synthesis [193, 194], robotics [217], and related vision applications [186, 95].

Regardless of remarkable progress in SIDP [134, 256, 133, 182, 2, 142, 145], the recent state-of-the-art deep-learning methods, for the time being, just predict a single depth value per pixel at test time [134]. Yet, as is known, trained models have accuracy limits [101]. As a result, for broader adoption of SIDP in applications, such as robot vision and control, it is essential to have information about the reliability of predicted depth. Consequently, we model the SIDP task using a continuous distribution function. Unfortunately, it is challenging, if not

1 As many 3D scene configurations can have the same image projection.



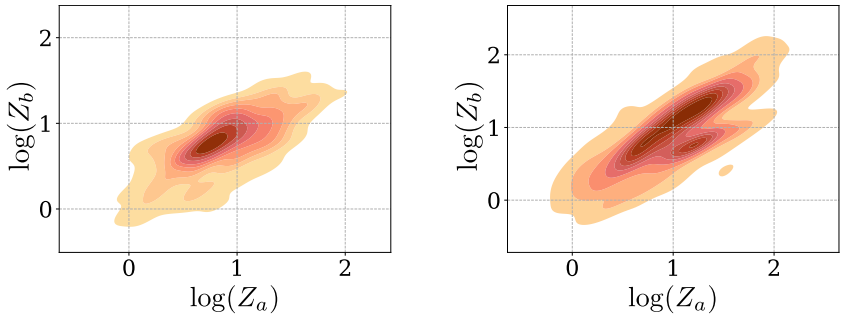
(a) Test Image (b) DPT [186] (c) AdaBins [15]

(d) NeWCRFs [256] (e) **Ours** (f) Ground Truth

Figure 2.1: **Qualitative Comparison**. By modeling scene depth as multivariate Gaussian and enforcing the parametric low-rank covariance constraints in the loss function, we observe that our model can reliably predict depth for both high-frequency and low-frequency scene details. In the above example, we can notice better qualitative results than the state-of-the-art methods.

impossible, to precisely model the continuous 3D scene. In this regard, existing methods generally resort to increasing the size and quality of the dataset for better scene modeling and improve SIDP accuracy. On the contrary, little progress is made in finding novel mathematical modeling strategies and exploiting the prevalent scene priors. To this end, we propose a multivariate Gaussian distribution to model scene depth. In practice, our assumption of the Gaussian modeling of data is in consonance with real-world depth data (see Fig. 2.2) and generalizes well across different scenes. Furthermore, many computer and robot vision problems have successfully used it and benefited from Gaussian distribution modeling in the past [30, 269, 212, 79, 247, 189, 169].

Let's clarify this out way upfront that this is not for the first time an approach with a motivation of continuous modeling for SIDP is proposed [6, 117, 101, 109, 88, 178]. Yet, existing methods in this direction model depth per pixel independently. It is clearly unreasonable,



(a) First scene (b) Second scene

Figure 2.2: The marginal ground-truth depth distribution for a pixel pair Z_a, Z_b for two scenes. The depth values for the pixel pair are taken from the fixed image location in the dataset, but the selected images are visually similar for the suitability of the feature and its corresponding depth values. The statistics show that the Gaussian distribution assumption with covariance modeling is a sensible choice for SIDP problem and not an unorthodox belief arranged or staged for an intricate formulation.

in SIDP modeling, to assume absolute democracy among each pixel, especially for very closeby scene points. Therefore, it is natural to think of modeling this problem in a way where depth at a particular pixel can help infer, refine, and constrain the distribution of depth value for other image pixels. Nevertheless, it has yet to be known a priori the neighboring relation among pixels in the scene space to define the depth covariance among them. We do that here by defining a very general covariance matrix of dimension `number of pixels` $\times$ `number of pixels`, i.e., depth prediction at a given pixel is assumed to be dependent on all other pixels' depth.

Overall, we aim to advocate multivariate Gaussian modeling with a notion of depth dependency among pixels as a useful scene prior. Now, add a fully dependent covariance modeling proposal to it—as suitable relations among pixels are not known. This makes the overall loss function computationally expensive. To efficiently optimize the proposed formulation, we parameterize the covariance matrix, assuming that it lies in a rather low-dimensional manifold so that it can be

learned using a simple neural network. For training our deep network, we utilize the negative log likelihood as the loss function. The trained model when tested on standard benchmark datasets gives state-of-the-art results for SIDP task (see Fig. 2.1 for qualitative comparison).

Contributions. To summarize, our key contributions are:

- A novel formulation to perform multivariate Gaussian covariance modeling for solving the SIDP task in a deep neural network framework is introduced.
- The introduced multivariate Gaussian covariance modeling for SIDP is computationally expensive. To solve it efficiently, the paper proposes to learn the low-rank covariance matrix approximation by deep neural networks.
- Contrary to the popular SIDP methods, the proposed approach provides better depth as well as a measure of the suitability of the predicted depth value at test time.

2.2 RELATED WORK

Predicting the scene depth from a single image is a popular problem in computer vision with long-list of approaches. To keep our review of the existing literature succinct and on-point, we discuss work of direct relevance to the proposed method. Roughly, we divide well-known methods into two sub-category based on their depth prediction modeling.

***(i)* General SIDP.** By general SIDP, we mean methods that predict one scalar depth value per image pixel at inference time. Earlier works include Markov Random Field (MRF) or Conditional Random Fields (CRF) modeling [197, 198, 199, 230]. With the advancement in neural network-based approaches, such classical modeling ideas are further improved using advanced deep-network design [147, 127, 256, 242, 240]. A few other stretches along this line use piece-wise planar scene assumption [124]. Other variations in deep neural network-based SIDP methods use ranking, ordinal relation constraint, or structured-guided sampling strategy [270, 33, 239, 135, 58]. The main drawback with the above deep-leaning methods is that they provide an over-smoothed depth solution, and most of them rely on some heuristic formulation for depth-map refinement as a post-refinement step.

Recently, transformer networks have been used for better feature aggregation via an increase in the network receptive field [248, 15, 28, 256] or with the use of attention supervision [28] leading to better SIDP accuracy. Another mindful attempt is to exploit the surface normal and depth relation. To this end, [90] introduces both normal and depth loss for SIDP, whereas [253] proposes using virtual normal loss for imposing explicit 3D scene constraint and utilizing long-range scene dependencies. Long et al. [154] improved over [253] by introducing an adaptive strategy to compute local patch surface normal by randomly sampling for candidate triplets. A short time ago, [9] showed commendable results using normal-guided depth propagation [181] with a depth-to-normal and learnable normal-to-depth module.

***(ii)* Probabilistic SIDP.** In comparison, there are limited pieces of literature that directly target to solve SIDP in a probabilistic way, where the methods could predict the scene depth and simultaneously can reason about its prediction quality. Generally, popular methods from uncertainty modeling in deep-network are used as it is for such purposes. For instance, Kendall et al. [101] Bayesian uncertainty in deep-networks, Lakshminarayanan et al. [117] deep ensemble-based uncertainty modeling, Amini et al. [6] deep evidential regression approach, is shown to work also for the depth prediction task. Yet, these methods are very general and can be used for most, if not all, computer vision problems [61]. Moreover, these methods treat each pixel independently, which may lead to inferior SIDP modeling.

This brings us to the point that application-specific priors, constraints, and settings could be exploited to enhance the solution, and we must not wholly depend on general frameworks to tackle the problem with similar motivation [101, 117, 6]. Therefore, this paper advocate using per-pixel multivariate Gaussian covariance modeling with efficient low-rank covariance parametric representation to improve SIDP for its broader application. Furthermore, we show that the depth likelihood due to multivariate Gaussian distribution modeling can help define better loss function and allow depth covariance learning based on scene feature regularity. With our modeling formulation, the derived loss function naturally unifies the essence of L2 loss, scale-invariant loss, and gradient loss. These three losses can be derived as a special case of our proposed loss.

2.3 PROPOSED METHOD

To begin with, let's introduce problem setting and some general notations, which we shall be using in the rest of the paper. That will help concisely present our formulation, its useful mathematical insights, discussion, and application to Bayesian uncertainty estimation in deep networks.

Problem Setting. Given an image $\mathbf{I} \in \mathbb{R}^{m \times n}$ at test time, our goal is to predict the reliable per-pixel depth map $\mathbf{Z} \in \mathbb{R}^{m \times n}$, where m, n symbolize the number of image rows and cols, respectively. For this problem, we reshape the image and corresponding ground-truth depth map as a column vector represented as $I \in \mathbb{R}^{N \times 1}$ and $Z^{gt} \in \mathbb{R}^{N \times 1}$, respectively. Here, $N = m \times n$ is the total number of pixels in the image and $\mathcal{D}$ denotes the train set.

2.3.1 *Multivariate Gaussian Modeling*

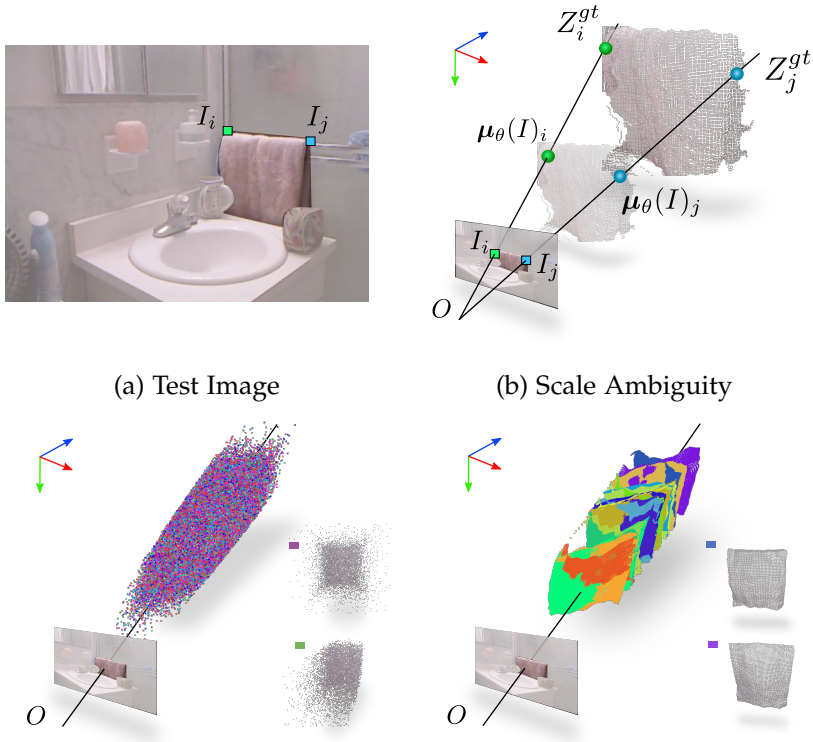
Let's assume the depth map Z corresponding to image I follows a N-dimensional Gaussian distribution. Accordingly, we can write the distribution Φ given I as

$$\Phi(Z|\theta, I) = \mathcal{N}\big(\boldsymbol{\mu}_\theta(I), \boldsymbol{\Sigma}_\theta(I, I)\big). \tag{2.1}$$

Where, $\boldsymbol{\mu}_\theta(I) \in \mathbb{R}^{N \times 1}$ and $\boldsymbol{\Sigma}_\theta(I, I) \in \mathbb{R}^{N \times N}$ symbolize the mean and covariance of multivariate Gaussian distribution $\mathcal{N}$ of predicted depth, respectively. The θ represents the parametric description of mean and covariance, which the neural network can learn at train time. It is important to note that with such network modeling, it is easy for the network to reliably reason about the scene depth distribution of similar-looking images at test time. Using the general form of multivariate Gaussian density function, the log probability density of Eq.(2.1) could be elaborated as

$$\begin{aligned} \log \Phi(Z|\theta, I) = -\frac{N}{2} \log 2\pi - \frac{1}{2} \log \det(\boldsymbol{\Sigma}_\theta(I, I)) - \\ \frac{1}{2}(Z - \boldsymbol{\mu}_\theta)^T (\boldsymbol{\Sigma}_\theta(I, I))^{-1} (Z - \boldsymbol{\mu}_\theta). \end{aligned} \tag{2.2}$$

Eq.(2.2) is precisely the formulation we strive to implement. Yet, computing the determinant and inverse of a $N \times N$ covariance matrix can



(a) Test Image (b) Scale Ambiguity

(c) Independent Modeling (d) Our Modeling

Figure 2.3: (a) Test image of an indoor bathroom scene. (b) The problem of scale ambiguity: showing several possible 3D point-cloud configurations of the towel with same imaging region. (c) If the depth values from the towel region are back-projected in the scene space under the independent Gaussian distribution assumption of the depth map. Clearly, the 3D point cloud results are not encouraging. (d) Samples due to our multivariate Gaussian distribution modeling that constrain the pixel depth with learned covariance. We observe the samples drawn from our modeling provide better 3D point clouds. Note: depth map is transformed to point cloud for visualization.

be computationally expensive, *i.e.*, $O(N^3)$, for a reasonable image resolution. Previous methods in this direction usually restrict covariance to be diagonal [101, 117], *i.e.*, $\mathbf{\Sigma}_\theta(I,I) = \mathbf{diag}(\boldsymbol{\sigma}^2_\theta(I))$ with $\boldsymbol{\sigma}_\theta$ as the standard deviation learned by the network with parameter θ. Even though such a simplification leads to computationally tractable algorithm $O(N)$, it leads to questionable depth prediction at test time. The reason for that is in SIDP, each pixel's depth could vary by a single scale value which must be the same for all the pixels under the rigid scene assumption (see Fig.2.3b). By assuming covariance to be zero, each pixel is modeled independently; hence the coherence among scene points is lost completely. It can also be observed from Fig.(2.3c) when the covariance matrix is restricted to a diagonal matrix, the sampled depth from $\Phi(Z|\theta, I)$ is incoherently scattered. Therefore, it is pretty clear that multivariate covariance modeling is essential (see Fig. 2.3d) despite being computationally expensive.

To overcome the computational bottleneck in covariance modeling, we propose to exploit $\mathbf{\Sigma}_\theta$ parametric form with low-rank assumption. It is widely studied in statistics that multivariate data relation generally has low-dimensional structure [229, 268, 118]. Since, covariance matrix is symmetric and positive definite, we write $\mathbf{\Sigma}_\theta$ in parametric form *i.e.*,

$$\mathbf{\Sigma}_\theta(I,I) = \Psi_\theta(I)\Psi_\theta(I)^T + \sigma^2\mathbf{eye}(N), \tag{2.3}$$

where, $\Psi_\theta(I) \in \mathbb{R}^{N\times M}$ is learned by deep networks with parameter θ with $M \lll N$. $\mathbf{eye}(N) \in \mathbb{R}^{N\times N}$ is a slang for identity matrix. $\Psi_\theta(I)\Psi_\theta(I)^T$ is symmetric and $\sigma^2\mathbf{eye}(N)$ guarantees positive definite matrix with $\sigma > 0$ as some positive constant. By using the popular matrix inversion lemma [180] and Eq.(2.3) parametric form, log probability density defined in Eq.(2.2) can be re-written as

$$\begin{aligned}\log\Phi(Z|\theta,I) = -\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2}\log\det(\mathbf{A}) - \\ \frac{\sigma^{-2}}{2}\mathbf{r}^T\mathbf{r} + \frac{\sigma^{-4}}{2}\mathbf{r}^T\Psi_\theta(I)\mathbf{A}^{-1}\Psi_\theta(I)^T\mathbf{r},\end{aligned} \tag{2.4}$$

with $\mathbf{r} = Z - \boldsymbol{\mu}_\theta(I)$, and $\mathbf{A} = \sigma^{-2}\Psi_\theta(I)^T\Psi_\theta(I) + \mathbf{eye}(M)$. It can be shown that the above form for modeling covariance is computationally tractable with complexity $O(NM + M^3)$ [189] as compared to $O(N^3)$ since $M \lll N$. We use Eq.(2.4) as negative log likelihood (NLL) loss function, *i.e.*, $\mathcal{L}_{NLL} = -\log\Phi(Z^{gt}|\theta, I)$ to train the network for learning

per-pixel depth, and covariance w.r.t.all the pixels, hence overcoming the shortcomings with prior works in SIDP.

2.3.2 *Deeper Insights into the Formulation*

A detailed analysis of Eq.(2.4) and how it naturally encapsulates the notion of popular loss functions are presented for better understanding. Concretely, we show that "L2 Loss", "Scale Invariant Loss (SI Loss)", and "Gradient Loss (G-Loss)" as a special case of Eq.(2.4); thus, our formulation is more general. Later, in the subsection, we apply Eq.(2.4) to well-known Bayesian uncertainty modeling [61] in deep neural networks showing improved uncertainty estimation than independent Gaussian assumption.

2.3.2.1 *Relation to Popular Loss Function*

By taking Eq.(2.4) NLL form as the training loss, i.e., $-\log \Phi(Z^{gt}|\theta, I)$, we show that using the special values for $\Psi_\theta(I)$, the NLL loss can be reduced to some widely-used losses (see Fig. 2.4). Here, symbolizes $Z^{gt} \in \mathbb{R}^{N\times 1}$. Denoting $\mathbf{r} = Z^{gt} - \boldsymbol{\mu}_\theta(I)$, we derive the relation.

(i) ***Case I***. Substituting $\Psi_\theta(I) = \mathbf{0}_N$ in Eq.(2.4) will give

$$-\log \Phi(Z^{gt}|\theta, I) \propto \mathbf{r}^T\mathbf{r}, \tag{2.5}$$

which is equivalent to the "L2 loss" function. Here, $\mathbf{0}_N$ is a column vector with N elements, all set to 0.

(ii) ***Case II***. Substituting $\Psi_\theta(I) = \mathbf{1}_N$ in Eq.(2.4) will give

$$-\log \Phi(Z^{gt}|\theta, I) \propto \mathbf{r}^T\mathbf{r} - \frac{\alpha}{N}(\mathbf{r}^T\mathbf{1}_N)^2, \tag{2.6}$$

where, $\alpha = (\sigma^{-2}N)/(\sigma^{-2}N + 1)$ and $\mathbf{1}_N$ is a column vector with N elements set to 1. Assuming $\sigma^{-2}N \ggg 1$, which is mostly the experimental setting in SIDP, then $\alpha \approx 1$ and Eq.(2.6) becomes equivalent to "Scale Invariant Loss".

(iii) ***Case III***. Here, we want to show the relation between Eq.(2.4) and gradient loss. But, unlike previous cases, it's a bit involved. So, for simplicity, consider the gradient of the flattened depth map (*i.e.*, a column vector)[2]. The general squared gradient loss between the ground-truth

2 ignoring border pixels for simple 1D case

and predicted depth can be computed as $(\nabla Z^{gt} - \nabla \boldsymbol{\mu}_\theta(I))^T(\nabla Z^{gt} - \nabla \boldsymbol{\mu}_\theta(I))$, where $\nabla \in \mathbb{R}^{N\times N}$ is the gradient operator for computing the first order difference of Z^{gt} and $\boldsymbol{\mu}_\theta(I)$ [191]. Taking out the common factor, we can re-write the gradient loss as $\big(\nabla(Z^{gt} - \boldsymbol{\mu}_\theta(I))\big)^T\big(\nabla(Z^{gt} - \boldsymbol{\mu}_\theta(I))\big)$. Simplifying using the matrix transpose property and it can be written in compact form as $\mathbf{r}^T(\nabla^T\nabla)\mathbf{r}$, which is equivalent to the Gaussian multivariate form in Eq.(2.2). Let's denote $J \triangleq (\nabla^T\nabla)^{-1}$, where $J_{i,j} = \min\{i,j\} - ij/(N+1)$ [44]. However, J is difficult to parameterize and decompose into low-dimensional form. Concretely, we want to factorize J into $\Psi_\theta(I)\Psi_\theta(I)^T + \sigma^2\mathbf{eye}(N)$ that fits the notion developed in Eq.(2.2) and Eq.(2.3), with $\Psi_\theta(I) \in \mathbb{R}^{N\times M}$ and $M \lll N$.

Fortunately, it is possible to approximate J as $J \approx \big(\Psi_\theta(I)\Psi_\theta(I)^T + \sigma^2\mathbf{eye}(N)\big)$ by using well-known Eigen approximation [67]. To be precise, setting $\Psi_\theta(I)$ to

$$\Psi_\theta(I)_{k,l} = \sqrt{\boldsymbol{\lambda}(J)_l}\mathbf{U}(J)_{k,l} \tag{2.7}$$

where $\boldsymbol{\lambda}(J) \in \mathbb{R}^{N\times 1}$ and $\mathbf{U}(J) \in \mathbb{R}^{N\times N}$ are the sorted eigenvalues and corresponding eigenvectors of J, respectively that can be computed using $\boldsymbol{\lambda}(J)_l = (2 - 2\cos\frac{l\pi}{N+1})^{-1}$ and $\mathbf{U}(J)_{k,l} = (-1)^{k+1}\sin\frac{kl\pi}{N+1}$ [174].

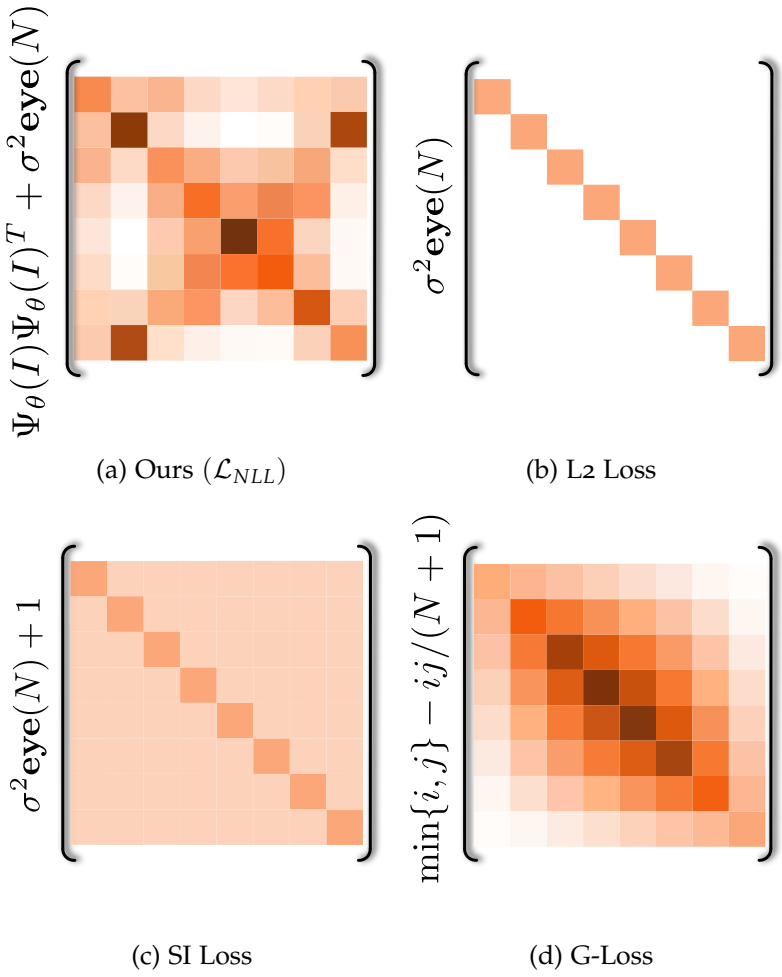
2.3.2.2 *Application in Uncertainty Estimation*

We apply Eq.(2.4) to the popular Bayesian uncertainty modeling in neural networks. Given $\Phi(Z|\theta, I)$ as aleatoric uncertainty for depth map Z [101], we can compute the Bayesian uncertainty by marginalising over the parameters' posterior distribution using the following well-known equation [17]:

$$\Phi(Z|I,\mathcal{D}) = \int \Phi(Z|\theta, I)\Phi(\theta|\mathcal{D})d\theta \tag{2.8}$$

where $\mathcal{D}$ is the train set. The analytic integration of Eq.(2.8) is difficult to compute in practice, and is usually approximated by Monte Carlo integration, such as ensemble [117] and dropout [61]. Suppose we have sampled a set of parameters $\Theta \triangleq \{\theta^s\}_{s=1}^S$ from $\Phi(\theta|\mathcal{D})$. The integration is popularly approximated as

$$\Phi(Z|I,\mathcal{D}) = \frac{1}{S}\sum_s \Phi(Z|\theta^s, I). \tag{2.9}$$



(a) Ours ($\mathcal{L}_{NLL}$) (b) L2 Loss

(c) SI Loss (d) G-Loss

Figure 2.4: **The covariance matrix of loss function.** (a) Ours $\mathbf{\Sigma}_\theta$. (b)-(d) shows the equivalent covariance matrix for the (b) L2 loss, (c) scale invariant loss, and (d) gradient loss. It can be observed that our covariance already contains most, if not all, information that could be recovered by employing different loss functions, hence showing the generality of our formulation.

The $\Phi(Z|I,\mathcal{D})$ denotes the mixture of Gaussian distributions [117]. The mean and covariance of the distribution is computed as $\bar{\boldsymbol{\mu}}(I) = \frac{1}{S}\sum_s \boldsymbol{\mu}^s(I)$ and $\bar{\boldsymbol{\Sigma}}(I,I) = \bar{\Psi}(I)\bar{\Psi}(I)^T + \sigma^2\mathbf{eye}(N)$, respectively [234], which in fact has the same form as Eq.(2.3), where we compute $\bar{\Psi}$ using the following expression

$$\bar{\Psi} = \frac{1}{\sqrt{S}}\mathbf{concat}(\Psi^1,\ldots,\Psi^S,\boldsymbol{\mu}^1-\bar{\boldsymbol{\mu}},\ldots,\boldsymbol{\mu}^S-\bar{\boldsymbol{\mu}}). \tag{2.10}$$

So, from the derivations in Sec.(2.3.2.1) and Sec.(2.3.2.2), it is quite clear that our proposed Eq.(2.4) is more general and encapsulates flavors of popular loss functions widely used in deep networks. For the SIDP problem we need such a loss function for deep neural network parameters learning. Next, we discuss the implementation in our proposed pipeline and usefulness of our introduced loss function.

2.3.3 *Overall Pipeline*

To keep our pipeline description simple, let's consider the image and depth map in 2D form instead of a column vector. For brevity, we slightly abuse the notation hereafter. Here, we use the same notation we defined for the 1D Gaussian distribution case for simplicity. For a better understanding of our overall pipeline, we provide architectural implementation details following Fig.(2.5) blueprint, *i.e.*, *(i)* Encoder details, *(ii)* Decoder details, followed by *(iii)* Train and test time settings.

***(i)* Encoder Details.** Our encoder takes the image $\mathbf{I} \in \mathbb{R}^{m\times n}$ as input and gives a hierarchical feature maps $\mathbf{F} = \{\mathbf{F}^1,\mathbf{F}^2,\mathbf{F}^3,\mathbf{F}^4\}$ as output, where $\mathbf{F}^i \in \mathbb{R}^{C^i\times m^i\times n^i}$ denotes a feature map with channels C^i, and resolution $m^i \times n^i$. We adopt the Swin-Large [152] as the encoder. Specifically, it includes four stages of non-linear transformation to extract features from the input image, where each stage contains a series of transformer blocks to learn non-linear transformation and a downsampling block to reduce the resolution of the feature map by 2. We collect the output feature map from the last block of the i-th stage as $\mathbf{F}_i$.

***(ii)* Decoder Details.** The *U-decoder* (see Fig.2.5) estimates a set of depth maps $\{\boldsymbol{\mu}^i_\theta(\mathbf{I})\}_{i=1}^4$. The U-decoder first estimates $\boldsymbol{\mu}^4_\theta(\mathbf{I})$ only from $\mathbf{F}^4$ by a convolution layer, then upsamples and refines the depth map in a hierarchical manner. At the i-th stage, where i decreases from 3 to

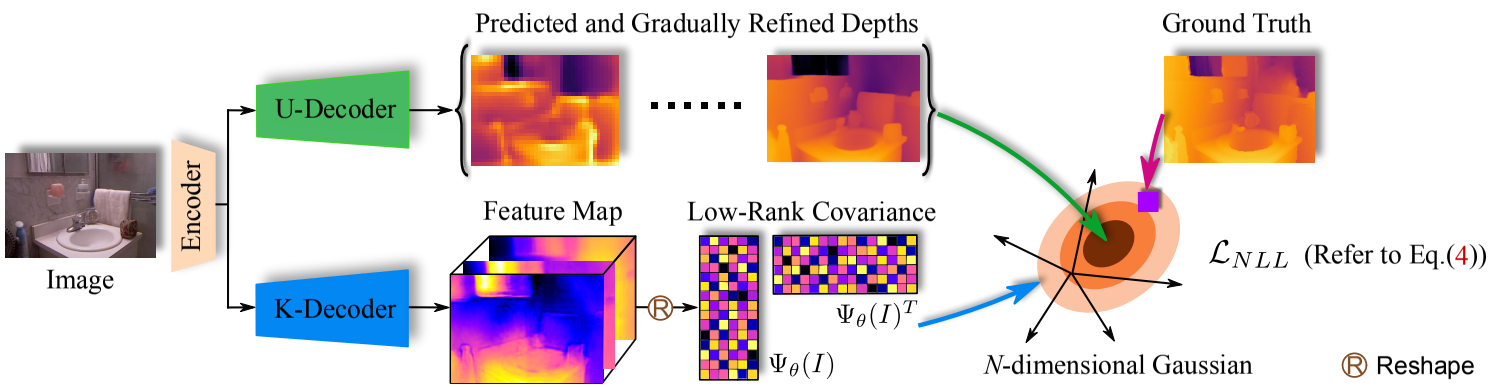


Figure 2.5: **Overview of our framework**. Given an image, first an encoder is employed to extract features. Then the U-Decoder will predict and gradually refine the depth maps. And the K-Decoder is responsible for predicting the factor $\Psi_\theta(I)$ for modeling the covariance. In the end, we compute the negative log likelihood of the N-dimensional Gaussian distribution as the loss function to supervise training.

1, we first concatenate $\boldsymbol{\mu}_\theta^{i+1}(\mathbf{I})$ and $\mathbf{F}^{i+1}$ from the previous stage, and then feed into a stack of convolution layers to refine the depth map and feature map. The refined depth map is upsampled via bi-linear interpolation to double the resolution, and denoted as $\boldsymbol{\mu}_\theta^{i}(\mathbf{I})$. Similarly, the refined feature map is upsampled and added to $\mathbf{F}^i$. In the end, we upsample all the depth maps $\{\boldsymbol{\mu}_\theta^{i}(\mathbf{I})\}_{i=1}^4$ into $m \times n$ resolution via bi-linear interpolation as the final output of the U-decoder.

The *K-decoder* (see Fig. 2.5) estimates $\Psi_\theta(\mathbf{I})$. It first upsamples and refines the feature maps in $\mathbf{F}$. Specifically, at the i-th stage, where i decreases from 3 to 1, it upsamples $\mathbf{F}^{i+1}$ from the previous stage and adds to the $\mathbf{F}^i$. We utilize a stack of convolution layers to further refine the added feature map. In the end, we upsample the refined feature map $\mathbf{F}^1$ to $m \times n$ resolution by bi-linear interpolation, and predict the $\Psi_\theta(\mathbf{I})$ by a convolution layer.

***(iii)* Train and Test Time Setting.** At train time, we collect $\{\boldsymbol{\mu}_\theta^{i}(\mathbf{I})\}_{i=1}^4$, and $\Psi_\theta(\mathbf{I})$ and compute loss using our proposed loss function (refer to Sec. 2.3.4). At test time, our approach provides $\boldsymbol{\mu}_\theta^{1}(\mathbf{I})$ as the final depth map prediction. Furthermore, we can query $\Psi_\theta(\mathbf{I})$ to infer the distribution of the depth map if necessary depending on the application.

2.3.4 *Loss Function*

As shown in Sec. 2.3.2.1, the negative log likelihood loss can approximate the scale invariant loss and the gradient loss when $\Psi_\theta(I)$ and σ take special values. Consequently, we propose the following overall loss function:

$$\mathcal{L}_{total} = \sum_{j=1}^{4} \mathcal{L}_{NLL}^{j} + \frac{1}{N}\sum_{i}(\boldsymbol{\mu}_\theta^{1}(I)_i - Z_i^{gt})^2 \tag{2.11}$$

where $\mathcal{L}_{NLL}^{j}$ is the negative log likelihood loss applying to $\boldsymbol{\mu}_\theta^{j}(I)$ and $\Psi_\theta(I)$. Note, however, Eq.(2.11) second term is optional. Yet, it is added to provide train time improvement.

2.4 EXPERIMENTS AND RESULTS

IMPLEMENTATION DETAILS. We implemented our framework in PyTorch 1.7.1 and Python 3.8 with CUDA 11.0. All the experiments

Method	Backbone	SILog ↓	Abs Rel ↓	RMS↓	RMS log↓	δ_1 ↑
GeoNet[181]	ResNet-50	-	0.128	0.569	-	0.834
DORN [58]	ResNet-101	-	0.115	0.509	-	0.828
VNL[253]	ResNeXt-101	-	0.108	0.416	-	0.875
TransDepth [248]	ViT-B	-	0.106	0.365	-	0.900
ASN [154]	HRNet-48	-	0.101	0.377	-	0.890
BTS [124]	DenseNet-161	11.533	0.110	0.392	0.142	0.885
DPT-Hybrid [186]	ViT-B	9.521	0.110	0.357	0.129	0.904
AdaBins [15]	EffNet-B5+ViT-mini	10.570	0.103	0.364	0.131	0.903
ASTrans [28]	ViT-B	10.429	0.103	0.374	0.132	0.902
NeWCRFs [256]	Swin-L	9.102	0.095	0.331	0.119	0.922
Ours	Swin-L	**8.323**	**0.087**	**0.311**	**0.110**	**0.933**
% Improvement		8.56%	8.42%	6.04%	7.56%	1.18%

Table 2.1: Comparison with the state-of-the-art methods on the NYU test set [206]. Please refer to Sec. 2.4.1 for details.

Method	SILog↓	Abs Rel↓	Sq Rel↓	iRMS↓
DLE [142]	11.81	9.09	2.22	12.49
DORN [58]	11.80	8.93	2.19	13.22
BTS [124]	11.67	9.04	2.21	12.23
BANet [5]	11.55	9.34	2.31	12.17
PWA [125]	11.45	9.05	2.30	12.32
ViP-DeepLab [182]	10.80	8.94	2.19	11.77
NeWCRFs [256]	10.39	8.37	1.83	11.03
Ours	**9.93**	**7.99**	**1.68**	**10.63**
% Improvement	4.43%	4.54%	8.20%	3.63%

Table 2.2: Comparison with the state-of-the-art methods on the the KITTI official test set [64]. We only list the results from the published methods.

and statistical results shown in the draft are simulated on a computing machine with Quadro RTX 6000 (24GB Memory) GPU support. We use evaluation metrics including SILog, Abs Rel, RMS, RMS log, δ_i, Sq Rel, iRMS to report our results on the benchmark dataset. For exact definition of the metrics we refer to [124].

Datasets. We performed experiments and statistical comparisons with the prior art on benchmark datasets such as NYU Depth V2 [206], KITTI [64], and SUN RGB-D [210].

(a) NYU Depth V2: This dataset contains images of indoor scenes with 480×640 resolution [206]. We follow the standard train and test split setting used by previous works for experiments [124]. Precisely, we use $24,231$ image-depth pairs for training the network and 654 images

Method	Backbone	SILog ↓	Abs Rel ↓	RMS↓	RMS log↓	δ_1 ↑
DORN [58]	ResNet-101	-	0.072	0.273	0.120	0.932
VNL [253]	ResNeXt-101	-	0.072	0.326	0.117	0.938
TransDepth [248]	ViT-B	8.930	0.064	0.275	0.098	0.956
BTS [124]	DenseNet-161	8.933	0.060	0.280	0.096	0.955
DPT-Hybrid [186]	ViT-B	8.282	0.062	0.257	0.092	0.959
AdaBins [15]	EffNet-B5+ViT-mini	8.022	0.058	0.236	0.089	0.964
ASTrans [28]	ViT-B	7.897	0.058	0.269	0.089	0.963
NeWCRFs [256]	Swin-L	6.986	0.052	0.213	0.079	0.974
Ours	Swin-L	**6.757**	**0.050**	**0.202**	**0.075**	**0.976**
% Improvement		3.28%	3.85%	5.16%	5.06%	0.21%

Table 2.3: Comparison with the state-of-the-art methods on the KITTI Eigen test set [55]. Please refer to Sec. 2.4.1 for details.

for testing the performance. Note that the depth map evaluation for this dataset has an upper bound of 10 meters.

(b) KITTI: This dataset contains images and depth data of outdoor driving scenarios. The official experimental split contains 42,949 training images, 1,000 validation images, and 500 test images with 352×1216 resolution [64]. Here, the depth map accuracy can be evaluated up to an upper bound of 80 meters. In addition, there are few works following the split from Eigen [55], which includes 23,488 images for training and 697 images for the test.

(c) SUN RGB-D: It contains data of indoor scenes captured by different cameras [210]. The depth values range from 0 up to 10 meters. The images are resized to 480×640 resolution for consistency. We use the official test set [210] of 5050 images to evaluate the generalization of the frameworks.

Method	Backbone	SILog ↓	Abs Rel ↓	RMS↓	RMS log↓	δ_1 ↑
AdaBins[15]	EffNet-B5+ViT-mini	13.652	0.110	0.321	0.137	0.906
NeWCRFs [256]	Swin-L	13.695	0.105	0.322	0.138	0.920
Ours	Swin-L	**11.985**	**0.090**	**0.282**	**0.120**	**0.936**
% Improvement		12.49%	14.29%	12.42%	13.04%	1.74%

Table 2.4: Comparison with AdaBins [15] and NeWCRFs [256] on SUN RGB-D test set [210]. All methods are trained on NYU Depth V2 train set without fine-tuning on SUN RGB-D. Please refer to Sec. 2.4.1 for details.

Training Details. We use Adam optimizer [106] to minimize our proposed loss function and learn network parameters. At train time, the learning rate is decreased from $3e^{-5}$ to $1e^{-5}$ by the cosine annealing scheduler. Our encoder–which is inspired from [152], is initialized by pre-training the network on ImageNet [46]. For the KITTI dataset, we train our framework for 10 and 20 epochs on the official split [64] and Eigen [55] split, respectively. For the NYU dataset [206], our framework is trained for 20 epochs. We randomly apply horizontal flipping on the image and depth map pair at train time for data augmentation.

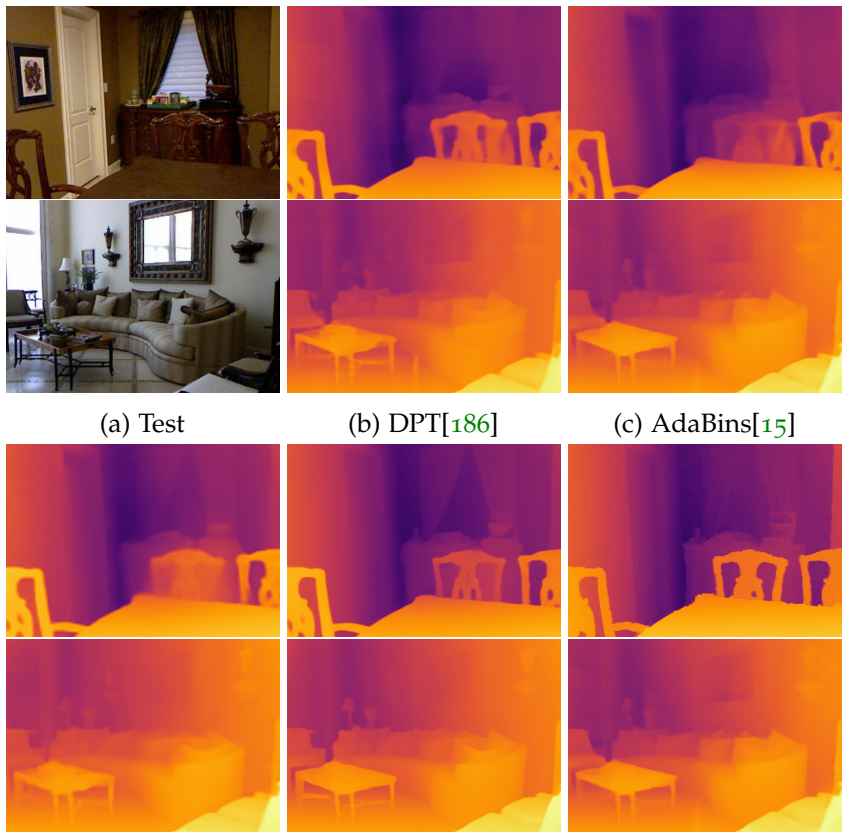
2.4.1 *Performance Comparison with Prior Works*

Tab. 2.1, 2.2, 2.3, and 2.4 show our method's statistical performance comparison with popular state-of-the-art (SOTA) methods on NYU Depth V2 [206], KITTI official [64] and Eigen [55] split, and SUN RGB-D [210]. From the tables, it is easy to infer that our approach consistently performs better on all the popular evaluation metrics. The percentage improvement over the previous SOTA is indicated in green for better exposition. In particular, on the NYU test set, which is a challenging dataset, we reduce the SILog error from the previous best result of 9.102 to 8.323 and increase δ_1 metric from 0.922 to 0.933[3]. Fig. 2.6 shows qualitative comparison results. It can be observed that our method's predicted depth is better at low and high-frequency scene details. For the SUN RGB-D test set, all competing models, including ours, are trained on the NYU DepthV2 train set without fine-tuning on SUN RGB-D [210]. In addition, we align the predictions from all the models with the ground truth by a scale and shift following [187]. Tab. 2.4 results indicate our method's better generalization capability than other approaches.

2.4.2 *Bayesian Uncertainty Estimation Comparison*

In this part, we compare with the classical Bayesian dropout [101], which uses independent Gaussian distribution to quantify uncertainty. As for our approach, we also use dropout to sample multiple depth predictions, and compute the negative log likelihood following the dis-

3 At the time of submission, our method's performance on the KITTI official leaderboard was the best among all the published works.



(a) Test (b) DPT[186] (c) AdaBins[15]

(d) NeWCRFs[256] (e) **Ours** (f) Ground Truth

Figure 2.6: **Qualitative Comparison**. Our method recovers better depth even for complex scenes. Our depth results even qualitatively looks closer to the ground truth than the prior art such as (b) DPT [186], (c) AdaBins [15], (d) NeWCRFs [256] on NYU Depth V2 test set [206].

tribution in Eq.(2.9). More specifically, in each block of Swin transformer [152], we randomly drop feature channels before the layer normalization [7] operation with probability 0.01. We first sample $S = 10$ predictions for each test image, then compute the mean and covariance of the mixture of Gaussian distributions in Eq.(2.9), and further approximate the entire distribution as single Gaussian following [117]. We present the comparison results of the negative log likelihood in Fig. 2.7. Our multivariate Gaussian distribution achieves much lower negative log likelihood cost.

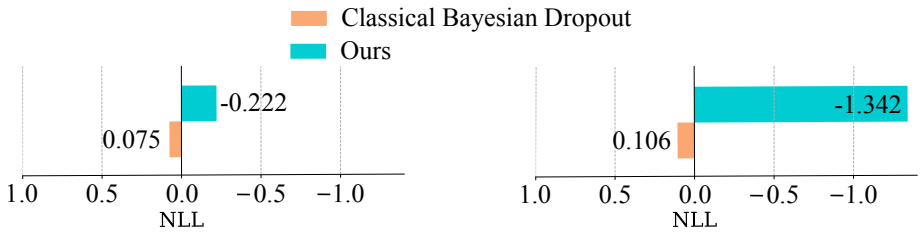


Figure 2.7: Comparison with the classical Bayesian dropout for uncertainty estimation. The left and right figures present the Negative Log Likelihood (NLL) of the predicted depth map distribution on KITTI Eigen [64, 55] split and NYU test set [206] respectively. Our multivariate Gaussian distribution achieves lower NLL than the independent Gaussian distribution in classical Bayesian dropout.

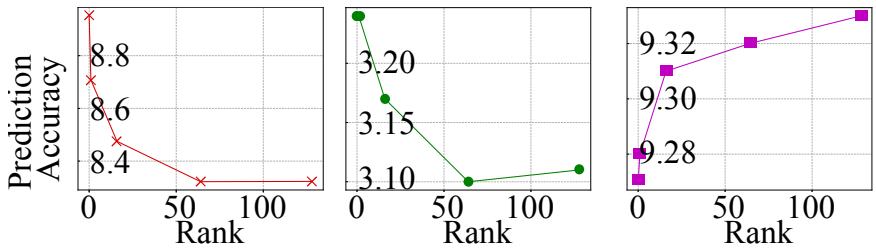
2.4.3 *Ablations and Further Analysis*

To better understand our introduced approach, we performed ablations on the NYU Depth V2 dataset [206] and studied our trained model's inference time and memory footprint for its practical suitability.

***(i)* Effect of NLL Loss.** To realize the significance of NLL loss in Eq.2.11, we replaced it with $L2$ loss, SI loss [55], gradient loss, and virtual normal loss [253] one by one, while keeping the remaining term in Eq.2.11 fixed. The statistical results are shown in Tab.2.5. The stats show that our proposed NLL loss achieves the best performance over all the widely used metrics.

Loss	SILog ↓	Abs Rel ↓	RMS ↓	δ_1 ↑
L2	8.912	0.090	0.324	0.929
SI [55]	8.762	0.089	0.322	0.929
Gradient	8.886	0.090	0.323	0.929
VNL [253]	8.543	0.090	0.325	0.926
Ours	**8.323**	**0.087**	**0.311**	**0.933**

Table 2.5: Comparison of our NLL loss function with widely used loss functions for solving SIDP task.



(a) SILog Metric (b) RMS Metric ×10 (c) δ_1 Metric×10

Figure 2.8: Depth prediction accuracy of our method using different evaluation metrics w.r.t change in the rank of the covariance matrix. The increase in the rank improves prediction accuracy and shows saturation at 128, thereby showing the effectiveness of our low-dimensional modeling.

***(ii)* Performance with the change in the Rank of Covariance.** We vary the rank of $\Psi_\theta(I)$, and observe our method's prediction accuracy. We present the accuracy under various evaluation metrics in Fig. 2.8. With increase in the rank, the distribution is better approximated, and the performance improves, but saturates later showing the suitability of its low-dimensional representation.

***(iii)* Evaluation on NeWCRFs.** We evaluate our loss function on NeWCRFs [256] network design using their proposed training strategies. The depth prediction accuracy is shown in Tab. 2.6. The results convincingly indicate the benefit of our proposed loss on a different SIDP network.

***(iv)* Inference Time & Parameter Comparison.** We compared our method's inference time, and the number of model parameters to the recent state-of-the-art NeWCRFs [256]. The inference time is measured

Method	SILog ↓	Abs Rel ↓	RMS ↓	δ_1 ↑
NeWCRFs	9.102	0.095	0.331	0.922
+Our Loss	**8.619**	**0.086**	**0.316**	**0.935**

Table 2.6: Results using our loss on [256] network on NYU Depth.

on the NYU Depth V2 test set [206] with batch size 1. As shown in Tab.2.7, our method achieves lower scale invariant logarithmic error (SILog) with fewer network parameters and comparable FPS. Such a statistical performance further endorse our approach's practical adequacy.

Method	SILog↓	Speed (FPS)↑	Param (M)↓
NeWCRFs	9.171	**10.551**	258
Ours	**8.323**	9.909	**244**

Table 2.7: Comparison of the inference time and parameters with NeWCRFs [256] on NYU Depth V2 [206].

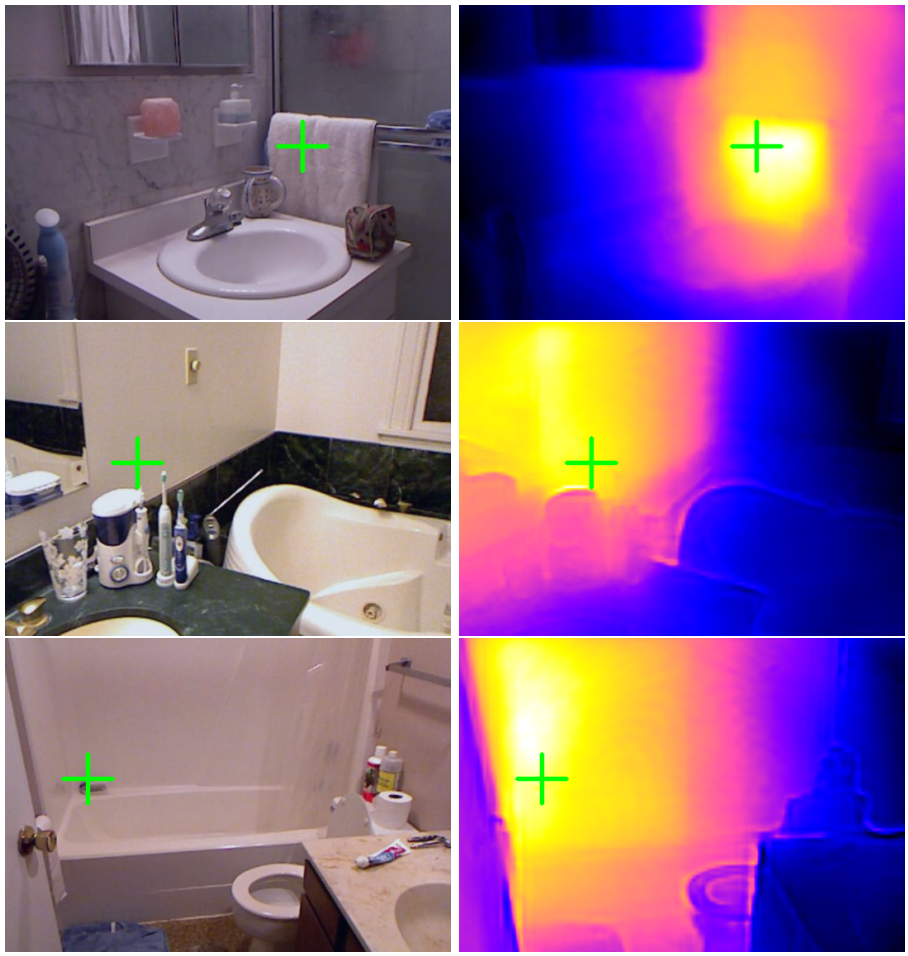
2.4.4 *Visualization of Learned Covariance*

To understand the covariance learned by the proposed negative log likelihood loss function, we visualize the covariance for selected pixels. More specifically, for each image we select a pixel (marked as a green cross), and visualize the covariance between the pixel and all other pixels. The results are shown in Fig. 2.9. We observe that the pixels from nearby regions or the same objects usually have higher covariance.

2.5 CONCLUSION

This work suitably formalizes the connection between robust statistical modeling techniques, *i.e.*, multivariate covariance modeling with low-rank approximation, and popular loss functions in neural network-based SIDP problem. The novelty presented in this chapter arises from the fact that the proposed pipeline and loss term turns out to be more general, hence could be helpful in the broader application of SIDP in several tasks, such as depth uncertainty for robot vision, control and others. Remarkably, the proposed formulation is not only theoretically

compelling but observed to be practically beneficial, resulting in a loss function that is used to train the proposed network showing state-of-the-art SIDP results on several benchmark datasets.



(a) (b)

Figure 2.9: **Visualization of Covariance**. Left: test image. Right: covariance with respect to the pixel which is marked as a green cross. The yellow and light regions have higher covariance than the blue and dark ones.

3 VARIATIONAL CONSTRAINT

This chapter is based on our paper: Ce Liu et al. "VA-DepthNet: A Variational Approach to Single Image Depth Prediction." In: *The Eleventh International Conference on Learning Representations*. 2022.

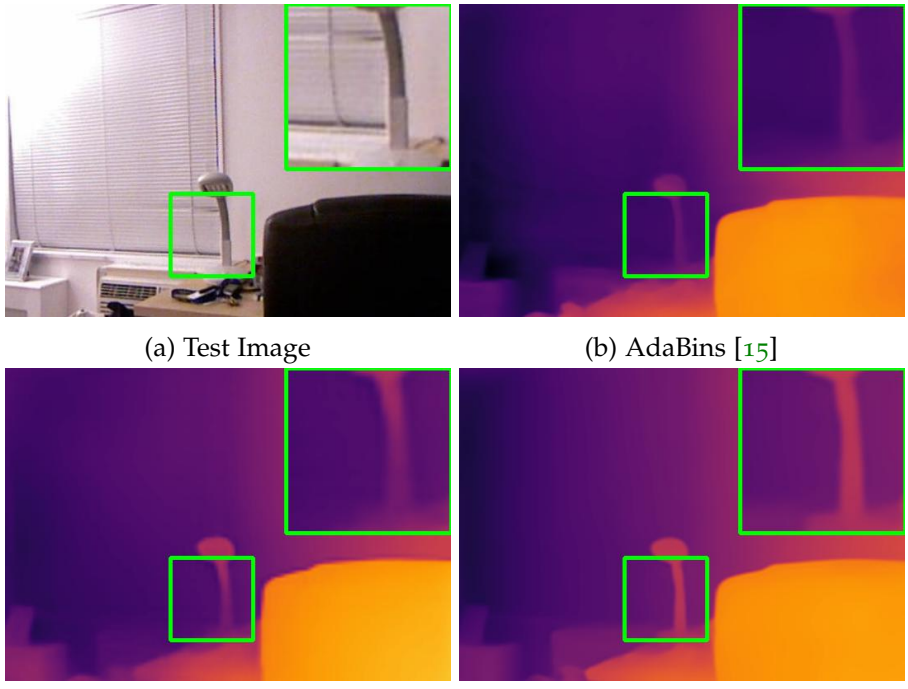
3.1 INTRODUCTION

Over the last decade, neural networks have introduced a new prospect for the 3D computer vision field. It has led to significant progress on many long-standing problems in this field, such as multi-view stereo [92], [100], visual simultaneous localization and mapping [218], novel view synthesis [165], *etc.*Among several 3D vision problems, one of the challenging, if not impossible, to solve is the single-image depth prediction (SIDP) problem. SIDP is indeed ill-posed—in a strict geometric sense, presenting an extraordinary challenge to solve this inverse problem reliably. Moreover, since we do not have access to multi-view images, it is hard to constrain this problem via well-known geometric constraints [155, 173, 60, 115, 114]. Accordingly, the SIDP problem generally boils down to an ambitious fitting problem, to which deep learning provides a suitable way to predict an acceptable solution to this problem [256, 253].

Impressive earlier methods use Markov Random Fields (MRF) to model monocular cues and the relation between several over-segmented image parts [198, 199]. Nevertheless, with the recent surge in neural network architectures [112, 208, 78], which has an extraordinary capability to perform complex regression, many current works use deep learning to solve SIDP and have demonstrated high-quality results [256, 5, 15, 55, 58, 124, 125]. Popular recent methods for SIDP are mostly supervised. But even then, they are used less in real-world applications than geometric multiple view methods [116, 168]. Nonetheless, a good solution to SIDP is highly desirable in robotics [250], virtual-reality [86], augmented reality [52], view synthesis [86] and other related vision tasks [150].

In this chapter, we advocate that despite the supervised approach being encouraging, SIDP advancement should not wholly rely on the increase of dataset sizes. Instead, geometric cues and scene priors could help improve the SIDP results. Not that scene priors have not been studied to improve SIDP accuracy in the past. For instance, [33] uses pairwise ordinal relations between points to learn scene depth. Alternatively, [253] uses surface normals as an auxiliary loss to improve performance. Other heuristic approaches, such as [181], jointly exploit the depth-to-normal relation to recover scene depth and surface normals. Yet, such state-of-the-art SIDP methods have limitations: for example, the approach in [33] - using ordinal relation to learn depth - over-smooths the depth prediction results, thereby failing to preserve high-frequency surface details. Conversely, [253] relies on good depth map prediction from a deep network and the idea of virtual normal. The latter is computed by randomly sampling three non-collinear points with large distances. This is rather complex and heuristic in nature. [181] uses depth and normal consistency, which is good, yet it requires good depth map initialization.

This brings us to the point that further generalization of the regression-based SIDP pipeline is required. As mentioned before, existing approaches in this direction have limitations and are complex. In this chapter, we propose a simple approach that provides better depth accuracy and generalizes well across different scenes. To this end, we resort to the physics of variation [166, 25] in the neural network design for better generalization of the SIDP network, which by the way, keeps the essence of affine invariance [253]. An image of a general scene—indoor or outdoor, has a lot of spatial regularity. And therefore, introducing a variational constraint provides a convenient way to ensure spatial regularity and to preserve information related to the scene discontinuities [25]. Consequently, the proposed network is trained in a fully-supervised manner while encouraging the network to be mindful of the scene regularity where the variation in the depth is large. In simple terms, depth regression must be more than parameter fitting, and at some point, a mindful decision must be made—either by imaging features or by scene depth variation, or both. As we demonstrate later in the chapter, such an idea boosts the network's depth accuracy while preserving the high-frequency and low-frequency scene information (see Fig.3.1).



(a) Test Image (b) AdaBins [15]

(c) NeWCRFs [256] (d) **Ours**

Figure 3.1: Qualitative comparison of our method's depth result with recent state-of-the-art methods such as AdaBins [15], NeWCRFs [256] on NYU Depth V2 test set [206]. It can be observed that our method predicts high-frequency details better than other recent methods.

Our neural network for SIDP disentangles the absolute scale from the metric depth map. It models an unscaled depth map as the optimal solution to the pixel-level variational constraints via weighted first-order differences, respecting the neighboring pixel depth gradients. Compared to previous methods, the network's task has been shifted away from pixel-wise metric depth learning to learning the first-order differences of the scene, which alleviates the scale ambiguity and favors scene regularity. To realize that, we initially employ a neural network to predict the first-order differences of the depth map. Then, we construct the partial differential equations representing the variational constraints by reorganizing the differences into a large matrix, *i.e.*, an over-determined system of equations. Further, the network learns a weight matrix to eliminate redundant equations that do not favor the introduced first-order difference constraint. Finally, the closed-form depth map solution is recovered via simple matrix operations.

When tested on the KITTI [64] and NYU Depth V2 [206] test sets, our method outperforms prior art depth prediction accuracy by a large margin. Moreover, our model pre-trained on NYU Depth V2 better generalizes to the SUN RGB-D test set.

3.2 PRIOR WORK

Depth estimation is a longstanding task in computer vision. In this work, we focus on a fully-supervised, single-image approach, and therefore, we discuss prior art that directly relates to such approach. Broadly, we divide the popular supervised SIDP methods into three sub-categories.

***(i)* Depth Learning using Ranking or Ordinal Relation Constraint.** [270] and [33] argue that the ordinal relation between points is easier to learn than the metric depth. To this end, [270] proposes constrained quadratic optimization while [33] relies on the variation of the inception module to solve the problem. Later, [239] proposes structure-guided sampling strategies for point pairs to improve training efficiency. Recently, [135] elaborates on the use of listwise ranking method based on the Plackett-Luce model [158]. The drawback of such approaches is that the ordinal relationship and ranking over smooth the depth solution making accurate metric depth recovery challenging.

***(ii)* Depth Learning using Surface Normal Constraint.** [90] introduces normal loss in addition to the depth loss to overcome the distorted and

blurry edges in the depth prediction. [253] proposes the concept of virtual normal to impose 3D scene constraint explicitly and to capture the long-range relations in the depth prediction. The long-range dependency in 3D is introduced via random sampling of three non-colinear points at a large distance from the virtual plane. Lately, [154] proposes an adaptive strategy to compute the local patch surface normals at train time from a set of randomly sampled candidates and overlooks it during test time.

***(iii)* Depth Learning using other Heuristic Refinement Constraint.** There has been numerous works attempt to refine the depth prediction as a post-processing step. [147], [127] and [256] propose to utilize the Conditional Random Fields (CRF) to smooth the depth map. [124] utilizes the planar assumptions to regularize the predicted depth map. [181] adopts an auxiliary network to predict the surface normal, and then refine the predicted depth map following their proposed heuristic rules. There are mainly two problems with such approaches: Firstly, these approaches rely on a good depth map initialization. Secondly, the heuristic rules and the assumptions might result in over-smoothed depth values at objects boundaries.

Meanwhile, a few works, such as [185, 36, 129] were proposed in the past with similar inspirations. [185, 36] methods are generally motivated towards depth map refinement predicted from an off-the-shelf network. On the other hand, [36] proposes to use an affinity matrix that aims to learn the relation between each pixel's depth value and its neighbors' depth values. However, the affinity matrix has no explicit supervision, which could lead to imprecise learning of neighboring relations providing inferior results. On the contrary, our approach is mindful of imposing the first-order difference constraint leading to better performance. Earlier, [129] proposed two strategies for SIDP, i.e., fusion in an end-to-end network and fusion via optimization. The end-to-end strategy fuses the gradient and the depth map via convolution layers without any constraint on convolution weights, which may not be an apt choice for a depth regression problem such as SIDP. On the other hand, the fusion via optimization strategy is based on a non-differentiable strategy, leading to a non-end-to-end network loss function. Contrary to that, our method is well-constrained and performs quite well with a loss function that helps end-to-end learning of our proposed network. Not long ago, [122] proposed to estimate relative depths between pairs of images and ordinary depths at a different

scale. By exploiting the rank-1 property of the pairwise comparison matrix, it recovers the relative depth map. Later, relative and ordinary depths are decomposed and fused to recover the depth. On a slightly different note, [123] studies the effectiveness of various losses and how to combine them for better monocular depth prediction.

To sum up, our approach allows learning of confidence weight to select reliable gradient estimation in a fully differentiable manner. Further, it proffers the benefits of the variational approach to overcome the limitations of the existing state-of-the-art methods. More importantly, the proposed method can provide excellent depth prediction without making extra assumptions such as good depth initialization, piece-wise planar scene, and assumptions used by previous works mentioned above.

3.3 METHOD

In this section, we first describe our proposed variational constraint and then present the overall network architecture leading to the overall loss function.

3.3.1 *Variational Constraint*

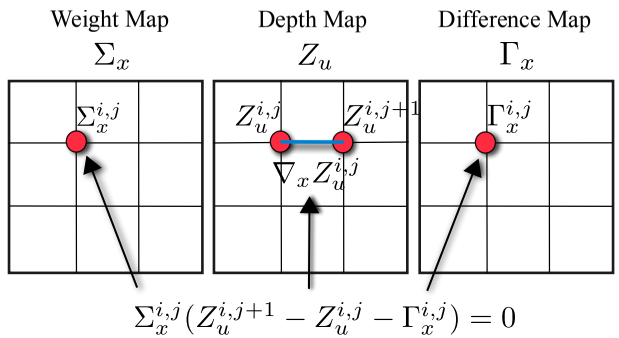
Here we introduce our variational constraint and how it can be useful for depth estimation. Consider an unscaled depth map as $Z_u \in \mathbb{R}^{H\times W}$, with (H, W) symbolizing the height and width, respectively. Assuming $\Gamma_x \in \mathbb{R}^{H\times W}$ and $\Gamma_y \in \mathbb{R}^{H\times W}$ as the gradient of Z_u in the x and y axis, we write

$$\nabla Z_u = [\Gamma_x,\ \Gamma_y]^{\mathrm{T}}. \tag{3.1}$$

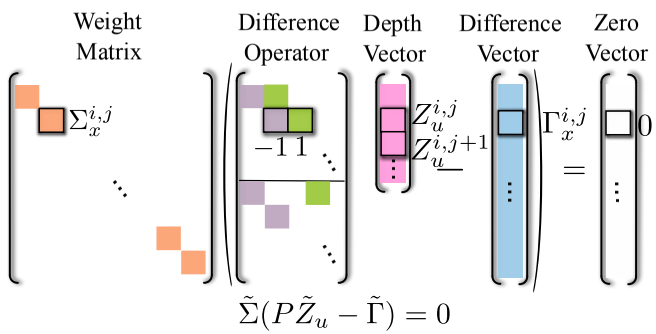
Here, x and y subscript corresponds to the direction from left to right (x-axis) and top to bottom of the image (y-axis), respectively. Elaborating on this, we can write

$$\Gamma_x^{i,j} = \nabla_x Z_u^{i,j} = Z_u^{i,j+1} - Z_u^{i,j};\ \ \Gamma_y^{i,j} = \nabla_y Z_u^{i,j} = Z_u^{i+1,j} - Z_u^{i,j}. \tag{3.2}$$

Suppose we augment Eq.(3.2) expression for all (i, j), $i \in \{1, ..., H\}$ and $j \in \{1, ..., W\}$. In that case, we will end up with an over-determined system with $2HW$ equations in total. Given the predicted Γ_x and Γ_y, we aim to recover the HW unknown variables in Z_u.



(a) First order depth variation along x-axis.



(b) Overall matrix by ordering the terms.

Figure 3.2: **Illustration of the idea**. (a) Depth gradient constraint along x axis at location (i, j) in 4×4 matrix form. (b) Construction of the overall matrix formulation with constraints at all the pixel locations.

However, some of the equations could be spurious and deteriorate the overall depth estimation result rather than improving it. As a result, we must be mindful about selecting the equation that respects the imposed first-order constraint and maintains the depth gradient to have a meaningful fitting for better generalization. To that end, we introduce confidence weight $\Sigma_x \in [0,1]^{H\times W}$, $\Sigma_y \in [0,1]^{H\times W}$ for gradient along x, y direction. Consequently, we multiply the above two equations by the confidence weight term $\Sigma_x^{i,j}$ and $\Sigma_y^{i,j}$, respectively. On one hand, if the confidence is close to 1, the equation will have priority to be satisfied by the optimal Z_u. On the other hand, if the confidence is close to 0, we must ignore the equation. For better understanding, we illustrate the first-order difference and weighted matrix construction in Fig. 3.2 (a) and Fig. 3.2(b).

Next, we reshape the Σ_x, Σ_y, Γ_x, Γ_y, and Z_u into column vectors $\tilde{\Sigma}_x \in [0,1]^{HW\times 1}$, $\tilde{\Sigma}_y \in [0,1]^{HW\times 1}$, $\tilde{\Gamma}_x \in \mathbb{R}^{HW\times 1}$, $\tilde{\Gamma}_y \in \mathbb{R}^{HW\times 1}$, and $\tilde{Z}_u \in \mathbb{R}^{HW\times 1}$, respectively. Organizing $\tilde{\Sigma} = \mathbf{diag}([\tilde{\Sigma}_x; \tilde{\Sigma}_y]) \in \mathbb{R}^{2HW\times 2HW}$ and $\tilde{\Gamma} = \mathbf{concat}[\tilde{\Gamma}_x; \tilde{\Gamma}_y] \in \mathbb{R}^{2HW\times 1}$, we can write the overall expression in a compact matrix form using simple algebra as follows

$$\tilde{\Sigma} P \tilde{Z}_u = \tilde{\Sigma}\tilde{\Gamma} \tag{3.3}$$

where $P \in \{1, 0, -1\}^{2HW\times HW}$ is the first-order difference operator. Specifically, P is a sparse matrix with only a few elements as 1 or -1. The i^{th} row of P provides the first-order difference operator for the i^{th} equation. The position of 1 and -1 indicates which pair of neighbors to be considered for the constraint. Fig.3.2 (b) provides a visual intuition about this matrix equation.

Eq.(3.3) can be utilized to recover Z_u from the predicted Γ_x, Γ_y, Σ_x, and Σ_y. As alluded to above, we have more equations than unknowns, hence, we resort to recovering the optimal depth map $\tilde{Z}_u^* \in \mathbb{R}^{HW\times 1}$ by minimizing the following equation:

$$\tilde{Z}_u^* = \arg\min_{\tilde{Z}_u} ||\tilde{\Sigma}(P\tilde{Z}_u - \tilde{\Gamma})||_2. \tag{3.4}$$

The closed-form solution can be written as follows:

$$\tilde{Z}_u^* = \overbrace{(P^T\tilde{\Sigma}^2 P)^{-1} P^T \tilde{\Sigma}^2}^{K_{\tilde{\Sigma}}} \tilde{\Gamma}. \tag{3.5}$$

Denote $K_{\tilde{\Sigma}} \triangleq (P^T\tilde{\Sigma}^2 P)^{-1} P^T \tilde{\Sigma}^2$ in Eq.(3.5), we write overall equation as $\tilde{Z}_u^* = K_{\tilde{\Sigma}}\tilde{\Gamma}$. Next, we describe the overall network architecture.

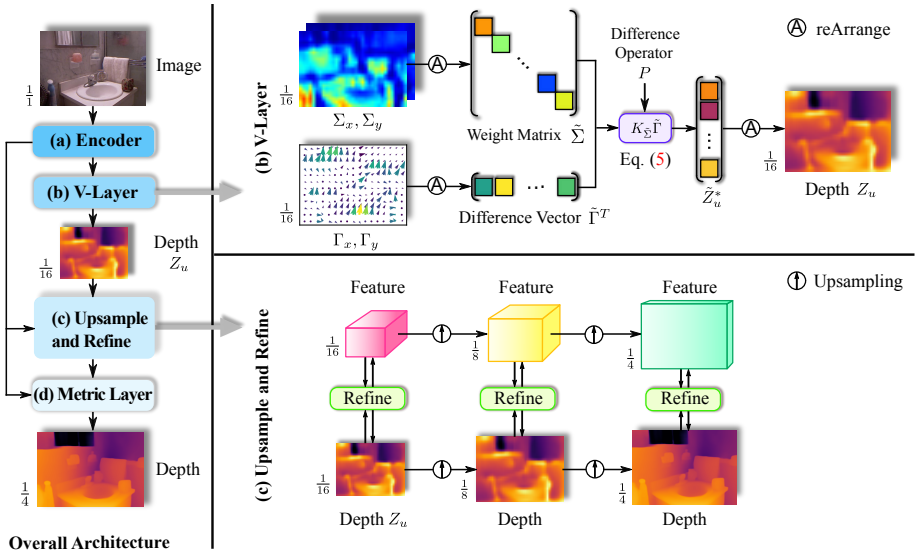


Figure 3.3: **Overview of our framework.** Given an input image, first an encoder is employed to extract features. Then we predict the depth map by the V-layer. Next, we gradually upsample and refine the depth map. In the end, we recover the metric depth by the metric layer.

3.3.2 *Overall Network Architecture*

Our overall network architecture is composed of four main modules as follows.

***(a)* Encoder.** Given an input image, the encoder computes the hierarchical feature maps through a series of stages. To be precise, our encoder has four stages. Each stage contains transformer blocks [152]. At the end of each stage, we collect the final feature map as the output of the encoder resulting in the encoded feature maps with strides 4, 8, 16, and 32, respectively. Our encoder module is inspired by [152], a recent state-of-the-art transformer network design. We use it as our backbone by removing the final global pooling layer and fully connected layer.

***(b)* Variational Layer (V-Layer).** The goal of this layer is to compute a map from encoded feature maps to unscaled depth map, which adheres to the first-order variational constraint. As of V-layer, we feed the feature maps of strides 16 and 32 as input which is the output of the encoder. Since these features are at different resolutions, we upsample the feature map of stride 32 to stride 16 via bi-linear interpolation and

concatenate to construct $I_\Phi \in \mathbb{R}^{C\times H\times W}$, where (H, W, C) symbolizing the height, width, and the number of channels, respectively.

Note that H, W is not the same as the original resolution of the ground-truth depth and images. We use of two convolutional layers on I_Φ to predict the depth gradient and corresponding weight for each pixel as follows:

$$\{\Gamma_x, \Gamma_y, \Sigma_x, \Sigma_y\} = f(I_\Phi; \theta) \tag{3.6}$$

where, $f(I_\Phi; \theta)$ denotes the convolutional layers with parameters θ. The predicted depth gradients Γ_x and Γ_y are observed to be more accurate at smooth surface than at boundaries. This brings us again to the point made above that we must take care of which first-order constraint must be included and discarded during regression. Using the Eq.(3.6) prediction, we construct the variational constraint Eq.(3.3), and obtain the unscaled depth map following Eq.(3.5). The resulting depth map has a resolution of $1/16$ to the original image, which is later upsampled to the appropriate resolution.

To capture more scene features, we generate multiple channels (denoted as S) of $\{\Gamma_x, \Gamma_y, \Sigma_x, \Sigma_y\}$ using Eq.(3.6). As a result, we have a group of depth maps stacked along the channel dimension. For a feature map with spatial resolution $H \times W$, our V-layer has a complexity of $O(H^3W^3)$. To overcome complexity issue, we perform V-layer operation on feature maps with stride 16 and then upsample and refine the depth maps in the later stage. The V-layer pipeline is shown in Fig. 3.3(b).

***(c)* Upsample and Refine.** This module upsamples and refines the input depth map via encoded features at a given depth map resolution. To this end, we perform refinement at three different resolutions in a hierarchical manner. Given the V-layer depth map at $1/16$ resolution, we first refine the depth via encoded features at this resolution. Concretely, this refinement is done using the following set of operations. (1) concatenate the feature map and the depth map; (2) use one convolutional layer with ReLU activation to fuse the feature and depth information; and (3) predict refined feature and depth map via a convolutional layer. Later, the refined feature and depth map are upsampled and fed into $1/8$ for later refinement using the same above set of operations. Finally, the exact is done at $1/4$ resolution. Note that these steps are performed in a sequel. At the end of this module, we have a depth map of $1/4$ of the actual resolution. The upsample and refine procedure are shown in Fig. 3.3(c).

***(d)* Metric Layer.** We must infer the global scene scale and shift to recover the metric depth. For this, we perform global max pooling on the encoded feature map of stride 32. The resulting vector is fed into a stack of fully connected layers to regress the two scalars, *i.e.*, one representing the scale and while other representing the shift. Using the feature map of stride 32 is motivated by the observation that we have a much richer global scene context using it than at higher depth resolution. It also provides a good compromise between computational complexity and accuracy.

3.3.3 *Loss Function*

Depth Loss. It estimates the scale-invariant difference between the ground-truth depth and prediction at train time [55]. The difference is computed by upsampling the predicted depth map to the same resolution as the ground truth via bi-linear interpolation. Denoting the predicted and ground-truth depth as $\hat{Z} \in \mathbb{R}^{m \times n}$, $Z_{gt} \in \mathbb{R}^{m \times n}$ we compute the depth loss as follows

$$\mathcal{L}_{depth}(\hat{Z}, Z_{gt}) = \frac{1}{N}\sum_{(i,j)}(e^{i,j})^2 - \frac{\alpha}{N^2}(\sum_{(i,j)} e^{i,j})^2, \text{ where, } e^{i,j} = \log \hat{Z}^{i,j} - \log Z^{i,j}_{gt}. \tag{3.7}$$

Here, N is the number of positions with valid measurements and $\alpha \in [0, 1]$ is a hyper-parameter. Note that the above loss is used for valid measurements only.

Variational Loss. We define this loss using the output of V-layer. Suppose the ground-truth depth map to be $Z_{gt} \in \mathbb{R}^{m \times n}$ and the predicted depth map for S channels as $Z_u \in \mathbb{R}^{S \times H \times W}$. Since the depth resolution is not same at this layer, we downsample the ground truth. It is observed via empirical study that low-resolution depth map in fact help capture the first-order variational loss among distant neighbors. Accordingly, we downsample the Z_{gt} instead of upsamping Z_u. We downsample Z_{gt} denoted as $Q_{gt} \in \mathbb{R}^{H \times W}$ by random pooling operation, i.e., we randomly select a location where we have a valid measurement since ground-truth data may have pixels with no depth values. The coordinates of selected location in $Z_{gt} \mapsto Z_u \in \mathbb{R}^{S \times H \times W}$ and the

corresponding depth value is put in $\hat{Q} \in \mathbb{R}^{S \times H \times W}$ via bi-linear interpolation. We compute the variational loss as

$$\mathcal{L}_{var}(\hat{Q}, Q_{gt}) = \frac{1}{N'} \sum_{(i,j)} |\text{Conv}(\hat{Q})^{i,j} - \nabla Q_{gt}^{i,j}| \tag{3.8}$$

where N' is the number of positions having valid measurements, ∇ symbolises the first-order difference operator, and Conv refers to the convolutional layer. Here, we use the Conv layer to fuse S depth maps into a single depth map and also to compute its horizontal and vertical gradient.

Total Loss. We define the total loss as the sum of the depth loss and the variational loss i.e., $\mathcal{L} = \mathcal{L}_{depth} + \lambda \mathcal{L}_{var}$, where λ is the regularization parameter set to 0.1 for all our experiments.

3.4 EXPERIMENTS AND RESULTS

Implementation Details We implemented our method in PyTorch 1.7.1 (Python 3.8) with CUDA 11.0. The software is evaluated on a computing machine with Quadro-RTX-6000 GPU. **Datasets.** We performed experiments on three benchmark datasets namely NYU Depth V2 [206], KITTI [64], and SUN RGB-D [210]. ***(a)*** **NYU Depth V2** contains images with 480×640 resolution with depth values ranging from 0 to 10 meters. We follow the train and test set split from [124], which contains 24,231 train images and 654 test images. ***(b)*** **KITTI** contains images with 352×1216 resolution where depth values range from 0 to 80 meters. The official split provides 42,949 train, 1,000 validation, and 500 test images. [55] provides another train and test set split for this dataset which has 23,488 train and 697 test images. ***(c)*** **SUN RGB-D** We preprocess its images to 480×640 resolution for consistency. The depth values range from 0 to 10 meters. We use the official test set (5050 images) for evaluation.

Training Details. We use [152] network as our backbone, which is pretrained on ImageNet [46]. We use the Adam optimizer [106] without weight decay. We decrease the learning rate from $3e^{-5}$ to $1e^{-5}$ by the cosine annealing scheduler. To avoid over-fitting, we augment the images by horizontal flipping. For KITTI [64], the model is trained for 10 epochs for the official split and 20 epochs for the Eigen split [55]. For NYU Depth V2 [206], the model is trained for 20 epochs.

Table 3.1: Comparison with the state-of-the-art methods on the NYU test set [206]. Please refer to Sec.3.4.1 for details.

Method	Backbone	SILog ↓	Abs Rel ↓	RMS↓	RMS log↓	δ_1 ↑
GeoNet [181]	ResNet-50	-	0.128	0.569	-	0.834
DORN [58]	ResNet-101	-	0.115	0.509	-	0.828
VNL [253]	ResNeXt-101	-	0.108	0.416	-	0.875
TransDepth [248]	ViT-B	-	0.106	0.365	-	0.900
ASN [154]	HRNet-48	-	0.101	0.377	-	0.890
BTS [124]	DenseNet-161	11.533	0.110	0.392	0.142	0.885
DPT-Hybird [186]	ViT-B	-	0.110	0.357	-	0.904
AdaBins [15]	EffNet-B5+ViT-mini	10.570	0.103	0.364	0.131	0.903
ASTrans [28]	ViT-B	10.429	0.103	0.374	0.132	0.902
NeWCRFs [256]	Swin-L	9.102	0.095	0.331	0.119	0.922
Ours	Swin-L	**8.198**	**0.086**	**0.304**	**0.108**	**0.937**
% Improvement		-9.93%	-9.47%	-8.16%	-9.24%	+1.63%

Table 3.2: Comparison with the state-of-the-art methods on the the KITTI official test set [64]. We only list the results from the published methods. Please refer to Sec.3.4.1 for details.

Method	Backbone	SILog↓	Abs Rel↓	Sq Rel↓	iRMS↓
DLE [142]	ResNet-34	11.81	9.09	2.22	12.49
DORN [58]	ResNet-101	11.80	8.93	2.19	13.22
BTS [124]	DenseNet-161	11.67	9.04	2.21	12.23
BANet [5]	DenseNet-161	11.55	9.34	2.31	12.17
PWA [125]	ResNeXt-101	11.45	9.05	2.30	12.32
ViP-DeepLab [182]	-	10.80	8.94	2.19	11.77
NeWCRFs [256]	Swin-L	10.39	8.37	1.83	11.03
Ours	Swin-L	**9.84**	**7.96**	**1.66**	**10.44**
% Improvement		-5.29%	-4.90%	-9.29%	-5.35%

Table 3.3: Comparison with the state-of-the-art methods on the KITTI Eigen test set [55].

Method	Backbone	SILog ↓	Abs Rel ↓	RMS↓	RMS log↓	δ_1 ↑
DORN [58]	ResNet-101	-	0.072	0.273	0.120	0.932
VNL [253]	ResNeXt-101	-	0.072	0.326	0.117	0.938
TransDepth [248]	ViT-B	8.930	0.064	0.275	0.098	0.956
BTS [124]	DenseNet-161	8.933	0.060	0.280	0.096	0.955
DPT-Hybird [186]	ViT-B	-	0.062	0.257	-	0.959
AdaBins [15]	EffNet-B5+ViT-mini	8.022	0.058	0.236	0.089	0.964
ASTrans [28]	ViT-B	7.897	0.058	0.269	0.089	0.963
NeWCRFs [256]	Swin-L	6.986	0.052	0.213	0.079	0.974
Ours	Swin-L	**6.817**	**0.050**	**0.209**	**0.076**	**0.977**
% Improvement		-2.42%	-3.85%	-1.88%	-3.80%	+0.03%

Evaluation Metrics. We report statistical results on popular evaluation metrics such as square root of the Scale Invariant Logarithmic error (**SILog**), Relative Squared error (**Sq Rel**), Relative Absolute Error (**Abs Rel**), Root Mean Squared error (**RMS**), and threshold accuracy.

3.4.1 *Comparison to State of the Art*

Tab.(3.1), Tab.(3.2), Tab.(3.3), and Tab.(3.4) provide statistical comparison results with the competing methods on NYU Depth V2, KITTI official split, KITTI Eigen split, and SUN RGB-D, respectively. Our proposed approach shows the best results for all the evaluation metrics. Particularly on the NYU test set, we reduce the SILog error from the previous best result, 9.102 to 8.198, and increase δ_1 from 0.922 to 0.937. For the SUN RGB-D test set, all competing models, including ours, are trained on the NYU Depth V2 training set [206] *without* fine-tuning on the SUN RGB-D. In addition, we align the predictions from all the models with the ground truth by a scale and shift following [187]. Tab.(3.4) results show our method's better generalization capability than other approaches.

3.4.2 *Ablation Study*

All the ablation presented below is conducted on NYU Depth V2 test set [206].

Table 3.4: Comparison with AdaBins and NeWCRFs on SUN RGB-D test set. All methods are trained on NYU Depth V2 train set without fine-tuning on SUN RGB-D.

Method	Backbone	SILog ↓	Abs Rel ↓	RMS↓	RMS log↓	δ_1 ↑
AdaBins[15]	EffNet-B5+ViT-mini	13.652	0.110	0.321	0.137	0.906
NeWCRFs [256]	Swin-L	13.695	0.105	0.322	0.138	0.920
Ours	Swin-L	**12.596**	**0.094**	**0.299**	**0.127**	**0.929**
% Improvement		-7.73%	-10.48%	-6.85%	-7.30%	+0.98%

Table 3.5: **Benefit of V-layer.** We replace the proposed V-layer with a single convolutional layer and a self-attention layer, and evaluate the accuracy of depth map predicted with and without subsequent refinements.

Layer	Refine	SILog ↓	Abs Rel ↓	RMS↓	RMS log ↓	δ_1 ↑	δ_2 ↑
Convolution	w/o	8.830	0.090	0.325	0.114	0.927	0.990
	w/	8.688	0.089	0.317	0.113	0.928	0.991
Self-Attention + PE	w/o	8.790	0.090	0.318	0.114	0.927	0.990
	w/	8.595	0.089	0.316	0.112	0.929	0.991
V-Layer	w/o	8.422	0.087	0.308	0.110	0.936	0.990
	w/	**8.198**	**0.086**	**0.304**	**0.108**	**0.937**	**0.992**

***(i)* Effect of V-Layer.** To understand the benefit and outcome of our variational layer compared to other popular alternative layers in deep neural networks for this problem, we performed this ablation study. We replace our V-layer firstly with a convolutional layer and later with a self-attention layer. Tab.(3.5) provides the depth prediction accuracy for this ablation. For each introduced layer in Tab.(3.5), the first and second rows show the performance of the depth map predicted *with* (w) and *without* (w/o) subsequent refinements (cf. Sec.3.3.2 (c)), respectively. For the self-attention layer, we follow the ViT [50] and set the patch size to be one as we use the feature map with stride 16. We also adopt the learnable position embedding (PE) with 128 dimensions. We set the number of heads to be 4 and the number of hidden units to be 512. As shown in Tab.(3.5), our V-layer indeed helps improve the accuracy of depth prediction compared to other well-known layers.

***(ii)* Performance with Different Network Backbone.** We evaluate the effects of our V-layer with different types of network backbones. For this ablation, we use Swin-Large [152], Swin-Small [152], and ConvNeXt-Small [153]. The SILog error is shown in Fig. 3.4. The results show that

t

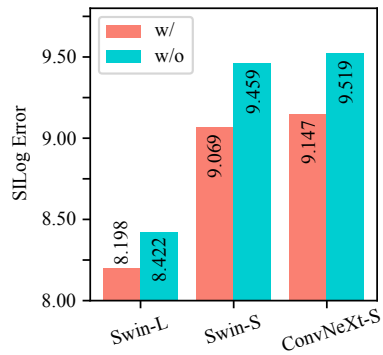


Figure 3.4: Evaluation on Swin-L, Swin-S, ConvNeXt-S **w/** and **w/o** the V-layer.

our V-layer improves the transformer and the convolutional network performance. An important observation is that our V-layer shows excellent improvements in depth prediction accuracy on weaker network backbones.

***(iii)* Performance with Change in the Value of S.** For this ablation, we change the value of S in the V-layer and observe its effects (cf. Sec.3.3.2 (b)). By increasing S, we generate more channels of $\tilde{\Gamma}$ and $\tilde{\Sigma}$ which in-effect increases V-layer parameters. In the subsequent step, we expand the number of channels to 128 by a convolutional layer to use the subsequent layers as they are. The results are shown in Tab.(3.6). For reference, we also present the result by replacing the V-layer with a convolutional layer in the first row in Tab.(3.6). By increasing S, we reduce the SILog error, at the price of the speed (FPS). Yet, no real benefit is observed with S more than 16.

***(iv)* Effect of Confidence Weight Matrix & Difference Operator in V-Layer.** For this ablation, we study the network's depth prediction under four different settings. (a) without V-layer and replace it with convolutional layer (b) without the confidence weight matrix (c) with learnable difference operator and (d) our full model. The depth prediction accuracy observed under these settings is provided in Tab.(3.7). Clearly, our full model has better accuracy. An important empirical observation, we made during this test is when we keep P learnable V-

Table 3.6: Analysis of the number of feature groups. More groups reduce the SILog error.

	SILog↓	Abs Rel↓	RMS↓	FPS ↑
w/o V-layer	8.688	0.089	0.317	**9.343**
1	8.456	0.088	0.310	8.175
16	8.198	0.086	**0.304**	7.032
128	**8.172**	**0.085**	0.309	3.320

Table 3.7: Analysis of the confidence weight matrix $\tilde{\Sigma}$ and the difference operator P.

	SILog↓	Abs Rel↓	RMS ↓
(a) w/o V-layer	8.688	0.089	0.317
(b) w/o $\tilde{\Sigma}$	8.537	0.089	0.316
(c) learnable P	8.355	0.088	0.310
(d) **full**	**8.198**	**0.086**	**0.304**

layer has more learnable parameters, the performance becomes worse than with fixed difference operator.

3.4.3 *Network processing time & Parameters*

We compared our method's inference time and the number of model parameters to the AdaBins [15] and the NeWCRFs [256]. The inference time is measured on the NYU Depth V2 test set with batch size 1. We have removed the ensemble tricks in AdaBins and NeWCRFs for an unbiased evaluation, resulting in a slight increase in SILog error as compared to Tab.(3.1) statistics. As is shown in Tab.(3.8), our method is faster and better than AdaBins and NeWCRFs using Swin-Small backbone. With the same backbone as the NeWCRFs, *i.e.*, Swin-Large, we achieve much better depth prediction results. Hence, our method with Swin-Small backbone provides a better balance between accuracy, speed and memory foot-print.

3.5 VISUALIZATION OF V-LAYER

We visualize the confidence weight map Σ_x, the difference map Γ_x, and the depth map Z_u from the V-layer in Fig.3.5. We observe that the depth value of a pixel shows correlation with respect to the image

Table 3.8: Comparison of the inference time and parameters to AdaBins and NeWCRFs on NYU Depth V2. We show our results using the Swin-Small and Swin-Large backbone.

	AdaBins [15]	NeWCRFs [256]	Ours (Small)	Ours (Large)
SILog Error ↓	10.651	9.171	**9.069**	**8.198**
Speed (FPS) ↑	5.638	10.551	**11.891**	7.032
Param (M) ↓	**75**	258	**76**	249

coordinates of the pixel. For example, in the last example in Fig.3.5, for different pixels at the door, the depth values are usually different but the first-order difference are approximately the same. This observation shows that the difference map might be easier to predict than the depth map.

3.6 CONCLUSION

In conclusion, a simple and effective approach for inferring scene depth from a single image is introduced. The proposed SIDP approach is shown to better exploit the rigid scene prior, which is generally overlooked by the existing neural network-based methods. Our approach does not make explicit assumptions about the scene other than the scene gradient regularity, which holds for typical indoor or outdoor scenes. When tested on popular benchmark datasets, our method shows significantly better results than the prior art, both qualitatively and quantitatively.

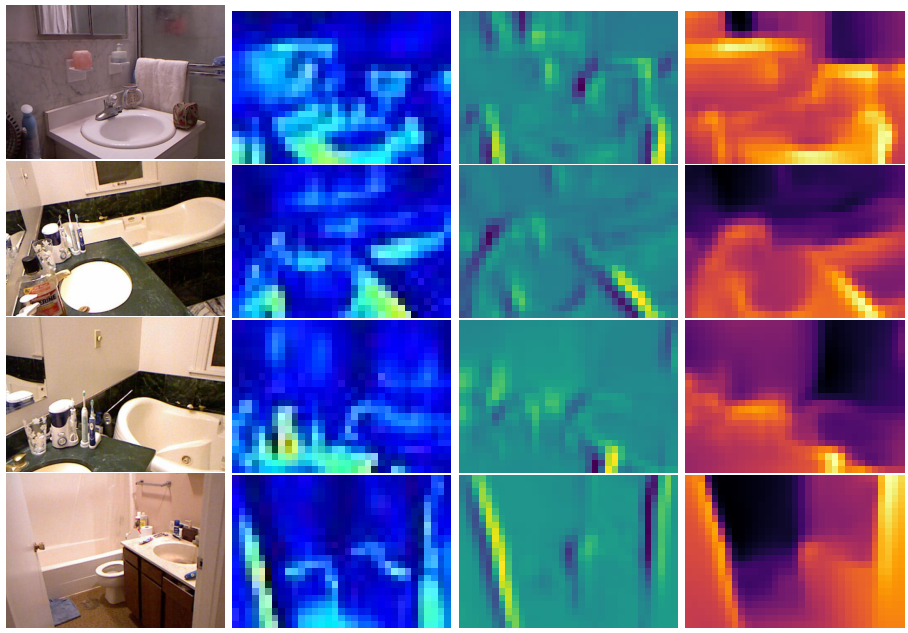


Figure 3.5: **Visualization of V-layer Prediction.** We visualize the confidence weight Σ_x, the difference map Γ_x and the depth map Z_u from the V-layer when predicting on NYU Depth V2 test set.

4 LINE PRIORS FOR DEPTH

This chapter is based on our paper: Ce Liu et al. "Deep line encoding for monocular 3d object detection and depth prediction." In: *32nd British Machine Vision Conference*. BMVA Press. 2021, p. 354.

4.1 INTRODUCTION

Recovering depth from RGB images has been a long-standing problem in vision and robotics. For example, a key step in 3D object detection is to predict the distance (depth) of foreground objects. Whereas in 3D reconstruction, the depth of all pixels is often required. It is known that for a single RGB image, in general, the problem is ill-posed [77]. Thereby the dominant approaches resort to multiple images from different viewpoints and locate points in 3D space by triangulation [77]. However, the human being can make a rough estimate of depth even from a single eye. The observation motivates researchers to explore various cues for single-image depth perception, such as shading [262, 87], defocus [72], and so on.

Restricted to street scenes, Dijk and Croon [49] show an important cue to infer objects' depth, *i.e.*, objects that are further away tend to appear higher in the image. However, bringing into full play the prior for accurate depth perception is by no means trivial. Because the road surface is assumed to be a rough plane, which is often violated in reality. For example, there are usually different slopes in different areas, and cars might even be on curb bricks. Road condition impacts greatly the actual relation between objects' depth and their image coordinates. How to model the above complex situation effectively is still an open question.

In this paper, we propose to exploit the basic primitives, like straight lines and vanishing points, in man-made environments, especially in autonomous driving scenarios. Because their angle or position indicates the slope of the road and even the 3D layout of the whole scene. To explicitly represent the semantics (*e.g.*, guard rail, horizontal line, *etc.*) and algebraic parameters of lines, we perform deep Hough trans-

form [53, 138] on the feature map of deep networks. It's known that in deep neural networks, the features in last convolutional blocks are rich in semantics [120]. The voting for a line is obtained by aggregating the features along the line, which encodes the semantic information from the entire line. In addition, the angle and position are indicated by the voting location in parameter space.

In parameter space, feature maps are sparser than the ones in image space in the sense that most of the locations are meaningless. Only a few elements represent straight lines that make sense, whereas the vast majority correspond to random aggregation. For efficiency, we propose the line pooling module to select important lines in parameter space. The lines in the scene are finally represented as a distributed vector.

We apply our design to off-the-shelf frameworks for monocular 3D object detection and depth prediction in autonomous driving scenarios. The main challenges in the two tasks are to predict the distance (depth) of foreground objects and estimate the dense depth map respectively. With deep line encoding, we advance the state-of-the-art on KITTI monocular 3D object detection and depth prediction benchmarks [64]. The improvements demonstrate the effectiveness of our design.

In summary, our main contributions are as follows: (1) We introduce the line information in scenes as a novel cue for single-image depth perception. (2) We propose a novel architecture to exploit the line information, which fits well into off-the-shelf frameworks. (3) We advance the state-of-the-art on KITTI single-image 3D object detection and depth prediction benchmarks.

4.2 RELATED WORK

In this section, we briefly review recent advances in monocular 3D object detection and depth prediction.

4.2.1 *Monocular 3D Object Detection*

In monocular 3D object detection, a series of attributes of the object is required to estimate, including the 3D coordinate, size and orientation. However, our method only aims to help the network better

estimate the Z coordinate (depth) of the object. Thereby we divide existing methods into three main categories according to their way to infer the depth of objects. It's noteworthy that, compared with the dense depth estimation task, only the depth of foreground objects are required. Thereby more priors can be exploited, and the methods can be more sophisticated.

The most popular way is to make use of the depth information provided by external models. A typical example of the external model is DORN [58], which learns to convert images to dense depth maps. Pseudo-LiDAR [231] back-projects the depth map into 3D points and then apply off-the-shelf LiDAR-based 3D object detectors. PatchNet [160] encodes camera calibration information by spatial coordinates transformation. However, instead of further improving depth estimation accuracy, such approaches focus more on making better use of the prediction results from existing models.

Deep3DBox [167] and RTM3D [130] infer depth through perspective relationships between 3D corners and their 2D projections. The motivation is to explicitly encode the geometric prior that objects that are further away appear smaller. However, other cues such as object's image position may be ignored. In addition, prediction error in orientation, dimension, and 2D bounding box will harm the accuracy of depth estimation.

4.2.2 *Monocular Depth Prediction*

Focusing on fully-supervised methods, we find that recent works mainly proceed in two directions.

One line is to design more sophisticated loss functions. Eigen *et al.* [55] proposed the scale-invariant loss to save the network from learning the absolute global scale. DORN [58] and SORD [48] treat depth network learning as an ordinal regression problem. Jiao *et al.* [96] investigated the long tail property of depth values and proposed the attention-driven loss. Wei *et al.* [254] formed a high-order geometric constraint called virtual normal, biasing the network to produce depth maps with better surface. Zhang *et al.* [264] jointly predict depth, surface normal and semantic segmentation to exploit cross-task affinity patterns.

Proposing novel network architectures is the other direction. Aich *et al.* [4] and Xu *et al.* [241] fuse the feature maps from different stages

of the backbone by the bidirectional attention modules and continuous graphical models respectively. Lee *et al.* [121] combine multi-scale depth map candidates in the Fourier domain. Chen *et al.* [32] proposed the spatial attention blocks to guide the network attention to global structures or local details across different feature layers. Huynh *et al.* [93] developed the depth-attention volume to exploit planar structures in the scene.

4.3 APPROACH

In this section, we start with an analysis of the simplified projection model. Its limitation motivates us to explicitly encode the line information from the scene. As an effective way, the Hough transform is briefly reviewed, and then we take a step further by proposing the novel line pooling module. At last, we present the overall architecture with deep line encoding.

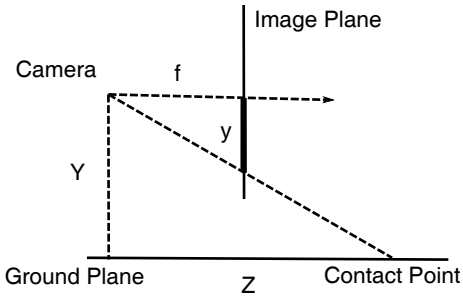
4.3.1 *Depth from Lines*

In autonomous driving scenarios, an important cue to depth prediction is object's vertical position in the image. As shown in Figure 4.1 (a), in the ideal case where the ground plane is perfectly horizontal, the distance of the object can be easily obtained by:

$$Z = \frac{fY}{y}, \tag{4.1}$$

where f and Y are camera's focal length and height respectively, and y is the image vertical coordinate difference between the principal point and the 2D projection of object's ground contact point.

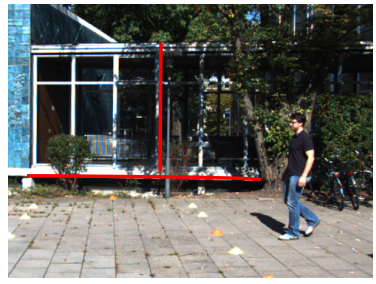
In real-world applications, the projection relation can be affected by various factors, such as slope, step, camera pose variation, and so on. Special structures in the scene, especially straight lines and vanishing points, can indicate the geometric layout of the scene and help the convolutional networks reason the real projection relation, as shown in Figure 4.1 (b)-(d).



(a) Projection Model

(b) Slope

(c) Step



(d) Pose Variation

Figure 4.1: (a) Illustration of depth estimation from the image coordinate of the object. (b)-(d) Examples of real situations. We highlight some representative lines (red) that provide information about slope, step and camera pose variation respectively.

4.3.2 *Deep Line Encoding*

In this section, we introduce the deep line encoding to make better ues of the line information from the scenes.

4.3.2.1 *Hough Transform*

Comparing with conventional networks, the Hough transform [53] provides a different perspective to represent lines in the scene. Given an image, a straight line l can be represented by a point (θ, ρ) in the parameter space, where θ is the angle between the x-axis and the normal vector of the line, and ρ is the distance from the origin to the line.

The traditional Hough transform algorithm usually takes a binary edge map as input. Its power is limited because the edge map is short of semantic context and prone to noise. Recently, Lin *et al.* [138] and

Han *et al.* [76] proposed the deep Hough transform. The input is replaced with features from deep networks, which enables end-to-end training and is more robust. Given a feature map X, the transformed map Y is calculated by the following equation:

$$Y(\theta, \rho) = \sum_{(x,y) \in l} X(x, y), \tag{4.2}$$

where l is the line parameterized by (θ, ρ).

It's known that in last convolutional blocks of deep networks, the feature maps are rich in semantics [120]. Thereby the aggregation of features along the line can be an encoding for the semantics of the entire line, which is necessary for the line pooling module to select the most important lines. The voting position (θ, ρ) in the transformed map Y indicates the algebraic parameters of the line.

In implementation, we set the origin to be the center of the feature map X, and discretize $\theta \in [0, 180^\circ)$ and $\rho \in [-\sqrt{W^2 + H^2}/2, \sqrt{W^2 + H^2}/2]$ into bins, where W and H are the width and height of the feature map X respectively.

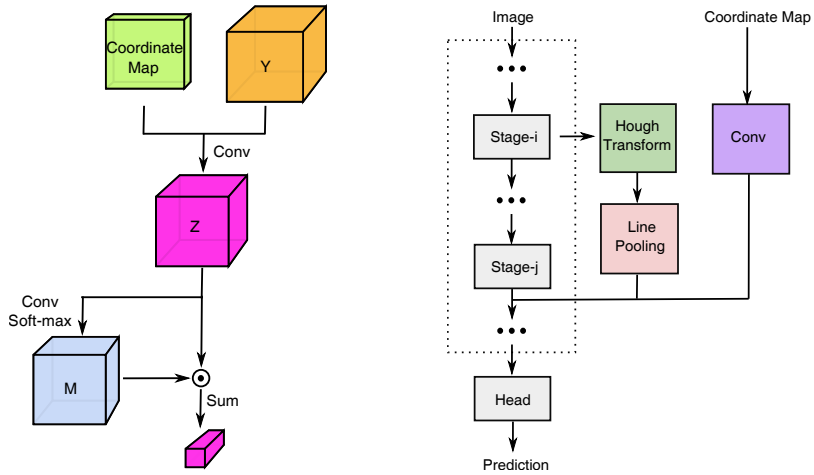
4.3.2.2 *Line Pooling*

In deep Hough transform, the size of the transformed map Y is usually large but most of the elements are invalid aggregation. As only a few elements of Y represent lines that make sense, we propose a line pooling module to compress Y. The pipeline of our proposed line pooling module is shown in Figure 4.2 (a).

4.3.3 *Overall Architecture*

Our method is generic and can be applied to various frameworks. For generality, we suppose the rough architecture for monocular 3D object detection or depth prediction to be as shown in Figure 4.2 (b). Given an input image, the backbone extracts features at first, and then the head makes prediction for the specific task.

In most cases, such as in VGG [207] and ResNet [78], the backbone is composed of a series of stages. The feature maps from early stages are of high resolution but semantically weak. After stacks of convolution and pooling layers, features from different locations are hierarchically



(a) Line Pooling (b) Overall Architecture

Figure 4.2: (a) Pipeline of the line pooling module. The circle with black dot denotes element-wise product. (b) Overview of the architecture with deep line encoding. Dotted box indicates the backbone.

aggregated in a complex way. Thereby in later stages the feature maps are semantically stronger but of lower resolution.

We perform deep Hough transform [138, 76] on feature maps from an early stage of i. The choice is natural since high resolution maps preserve more accurate location information. A key step in deep Hough transform [138, 76] is to aggregate the features along the line. The resulting vector can be viewed as an explicit representation for the semantics of the entire line, which is necessary for the line pooling module to distinguish between the guard rail and the horizontal line. The angle and position of the line are indicated by the voting location. After deep Hough transform [138, 76], key lines in the scene will be selected by the line pooling module and encoded as a short vector.

To incorporate the line information into the backbone, we concatenate the feature map F from the stage of j with the line vector. To do so, we up-sample the line vector to the same resolution as F in advance. The relative position of the object with respect to the lines or the principle point is an important factor, thereby we also append the coordinate map [148]. We concatenate F, the line vector and the coordinate map

and compress into the same number of channels as F. The resulting feature map will replace F and be fed into the stage of $j+1$.

While the deep line encoding module is plugged into the backbone, other parts of the framework are kept the same as original. We train the framework in an end-to-end manner, without any extra supervision than the common monocular 3D object detection or depth estimation frameworks. In experiment, we found the network can discover related lines automatically.

4.4 EXPERIMENT

In this section, we analyze and verify each component in deep line encoding by detailed ablation study and visualization. We also apply deep line encoding to the state-of-the-art frameworks for monocular 3D object detection and depth prediction.

4.4.1 *Monocular 3D Object Detection*

Dataset The KITTI 3D Object Detection dataset [64] consists of 7,481 training images and 7,518 testing images from autonomous driving scenes. Following Chen *et al.* [34], we split the training data into 3,712 training and 3,769 validation images. As has been the focus of prior work [20], we primarily compare methods using the car class.
VisualDet3D We apply deep line encoding to the state-of-the-art monocular 3D object detector, *i.e.*, VisualDet3D [151]. It is a single end-to-end network composed of a backbone and two task-specific heads. The backbone is responsible for extracting feature maps over the input image. The default is to take the first three stages of ResNet-101 [78] as backbone. The first head performs convolutional object classification; the second head performs convolutional 3D bounding box regression. The 3D bounding box is disentangled into a group of parameters to be regressed, including 2D projection of the 3D center, depth, size and orientation. The detector is trained with focal loss [137] for the classification branch and smooth l1 loss [65] for the regression branch.
Line Encoding We choose feature maps from the first stage to perform deep Hough transform. The transformed map will be compressed into a short vector by the line pooling module. The line vector and coordinate map will be fused with the feature map from the second stage

Configuration	Coordinate	Line Vector	AP_{3D}		
			Easy	Moderate	Hard
a			23.63	16.16	12.06
b	✓		25.21	16.48	12.46
c		✓	24.92	16.41	12.33
d	✓	✓	**26.49**	**16.75**	**13.07**

Table 4.1: Different configurations ablation study. Configuration a is from Liu *et al.* [151].

and then fed into the third stage.

Training Details We adopt the Adam algorithm [107] to optimize network parameters for 40 epochs. We use an initial learning rate of 1e-4, a cosine annealing scheduler [156] with target learning rate of 1e-5, a batch size of 8, and no weight decay. The data pre-processing and augmentation are exactly the same as VisualDet3D [151]. We increase the loss weight of depth prediction branch from 3.0 to 5.0, while other hyper-parameters are kept the same. For deep Hough transform [138, 76], we set the resolution of ρ and θ to be 3 pixel units and 3°.

4.4.2 *Ablation Study*

We perform detailed ablation study on the validation set. The experiments are repeated 5 times for average. We first evaluate the effects of critical components in deep line encoding, including the coordinate map and the line vector. Main results are shown in Table 4.1. From Table 4.1 (b), we observe that simply adding the coordinate map to the framework can boost the performance significantly. The result coincides with the conclusion of Liu *et al.* [148], *i.e.*, the vanilla network might fail to learn the absolute coordinate information. Table 4.1 (c) shows the improvement from the line vector alone, indicating that the information about straight lines in scenes is beneficial for depth perception. We observe a further improvement from Table 4.1 (d) by combining the coordinate map with the line vector.

Then we evaluate the effects of each component in the line pooling module, the results are shown in Table 4.2. One of the most important design is the coordinate map, which represents the algebraic parame-

Pooling	AP_{BEV}			AP_{3D}		
	Easy	Moderate	Hard	Easy	Moderate	Hard
LinePooling	**34.06**	**22.59**	**16.96**	**26.43**	**16.72**	**13.02**
-Coordinate	33.25	22.11	16.77	24.96	16.44	12.52
-Soft-max (avg)	33.34	22.55	16.91	24.77	16.26	12.55
-Soft-max (max)	31.50	21.68	16.29	23.28	15.80	11.75

Table 4.2: Pooling ablation study.

ters of lines. If we remove the coordinate map, there will be a drop in performance, especially in the hard case of AP_{3D}. The other key design is the soft-max operation. We verify its effects by directly selecting the average or maximum in each channel of Z. The results show that the soft-max operation is better. Particularly, selecting the maximum even harms the performance.

We further apply deep line encoding to M3D-RPN [20] to demonstrate the generalization ability across frameworks. M3D-RPN [20] is a simple single-stage network composed of a backbone and two task-specific heads. It takes DenseNet-121 [91] as backbone, while removing the final pooling layer and dilating each convolutional layer in the last Dense-Block by a factor of 2. The heads are used for object classification and attributes regression respectively. We optimize the model for 100 thousands iterations. Other hyper-parameters are kept the same.

We select the feature map from the second Dense-Block to extract the line vector. Then together with the coordinate map, we fuse the line vector with the original feature map and feed into following modules. As shown in Table 4.3, deep line encoding module improves the performance of M3D-RPN [20] under all the metrics. We also list the performance of VisualDet3D [151] for comparison.

4.4.3 *Computation Cost*

The number of trainable parameters and test time are shown in Table 4.4. Following Lin *et al.* [138], the deep Hough transform is implemented as matrix multiplication. We compress the feature maps into 16 channels before feeding into the deep Hough transform module. The accumulator is then expanded to 256 channels through convolu-

Method	AP_{BEV}			AP_{3D}		
	Easy	Moderate	Hard	Easy	Moderate	Hard
M3D-RPN [20]	20.85	15.62	11.88	14.53	11.07	8.65
+ **Line Encoding (ours)**	**22.97**	**16.59**	**13.33**	**16.36**	**11.93**	**9.35**
VisualDet3D [151]	29.70	20.98	16.20	23.63	16.16	12.06
Line Encoding (ours)	**34.06**	**22.59**	**16.96**	**26.43**	**16.72**	**13.02**

Table 4.3: Frameworks ablation study.

tional layers. Thereby the line vector has a length of 256. We evaluate the test time for the two models both on NVIDIA Tesla V-100 GPU with a batch size of 1.

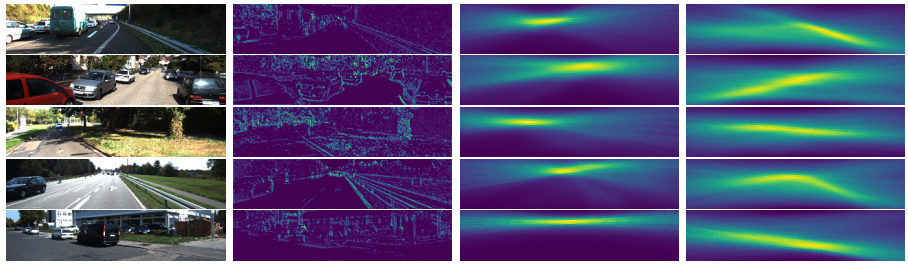
Method	Param (M)	Test Time (s/image)
VisualDet3D [151]	55.68	0.053
+ **Line Encoding (ours)**	61.84	0.060

Table 4.4: Model size and processing time.

4.4.4 *Visualization of Lines*

We perform inverse Hough transform on the probability map M to visualize the lines selected by the line pooling module. The results are shown in Figure 4.3. The first column shows the input image. The second column shows an example of the feature map to perform deep Hough transform. The third and the forth columns show the inverse Hough transform of different channels of M.

In Figure 4.3 (c) and (d), the bright areas show the distribution of important lines generated by different channels of M. The areas are blur because the probability is often distributed among a combination of multiple lines. However, we still observe that the selected lines often align with special structures in scenes, such as guard rails, horizontal lines, vanishing points and so on. Specifically, Figure 4.3 (c) focuses on vanishing points or horizontal lines, while Figure 4.3 (d) is sensitive to the guard rails. Although we select only a single value from each channel, the soft-max operation is flexible such that the probability can



(a) Image (b) Feature Map (c) 1_{st} Example (d) 2_{nd} Example

Figure 4.3: Visualization of the probability map M.

concentrate on a single global maximum, or several local maxima. On Figure 4.3 (d) it is obvious that both the left and right guard rails are selected.

4.4.5 *Comparison with State-of-The-Art Methods*

Table 4.5 presents the performance of recent methods on the KITTI monocular 3D object detection benchmark [64]. Our method shows superiority over VisualDet3D [151] and achieves the state-of-the-art on easy and moderate cases. Particularly, on the easy case of AP_{3D} our method increases by 2.58 points of AP (24.23 *vs.*21.65). For the hard case, our method lags behind MonoPair [35]. However, our method does not exploit many popular improvements, such as the feature pyramid network [136] and deep layer aggregation [255], which are helpful for small objects. These improvements are complementary to deep line encoding and should boost the accuracy of hard case further.

4.4.6 *Monocular Depth Prediction*

Dataset The KITTI monocular depth prediction dataset [64] consists of 42,949 training images, 1,000 validation images, and 500 test images, annotated with sparse point clouds. The NYU Depth V2 dataset [206] consists of 120,000 images captured in indoor scenes. Following the official split, we use 249 scenes for training and 215 scenes (654 images) for testing. In the training set, 24,231 images and depth maps are associated and sampled using timestamps by even-spacing in time. We train and test on the center cropping proposed by Eigen *et al.* [55].

Method	AP_{BEV}			AP_{3D}		
	Easy	Moderate	Hard	Easy	Moderate	Hard
M3D-RPN [20]	21.02	13.67	10.23	14.76	9.71	7.42
MonoPair [35]	24.12	18.17	**15.76**	16.28	12.30	**10.42**
RTM3D [130]	19.17	14.20	11.99	14.41	10.34	8.77
PatchNet [160]	22.97	16.86	14.97	15.68	11.12	10.17
VisualDet3D [151]	29.81	17.98	13.08	21.65	13.25	9.91
+ **Line Encoding (ours)**	**31.09**	**19.05**	14.13	**24.23**	**14.33**	10.30

Table 4.5: Performance comparison on KITTI monocular 3D object detection benchmark.

GAC We select the open-source framework GAC [151] to evaluate deep line encoding. GAC is based on the U-Net structure [195], and is composed of a backbone and a head. The default is to take the ResNet-34 [78] as backbone. The head is to fuse features from different stages into a feature map with high resolution and rich semantic information, and then perform convolutional depth regression. The network is trained with a scale-invariant loss [55] and a smoothness loss.
Line Encoding We perform deep Hough transform [138, 76] on the features from the second stage of the backbone. The line vector will be fused with coordinate maps and feature maps from the third stage.
Training Details We adopt the Adam algorithm [107] to optimize network parameters for 8 epochs. We use an initial learning rate of 1e-4, a cosine annealing scheduler [156] with target learning rate of 1e-5, a batch size of 8, and no weight decay. For deep Hough transform [138, 76], we set the resolution of ρ and θ to be 3 pixel units and 3°.
Performance on KITTI As shown in Table 4.6, with deep line encoding, we achieve a better performance than GAC [151] under all the metrics. Especially in sqErrorRel, we observe a significant improvement (2.22 *vs.*2.61). Comparing with other published methods, we achieve the state-of-the-art in terms of the metric of sqErrorRel, and the second best rating under absErrorRel and iRMSE.
Performance on NYUDv2 NYU Depth V2 data set is captured in indoor scenes by Microsoft Kinect. Thereby the camera poses with respect to the ground plane are more variable than in KITTI, and the simplified projection model in Figure 4.1 (a) might be inapplicable in a lot of images. However, we still observe an improvement in Table 4.7

Method	SILog	sqErrorRel	absErrorRel	iRMSE
DORN [58]	11.77	2.23	**8.78**	12.98
VNL [254]	12.65	2.46	10.15	13.02
SORD [48]	12.39	2.49	10.10	13.48
BANet [4]	**11.55**	2.31	9.34	**12.17**
GAC [151]	12.13	2.61	9.41	12.65
+ **Line Encoding (ours)**	11.81	**2.22**	9.09	12.49

Table 4.6: Performance comparison on KITTI single-image depth prediction benchmark.

Method	SILog	sqErrorRel	absErrorRel	iRMSE
GAC [151]	13.72	3.79	13.15	8.67
+ **Line Encoding (ours)**	**12.71**	**3.20**	**12.44**	**8.22**

Table 4.7: Performance comparison on NYU Depth V2 test set.

with deep line encoding. The improvement shows that the line information is beneficial even in indoor scenes.

4.5 CONCLUSION

Recovering depth from a single RGB image is a challenging task. In this paper, we have shown that line structures in scenes provide valuable information for depth perception. Furthermore, we presented a simple architecture to exploit the lines inside ConvNets. Our method obtains state-of-the-art results on single-image 3D object detection and depth prediction benchmarks. Finally, our study suggests that despite the success of deep ConvNets, it is still necessary to incorporate prior knowledge and design more efficient representation for the specific task.

5

STEREO RISK

This chapter is based on our paper: Ce Liu et al. "Stereo Risk: A Continuous Modeling Approach to Stereo Matching." In: *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

5.1 INTRODUCTION

Stereo Matching is one of the most important and fundamental problems in computer vision [85, 97, 200, 215]. Given a rectified stereo image pair captured at the same timestamp, the goal of stereo matching is to estimate the per-pixel displacement from left to right images, popularly known as a disparity map. Under the rectified image pair setup, the stereo matching problem boils down to a well-structured 1D search problem in the image space [215].

Due to its effectiveness and affordability, stereo camera rigs have been widely adopted in commercial and industrial applications, including autonomous driving cars [57, 16], smartphones [164, 159, 175], and other robotic automation systems [105, 89].

Classical well-known stereo matching methods—often categorized as local methods, use a predefined support window to find suitable matches between stereo image pair [200, 81]. Yet, approaches that optimize for all disparity values using a global cost function were observed to provide better results [110, 108, 18, 246]. In recent years, with the surge in high-quality, large-scale synthetic ground-truth data, availability of high-end GPUs' and advancements in deep-learning architecture, the neural network-based stereo matching models trained under supervised setting has outperformed classical methods accuracy by a significant margin [103, 26, 259, 139]. Nevertheless, one fundamental challenge still remains, *i.e.*how to model *continuous* scene disparity values given only a limited number of candidate pixels to match? After all, the scene is continuous in nature.

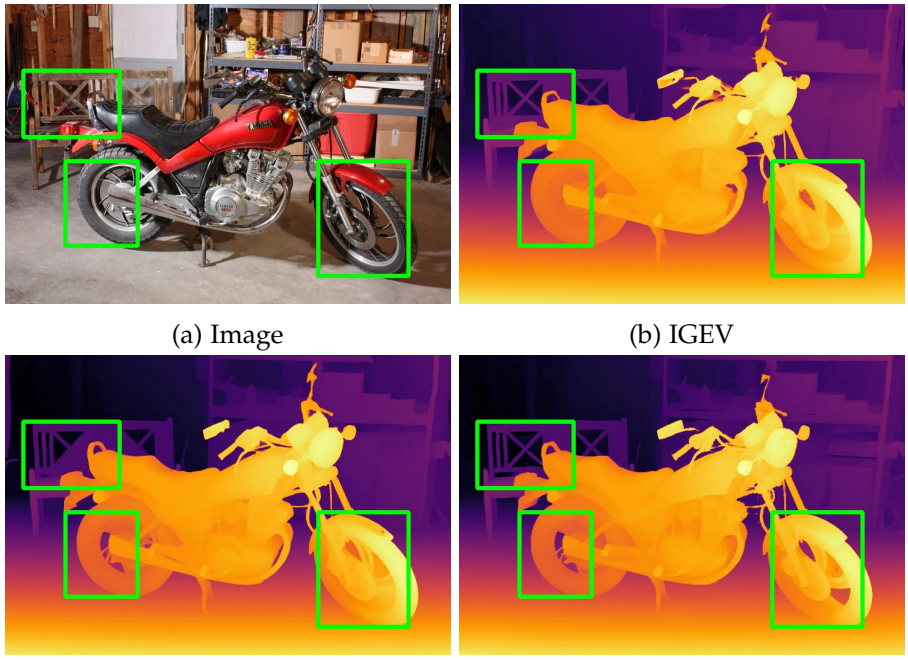
Many recent works have attempted to overcome the above challenge of predicting continuous scene disparities, which can be broadly divided into two categories. ***(i)*** **Regression-based approaches** predict a

real-valued offset by neural networks for each hypothesis of discrete disparity. The offset is then added to the discrete disparity hypothesis as the final continuous prediction. Typical examples include RAFT-Stereo [139], CDN [62], and more recent IGEV [244] and DLNR [265]. ***(ii)* Classification-based approaches** first estimate the categorical distribution[1] for the discrete disparity hypotheses and then take the expectation value of the distribution as the final disparity, which can be any arbitrary real value even though the categorical distribution is discrete [103, 26, 259].

In this chapter, we aim to address the importance of continuous disparity modeling in stereo matching, given the categorical distribution of disparity hypotheses. We introduce a radically different perspective on the disparity prediction problem by framing it as a search problem of finding the minimum risk [126, 226, 14] of disparity values. Specifically, the risk is defined by averaging the prediction error with respect to all possible values of the ground-truth disparity. At the time of making the prediction, the ground truth is unavailable, which is therefore approximated by the disparity hypotheses with a categorical distribution. We search for a disparity value as our prediction that achieves minimal overall risk involved with it. Moreover, we demonstrate that the commonly used disparity expectation [103] is a special case of L^2 error function within the proposed risk formulation, which is sensitive to multi-modal distribution and may result in the over-smooth solution [29, 222]. In contrast, we advocate the use of the L^1 error function during risk minimization.

Despite the theoretical soundness of the L^1 risk minimization, there is no closed-form solution to L^1 formulation. To that end, in this paper, we search for the solution by computing derivatives of our proposed risk function and performing its continuous optimization. By interpolating the disparity categorical distribution, we define our continuous probability density function. Then, we propose a binary search algorithm to find the optimal disparity that minimizes the proposed risk efficiently. To enable the end-to-end network training, we compute the backward gradient of the final disparity with respect to the categorical distribution by the implicit function theorem [111].

1 A categorical distribution is a discrete probability distribution that describes the possible results of a random variable that can take on the K possible categories, with the probability of each category separately specified.



(a) Image (b) IGEV

(c) DLNR (d) Ours

Figure 5.1: **Qualitative Comparison.** We compare our method with recent state-of-the-art methods such as IGEV [244], DLNR [265] on Middlebury [200]. All methods are trained only on SceneFlow [161], and evaluated at quarter resolution. It can be observed that our method generalizes and predicts high-frequency details better than other recent methods.

We have extensively evaluated the proposed method on a variety of stereo matching datasets. Our approach demonstrates superior performance compared to many state-of-the-art methods on benchmarks such as SceneFlow [161], KITTI 2012 [64], and KITTI 2015 [163]. Moreover, our approach achieves significantly better cross-domain generalization, as observed on Middlebury [200], ETH 3D [203], KITTI 2012 & 2015. An example of qualitative comparison is given in Fig. 5.1. Ablation studies confirm the effectiveness of risk minimization, not only within the proposed network but also in the context of general stereo matching networks, such as ACVNet [243] and PCWNet [205].

5.2 RELATED WORK

5.2.1 *Deep Neural Network For Stereo Matching*

In recent years, the deep-learning based approaches have improved the accuracy of stereo matching by a significant margin. Designing powerful and efficient network architectures for stereo matching is a popular research topic. [257] apply deep convolutional networks [119] to learn discriminative features for image patches. DispNetCorr [161] designs explicit correlation in networks to construct cost volume. GC-Net [103] constructs volume by concatenation and refines by 3D convolution. PSM-Net [26] exploits spatial pyramid pooling [266] and stacked hourglass [171] to learn context information. STTR [132] applies transformers [227, 51] to relax the limitation of a fixed disparity range. Moreover, the uniqueness constraint is considered by optimal transport [43]. ACVNet [243] weights the matching costs by attention.

Another line of research is to improve efficiency. In GANet [259] the computationally costly 3D convolutions are replaced by the differentiable semi-global aggregation [81]. GWCNet [71] constructs the cost volume by group-wise correlation. AANet [245] proposes the adaptive cost aggregation to replace the 3D convolution for efficiency. AnyNet [232], DeepPruner [54], HITNet [216], CasMVSNet [70], PCWNet [205] and Bi3D [8] prune the range of disparity in the iterative manner. RAFT-Stereo [139], CREStereo[128], IGEV [244] and DLNR [265] use recurrent neural networks [38] to predict and refine the disparity iteratively.

In this chapter, our network structure is inspired by CasMVSNet [70], and consists of two stages to predict and refine the disparity map. The

hierarchical design reduces the time and memory cost, while keeping the matching accuracy.

5.2.2 *Continuous Disparity by Classification*

In deep networks that have cost volumes, the most popular way to predict the disparity from the volume is the weighted average operation, i.e. expectation. [29] find the average operation suffers from the over-smoothing problem, especially at the boundaries of objects. Therefore they propose the single-modal weighted average. [62] propose to predict a continuous offset to shift the distribution modes of disparity. Furthermore, they generate multi-modal ground truth disparity distributions and supervise the network to learn the distribution by Wasserstein distance [228]. SMD-Net [222] exploit bimodal mixture densities as output representation for disparities. UniMVSNet [176] attempts to unify the advantages of classification and regression by designing a novel representation, and further proposes a unified focal loss. [249] use top-K hypotheses for the disparity to alleviate the multi-modal problem. In this paper, we propose to minimize the risk under L^1 norm to capture continuous disparity and solve the multi-modal problem. Moreover, our approach can be trained in an end-to-end manner.

5.2.3 *Cross-Domain Generalization*

Existing real-world stereo datasets are small and insufficient to train neural networks from scratch, therefore exploiting synthetic images to pre-train networks and reducing the domain gap play an important role. [219, 220, 221] fine tune the stereo matching networks on the target domain using unsupervised loss. [149] jointly optimize networks for domain translation and stereo matching during training. [260, 211] normalize features to reduce domain shifts. [23, 140] design robust features for stereo matching. [141] find the cost volume built by cosine similarity generalizes better to different image features. [261] apply the stereo contrastive loss and selective whitening loss to improve feature consistency. [27] proposed the hierarchical visual transformation to learn shortcut-invariant robust representation from synthetic images. In this paper, we present a novel perspective to improve robustness by

L^1 risk minimization. We also show that our approach can be combined with above methods to further improve the robustness.

5.3 METHOD

5.3.1 *Probability Density of Continuous Disparity*

For each pixel in the left image, suppose the possible disparities are in the range of $[d_{\texttt{min}}, d_{\texttt{max}}]$. Typical stereo matching algorithms will compute a cost that merely can be described as a probability mass function (PMF) with a finite set of disparities $\mathbf{d} = [d_1, ..., d_N]^T$ and compute a discrete distribution $\mathbf{p}^m = [p_1^m, ..., p_N^m]^T$, where $d_i \in [d_{\texttt{min}}, d_{\texttt{max}}]$ and p_i^m is the probability that the ground truth disparity is d_i. The $\mathbf{p}^m$ is required to satisfy the conditions $p_i^m \geq 0$ and $\sum_i p_i^m = 1$.

The discrete formulation reasons the probability only at a finite set of disparities. However, in real-world applications, the ground-truth disparity is continuous. Therefore we propose to interpolate the discrete distribution by the Laplacian kernel, and the probability density function of disparity $x \in \mathbb{R}$ is computed by

$$p(x; \mathbf{p}^m) = \sum_i^N k(x, d_i) p_i^m \tag{5.1}$$

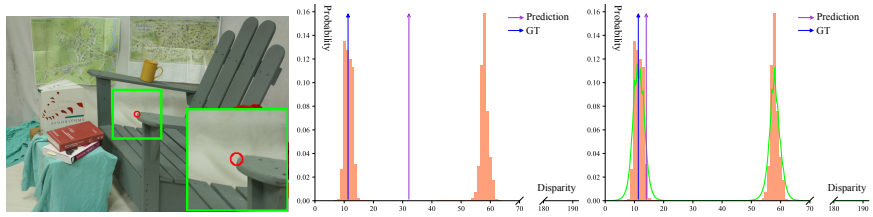
where $k(x, d_i)$ is defined as $\frac{1}{2\sigma} \exp -\frac{|x-d_i|}{\sigma}$, and σ is the hyper-parameter for bandwidth. The above density function is valid because $p(x; \mathbf{p}^m) \geq 0$ for $\forall x \in \mathbb{R}$ and $\int p(x; \mathbf{p}^m) dx = 1$. An illustration of the interpolation is shown in Fig. 5.2 (c). The orange bars represent the given discrete distribution $\mathbf{p}^m$, and the green curve is the interpolated density function. In the following we show the continuous formulation enables us to compute the derivative of the risk function.

5.3.2 *Risk of Disparity*

To choose a value as the final prediction, we propose to minimize the following risk:

$$\texttt{argmin}_y F(y, \mathbf{p}^m) = \texttt{argmin}_y \int \mathcal{L}(y, x) p(x; \mathbf{p}^m) dx \tag{5.2}$$

where $F(y, \mathbf{p}^m)$ is called as the risk at y, and $\mathcal{L}(y, x)$ is the error function between y and x. By risk we mean that if we take y as predicted



(a) Image (b) Expectation (c) Ours

Figure 5.2: **Illustration of the difference between the expectation and our method.** In (a) the pixel in the red circle is located at the boundary of the chair, thereby the distribution of the disparity has multiple modes. We plot the discrete distribution of disparity hypotheses by orange bars in (b) and (c). In (b) the prediction obtained by averaging is blurred and far from any of the modes. In (c) we find the optimal solution under L^1 norm, which is more robust and closer to the ground truth. The green curve is the interpolated probability density.

disparity, how much error there shall be with respect to the ground truth. Since the exact ground truth is unavailable at the time of making the prediction, we average the error across all possible ground-truth disparities with the distribution $p(x; \mathbf{p}^m)$.

Previous methods usually compute the expectation value of x as the final prediction for the disparity:

$$y = \int x p(x; \mathbf{p}^m) dx. \tag{5.3}$$

In our framework, we can derive the same prediction when using the squared L^2 norm as the error function. More specifically,

$$\texttt{argmin}_y F(y, \mathbf{p}^m) = \int x p(x; \mathbf{p}^m) dx \tag{5.4}$$

when $\mathcal{L}(y, x) = (y - x)^2$.

However, it is well known that the L^2 norm is not robust, and prones to outliers [17]. As shown in Fig. 5.2 (b), the expectation is inaccurate when there are multiple modes in the distribution. Instead, we select the L^1 norm in our risk function:

$$\texttt{argmin}_y F(y, \mathbf{p}^m) = \texttt{argmin}_y \int |y - x| p(x; \mathbf{p}^m) dx. \tag{5.5}$$

Given the distribution $p(x; \mathbf{p}^m)$ of the disparity, the optimal y will minimize the L^1 error with respect to all possible disparities weighted by the corresponding probability density. As shown in Fig. 5.2 (c), our final prediction is more robust to the incorrect modes and closer to the ground truth.

5.3.3 *Differentiable Risk Minimization*

One challenge of the L^1 norm is that there is no closed-form solution to the minimal risk in Eq.(5.5). To search for the optimal solution and enable end-to-end training, we introduce the details for the forward prediction and backward propagation below.

Forward Prediction. Given the discrete distribution $\mathbf{p}^m$, we find the optimal y of Eq.(5.5) efficiently based on the following two observations. Firstly, the target function $F(y, \mathbf{p}^m)$ is convex with respect to y, thereby we find the optimal solution at where $\partial F / \partial y = 0$.

$$G(y, \mathbf{p}^m) \triangleq \frac{\partial F(y, \mathbf{p}^m)}{\partial y} = \sum_i p_i^m \texttt{Sign}(y - d_i)(1 - \exp - \frac{|y - d_i|}{\sigma}) = 0 \tag{5.6}$$

where $\texttt{Sign}()$ is the sign function, which a slight abuse of notation. $\texttt{Sign}()$ can be thought of as an indicator function, where it is 1 if $y > d_i$ and -1 otherwise. Secondly, the second-order derivative $\partial^2 F / \partial^2 y \geq 0$, so the first-order derivative is a non-decreasing function. We find the optimal disparity, i.e. the zero point of $G(y, \mathbf{p}^m)$, by binary search, as shown in Alg. 1. In all experiments, we set the σ and τ as 1.1 and 0.1 respectively. For N disparity hypotheses, the binary search algorithm can find the optimal solution with time complexity of $O(\log N)$ [42].

Backward Propagation. As alluded to above, the procedure of the forward prediction (Alg. 1) to solve Eq.(5.5) contains non-differentiable operations. However, to enable end-to-end training, we have to compute $dy / d\mathbf{p}^m$ to backward propagate the gradient. Our method is inspired by the Implicit Function Theorem [111]. More specifically, because $G(y, \mathbf{p}^m) \equiv 0$ at the optimal y, we obtain

$$dG(y, \mathbf{p}^m) = \frac{\partial G}{\partial y} dy + \frac{\partial G}{\partial \mathbf{p}^m} d\mathbf{p}^m = 0. \tag{5.7}$$

Algorithm 1 Forward Prediction

Require: $\tau > 0$, $\sigma > 0$, $\mathbf{d} = [d_1, ..., d_N]$, $d_1 < d_2 < \cdots < d_N$, and $\mathbf{p}^m = [p_1^m, ..., p_N^m]$
  $d^l \leftarrow d_1$ ▷ Initialize search boundaries
  $d^r \leftarrow d_N$
  $g \leftarrow \tau + 1$ ▷ Initialize the derivative
  while $|g| > \tau$ **do**
    $d^m \leftarrow (d^l + d^r)/2.0$ ▷ Compute the mid point
    $g \leftarrow \sum_i p_i^m \texttt{Sign}(d^m - d_i)(1 - \exp -\frac{|d^m - d_i|}{\sigma})$ ▷ Compute the derivative by Eq.(5.6)
    if $g > 0$ **then** ▷ Update search boundaries
      $d^r \leftarrow d^m$
    else
      $d^l \leftarrow d^m$
    end if
  end while
  return d^m ▷ Return the mid point

By organizing the terms, we obtain

$$\frac{dy}{d\mathbf{p}^m} = -\frac{\partial G/\partial \mathbf{p}^m}{\partial G/\partial y} = [\ldots, \frac{\sigma \texttt{Sign}(d_i - y)(1 - \exp -\frac{|y-d_i|}{\sigma})}{\sum_j p_j^m \exp -\frac{|y-d_j|}{\sigma}}, \ldots]^T. \quad (5.8)$$

We clip the denominator, *i.e.*, $\sum_j p_j^m \exp -\frac{|y-d_j|}{\sigma}$ in the above equation to be no less than 0.1 to avoid large gradients.

5.3.4 *Network Architecture*

To find the disparity value, we match the image patches of left and right images by constructing stereo cost volumes, as in [104] and [26]. However, an exhaustive matching requires extensive memory and computation. For efficiency, we adopt a cascade structure following [70]. Specifically, we first sample the disparity hypothesis by a coarse matching, which is performed on low-resolution image features. The sampled hypothesis reduce the search space for matching to a large extent. Then we refine the sampled hypothesis at high-resolution image features. The overall pipeline is shown in Fig. 5.3, and includes 5 parts: (a) feature extraction (b) disparity hypotheses sampling (c) matching

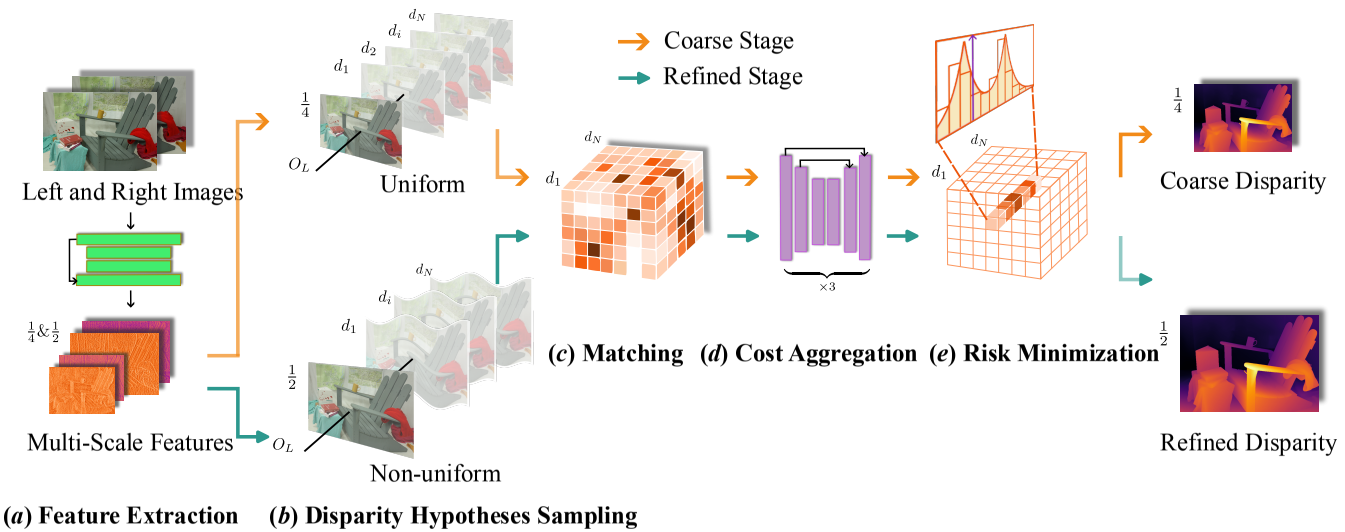


Figure 5.3: **Overall pipeline (Left to Right).** We first extract multi-scale features from left and right images respectively. The subsequent procedures are divided into two stages. In the coarse stage —shown in orange arrow, we sample disparity hypotheses uniformly and match on 1/4-resolution features. While in the refined stage—shown in green arrow, to match 1/2-resolution features efficiently. Disparity hypotheses are sampled centering around the disparity predicted from the coarse stage. In both stages, we first construct cost volumes by concatenation, and then apply the stacked hourglass networks to aggregate the matching cost, and finally search for the disparity that minimizes the proposed L^1 risk in Eq.(5.5).

(d) cost aggregation (e) risk minimization. We introduce the details of each part below.

***(a)* Feature Extraction.** Given an input image, the module aims to output multi-scale 2D feature maps. More specifically, we first use a ResNet [78] to extract 2D feature maps of resolution $1/4$ and $1/2$ with respect to the input image. The ResNet contains 4 stages of transformation with 3, 16, 3, 3 residual blocks respectively. And the spatial resolution is downsampled before the beginning of the first and third stages of transformation. Then we apply the spatial pyramid pooling [266] on the $1/4$-resolution feature map from the fourth stage to enlarge the receptive field. In the end, we upsample the enhanced feature map from $1/4$ to $1/2$ and fuse it with the $1/2$-resolution feature map from ResNet. The final outputs are the feature maps of $1/4$ and $1/2$ resolution. We apply the same network and weights to extract features from left and right images.

***(b)* Disparity Hypotheses Sampling.** The disparity hypotheses provide the candidates of pixel pairs to match. In the coarse stage, we sample 192 hypotheses uniformly within the range from 0 to the maximum possible disparity. In the refined stage, we reduce the sampling space according to the predicted disparity from the coarse stage. Specifically, for each pixel we sample 16 hypotheses between the maximum and minimum disparity in the local window of size 12×12.

***(c)* Matching.** We match the 2D feature maps from the left and right images according to the sampled disparity hypothesis. The features at each pair of candidates pixels for matching will be concatenated along the channel dimension, which forms a 4D stereo cost volume (feature$\times$ disparity$\times$height$\times$width). In the coarse stage, we match the feature map of $1/4$ resolution for efficiency. To capture high-frequency details, we match the $1/2$-resolution feature map in the refined stage.

***(d)* Cost Aggregation.** We use the stacked hourglass architecture [171] to transform the stereo cost volume and aggregate the matching cost. For the coarse and refined stages, the structures are the same except for the number of feature channels. Specifically, the network consists of three 3D hourglasss as in [26]. Each hourglass first downsamples the volume hierarchically to $1/2$ and $1/4$ resolution with respect to the input volume, and then upsample in sequence to recover the resolution. The procedure helps aggregate information across various scales. The final output is a volume that represents the discrete distribution of disparity hypotheses.

***(e)* Risk Minimization.** The module applies Alg. 1 to compute the optimal continuous disparity for each pixel given the discrete distribution of disparity hypotheses. During training, we additionally compute the gradient according to Eq.(5.8) to enable backward propagation.

5.3.5 *Loss Function*

Given the predicted disparity $x^{\texttt{pred}} \in \mathbb{R}$ and the ground-truth disparity $x^{\texttt{gt}} \in \mathbb{R}$, we compute the smooth L^1 loss [66]:

$$\mathcal{L}(x^{\texttt{gt}}, x^{\texttt{pred}}) = \begin{cases} 0.5(x^{\texttt{gt}} - x^{\texttt{pred}})^2 & \text{if } |x^{\texttt{gt}} - x^{\texttt{pred}}| < 1.0 \\ |x^{\texttt{gt}} - x^{\texttt{pred}}| - 0.5 & \text{otherwise} \end{cases} \tag{5.9}$$

We apply the above loss function to the predicted disparities from both the coarse and refined stages, and obtain $\mathcal{L}_{\texttt{coarse}}$ and $\mathcal{L}_{\texttt{refined}}$ respectively. The total loss $\mathcal{L} = 0.1\mathcal{L}_{\texttt{coarse}} + 1.0\mathcal{L}_{\texttt{refined}}$.

5.4 EXPERIMENTS AND RESULTS

Implementation Details. We implement our method in PyTorch 2.0.1 (Python 3.11.2) with CUDA 11.8. The software is evaluated on a computing machine with GeForce-RTX-3090 GPU.

Datasets. We perform experiments on four datasets namely SceneFlow [161], KITTI 2012 & 2015 [64, 163], Middlebury 2014 [200], and ETH 3D [203]. **(a) SceneFlow** is a synthetic dataset containing 35,454 image pairs for training, and 4,370 image pairs for test. **(b) KITTI 2012 & 2015** are captured for autonomous driving. There are 194 training image pairs and 195 test image pairs in KITTI 2012. And there are 200 training image pairs and 200 test image pairs in KITTI 2015. **(c) Middlebury 2014** is an indoor dataset including 15 image pairs for training. **(d) ETH 3D** is a gray-scale dataset providing 27 image pairs for training.

Training Details. We train our network on SceneFlow. The weight is initialized randomly. We use AdamW optimizer [157] with weight decay 10^{-5}. The learning rate decreases from 2×10^{-4} to 2×10^{-8} according to the one cycle learning rate policy. We train the network for 2×10^5 iterations. The images will be randomly cropped to 320×736. For KITTI 2012 & 2015 benchmarks, we further fine tune the network

Table 5.1: Comparison with state-of-the-art methods on SceneFlow test set. The first and second bests are in red and blue respectively. **Our method** in bold.

Method	Param (M)	Time (s)	EPE ↓	> 0.5px ↓	> 1px ↓	> 2px ↓
CFNet [204]	21.98	0.13	1.04	15.91	10.30	6.89
PCWNet [205]	34.27	0.25	0.90	17.59	8.08	4.57
ACVNet [243]	6.84	0.16	0.47	9.70	5.00	2.74
DLNR [265]	54.72	0.44	0.53	8.75	5.44	3.44
IGEV [244]	12.60	0.36	0.47	8.51	5.21	3.26
Ours	**11.96**	**0.35**	**0.43**	**8.10**	**4.22**	**2.34**

on the training image pairs for 2.5×10^3 iterations. The learning rate starts from 5×10^{-5} to 5×10^{-9}.

5.4.1 *In-Domain Evaluation*

Tab.(5.1), Tab.(5.2) and Tab.(5.3) provide statistical comparison results with the competing methods on SceneFlow, KITTI 2012 & 2015 bechmarks, respectively. All the methods have been trained or fine-tuned on the corresponding training set. In SceneFlow test set, our proposed approach shows the best results for all the evaluation metrics. Particularly, we reduce the $>$ 1px error from 5.00 to 4.22, and the $>$ 0.5px error from 8.51 to 8.10. In KITTI 2012 & 2015 benchmarks, the matching accuracy of our approach in the non-occluded regions rank the first among the published methods. Especially, in KITTI 2012, we reduce the $>$ 2px error in non-occluded regions by 0.11.

5.4.2 *Cross-Domain Generalization*

In this part, we compare the methods when dealing with environments never seen in the training set. Specifically, all methods are trained only on SceneFlow training set, and then evaluated on the training set of Middlebury, ETH 3D and KITTI 2012 & 2015 *without* fine-tuning.

The statistical comparison results are shown in Tab.(5.4), Tab.(5.5), Tab.(5.6) and Tab.(5.7), respectively. Our proposed approach achieves the first or the second best accuracies under all the evaluation metrics on the four real-world datasets. Particularly, for Middlebury we reduce the $>$ 1px error from 13.76 to 12.63. Further more, on ETH 3D we

Table 5.2: Comparison with state-of-the-art methods on KITTI 2012 Benchmark. † denotes using extra data for pre-training. The first and second bests are in red and blue respectively. **Our method** in bold. The results are obtained from KITTI official website.

Method	Param (M)	Time (s)	> 2px		> 3px	
			Noc	All	Noc	All
LEAStereo [37]	1.81		1.90	2.39	1.13	1.45
CFNet [204]	21.98	0.12	1.90	2.43	1.23	1.58
ACVNet [243]	6.84	0.15	1.83	2.34	1.13	1.47
ACFNet [31]			1.83	2.35	1.17	1.54
NLCA-Net v2 [188]			1.83	2.34	1.11	1.46
CAL-Net [31]			1.74	2.24	1.19	1.53
CREStereo [128] †			1.72	2.18	1.14	1.46
LaC+GANet [140]	9.43		1.72	2.26	1.05	1.42
IGEV [244]†	12.60	0.32	1.71	2.17	1.12	1.44
PCWNet [205]	34.27	0.23	1.69	2.18	1.04	1.37
Ours	**11.96**	**0.32**	**1.58**	**2.20**	**1.00**	**1.44**

Table 5.3: Comparison with state-of-the-art methods on KITTI 2015 Benchmark. † denotes using extra data for pre-training. The first and second bests are in red and blue respectively. **Our method** in bold. The results are obtained from KITTI official website.

Method	Param (M)	Time (s)	All			Noc		
			D1_bg	D1_fg	D1_all	D1_bg	D1_fg	D1_all
LEAStereo [37]	1.81		1.40	2.91	1.65	1.29	2.65	1.51
CFNet [204]	21.98	0.12	1.54	3.56	1.88	1.43	3.25	1.73
ACVNet [243]	6.84	0.15	1.37	3.07	1.65	1.26	2.84	1.52
ACFNet [31]			1.51	3.80	1.89	1.36	3.49	1.72
NLCA-Net v2 [188]			1.41	3.56	1.77	1.28	3.22	1.60
CAL-Net [31]			1.59	3.76	1.95	1.45	3.42	1.77
CREStereo [128] †			1.45	2.86	1.69	1.33	2.60	1.54
LaC+GANet [140]	9.43		1.44	2.83	1.67	1.26	2.64	1.49
IGEV [244] †	12.60	0.32	1.38	2.67	1.59	1.27	2.62	1.49
DLNR [265]	54.72	0.39	1.60	2.59	1.76	1.45	2.39	1.61
PCWNet [205]	34.27	0.23	1.37	3.16	1.67	1.26	2.93	1.53
CroCo-Stereo [233]†	417.15		1.38	2.65	1.59	1.30	2.56	1.51
Ours	**11.96**	**0.32**	**1.40**	**2.76**	**1.63**	**1.25**	**2.62**	**1.48**

Table 5.4: Cross-domain evaluation on Middlebury training set of quarter resolution. † denotes using extra data for pre-training. The first and second bests are in red and blue respectively. **Our method** in bold. All methods are trained on SceneFlow and evaluated on Middlebury training set without fine-tuning.

Method	Param (M)	Time (s)	> 0.5px		> 1px	
			Noc	All	Noc	All
CFNet [204]	21.98	0.11	29.50	34.30	17.85	22.16
ACVNet [243]	6.84	0.12	39.04	42.97	22.68	26.49
DLNR [265]	12.60	0.63	19.43	23.75	10.16	13.76
IGEV [244]†	12.60	0.34	19.05	23.33	10.44	14.05
PCWNet [205]	34.27	0.19	33.33	38.00	16.80	21.36
Ours	**11.96**	**0.25**	**19.22**	**23.33**	**9.32**	**12.63**

reduce the > 0.5px error from 10.39 to 8.59, and > 1px error from 4.05 to 2.71. It can be observed our approach is more robust and generalizes better than recent state of the arts on the cross-domain setting.

5.4.3 *Ablation Studies*

In this subsection, we perform ablation studies to analyze the effects of the risk minimization method for disparity prediction. All the models are trained on SceneFlow and then tested on Middlebury *without* fine-tuning.

***(a)* Effect of Risk Minimization.** We compare the expectation and the L^1-norm risk minimization for disparity prediction during training and test. We present the comparison results in Tab.(5.8). Even using the expectation to predict disparities during training, we still slightly improve the accuracy by changing to the L^1-norm risk minimization during test. Moreover, if we use the L^1-norm risk minimization in both training and test, the best accuracy is achieved under all metrics.

***(b)* Performance with Different Networks.** We replace the disparity prediction method in ACVNet [243] and PCWNet [205] from expectation to L^1-norm risk minimization *only* during test. The results are shown in Tab.(5.8). Our proposed method improves the accuracy under all metrics *without* re-training.

5.4.4 *Network Processing Time & Paremeters*

We present the networks' inference time and number of parameters in Tab.(5.1), Tab.(5.2), Tab.(5.4), and Tab.(5.5). For a fair comparison, all networks are evalutated on the same machine with a GeForce-RTX-3090 GPU. Our network outperforms many state of the arts on inference time, including IGEV and DLNR. Moreover, our network has fewer learnable parameters than PCWNet, IGEV and DLNR.

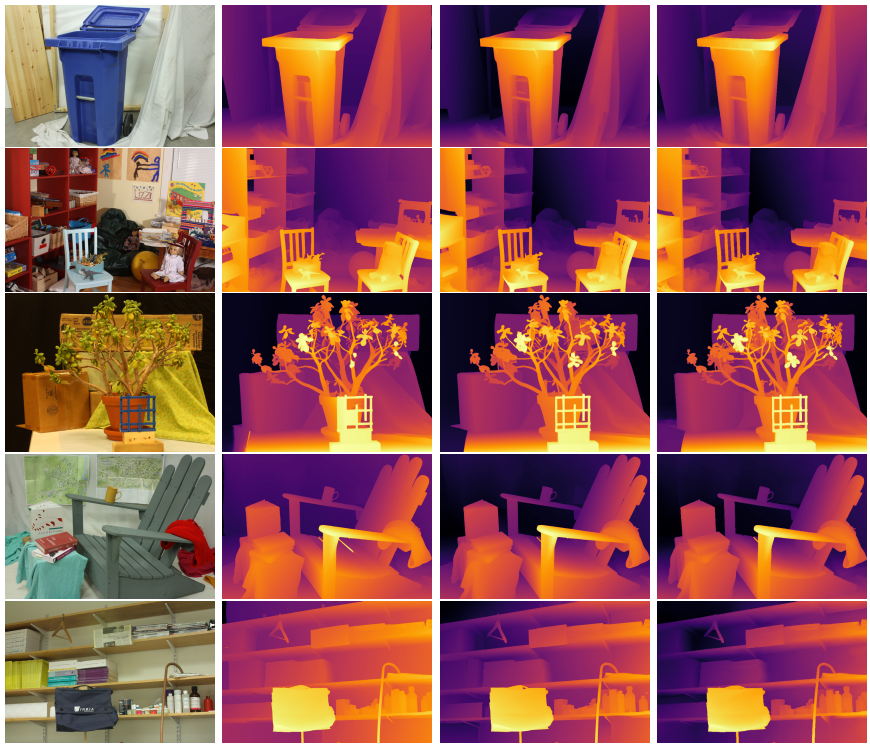
In addition, our proposed L^1-norm risk minimization module doesn't require extra learnable parameters. The running time is shown in Tab.(5.8). By changing the disparity prediction method from expectation to our proposed approach, the running time is slightly increased.

5.4.5 *Qualitative Results*

In this section, we present more qualitative results on Middlebury in Fig. 5.4. It can be observed that in general our method generalizes and predicts high-frequency details better than other recent methods.

5.4.6 *Conclusion*

Our work provides a novel way of thinking and solving stereo-matching problems in computer vision via the principle of risk minimization [226]. The paper provides in-depth theoretical and practical benefits of using our proposed formulation. It is shown that the presented approach is more robust to multi-modal distributions and outliers, and generalizes better on cross-domain stereo images. Furthermore, a new mathematical fabric to research stereo-matching problems is presented, enabling adaptations from fields such as robotics and control engineering.



(a) Image (b) IGEV (c) DLNR (d) Ours

Figure 5.4: **Qualitative Comparison.** We compare our method with recent state-of-the-art methods such as IGEV [244], DLNR [265] on Middlebury [200]. All methods are trained only on SceneFlow [161], and evaluated at quarter resolution.

Table 5.5: Cross-domain evaluation on ETH 3D training set. † denotes using extra data for pre-training. The first and second bests are in red and blue respectively. **Our method** in bold. All methods are trained on SceneFlow and evaluated on ETH 3D training set without fine-tuning.

Method	Param (M)	Time (s)	> 0.5px		> 1px	
			Noc	All	Noc	All
CFNet [204]	21.98	0.11	15.57	16.24	5.30	5.59
ACVNet [243]	6.84	0.12	21.83	22.64	8.13	8.81
DLNR [265]	12.60	0.34	18.66	19.07	13.11	13.39
IGEV [244]†	12.60	0.29	9.83	10.39	3.60	4.05
PCWNet [205]	34.27	0.20	18.25	18.88	5.17	5.43
Ours	**11.96**	**0.26**	**7.90**	**8.59**	**2.41**	**2.71**

Table 5.6: Cross-domain evaluation on KITTI 2012 training set. † denotes using extra data for pre-training. The first and second bests are in red and blue respectively. **Our method** in bold. All methods are trained on SceneFlow and evaluated on KITTI 2012 training set without fine-tuning.

Method	Param (M)	Time (s)	> 2px		> 3px	
			Noc	All	Noc	All
CFNet [204]	21.98	0.12	7.08	7.97	4.66	5.31
ACVNet [243]	6.84	0.15	20.34	21.44	14.22	15.18
DLNR [265]	12.60	0.39	12.01	12.81	8.83	9.46
IGEV [244]†	12.60	0.32	7.55	8.44	5.03	5.70
PCWNet [205]	34.27	0.23	6.63	7.49	4.08	4.68
Ours	**11.96**	**0.32**	**5.82**	**6.70**	**3.84**	**4.43**

Table 5.7: Cross-domain evaluation on KITTI 2015 training set. † denotes using extra data for pre-training. The first and second bests are in red and blue respectively. **Our method** in bold. All methods are trained on SceneFlow and evaluated on KITTI 2015 training set without fine-tuning.

Method	Param (M)	Time (s)	All			Noc		
			D1_bg	D1_fg	D1_all	D1_bg	D1_fg	D1_all
CFNet [204]	21.98	0.12	4.77	13.26	6.07	4.64	12.88	5.88
ACVNet [243]	6.84	0.15	12.35	19.97	13.52	12.04	18.82	13.06
DLNR [265]	9.43	0.39	18.67	14.86	18.08	18.42	14.18	17.78
IGEV [244] †	12.60	0.32	4.01	15.58	5.79	3.88	14.94	5.55
PCWNet [205]	34.27	0.23	4.25	14.40	5.81	4.11	13.95	5.60
Ours	**11.96**	**0.32**	**3.68**	**13.52**	**5.19**	**3.57**	**13.05**	**5.00**

Table 5.8: Ablation studies on Middlebury training set of quarter resolution. The first and second bests are in red and blue respectively. **Our method** in bold. All methods are trained on SceneFlow and evaluated on Middlebury training set without fine-tuning.

Backbone	Training	Test	Param(M)	Time(s)	> 1px		> 2px	
					Noc	All	Noc	All
ACVNet[243]	Expectation	Expectation	6.84	0.12	22.68	26.49	13.54	16.49
	Expectation	L1-Risk	6.84	0.18	22.32	26.14	13.13	16.05
PCWNet[205]	Expectation	Expectation	34.27	0.19	16.80	21.36	8.93	12.62
	Expectation	L1-Risk	34.27	0.26	16.53	21.08	8.65	12.30
Ours	Expectation	Expectation	11.96	0.17	9.88	13.27	4.92	7.29
	Expectation	L1-Risk	11.96	0.25	9.83	13.22	4.90	7.27
	L1-Risk	Expectation	11.96	0.17	9.83	13.19	4.79	7.06
	L1-Risk	**L1-Risk**	**11.96**	**0.25**	**9.32**	**12.63**	**4.49**	**6.70**

6 CONCLUSION AND OUTLOOK

6.1 CONCLUSION

In this thesis, we investigate the problem of depth perception from a single or stereo images. Focusing on the learning-based approaches, novel perspectives on thinking and modeling the depth or disparity are presented, which are not only theoretically compelling but also facilitate the design of neural networks and loss functions to enable higher-quality depth perception.

In Chapter 2, we suitably formalize the connection between robust statistical modeling techniques, *i.e.*, multivariate covariance modeling with low-rank approximation, and popular loss functions in neural network-based SIDP problem. The novelty presented in this chapter arises from the fact that the proposed pipeline and loss term turns out to be more general, hence could be helpful in the broader application of SIDP in several tasks, such as depth uncertainty for robot vision, control and others. Remarkably, the proposed formulation is not only theoretically compelling but observed to be practically beneficial, resulting in a loss function that is used to train the proposed network showing state-of-the-art SIDP results on several benchmark datasets.

In Chapter 3, we introduce a simple and effective approach to exploit the rigid scene prior in depth perception tasks by making use of the variational constraint. Our approach does not make explicit assumptions about the scene other than the scene gradient regularity, which holds for typical indoor or outdoor scenes. When tested on popular benchmark datasets, our method shows significantly better results than the prior art, both qualitatively and quantitatively.

In Chapter 4, we show that line structures in scenes provide valuable information for depth perception. Furthermore, we present a simple architecture to exploit the lines inside ConvNets. Our method advances state-of-the-art results on single-image 3D object detection and depth prediction benchmarks.

In Chapter 5, we turn attention to stereo images and introduce a novel way of thinking and solving stereo matching problems in com-

puter vision via the principle of risk minimization [226]. We provide in-depth theoretical and practical benefits of using our proposed formulation. It is shown that the presented approach is more robust to multi-modal distributions and outliers, and generalizes better on cross-domain stereo images. Furthermore, a new mathematical fabric to research stereo matching problems is presented, enabling adaptations from fields such as robotics and control engineering.

6.2 FUTURE WORK

Considering the inherent challenges in image-based depth perception and the existing limitations of the proposed methods, there are several promising research directions that warrant further exploration in the future.

6.2.1 *Mixture of Gaussian for Depth*

In practical applications, the real distribution of depth map given a single image might be asymmetric or multi-modal. Yet, the Gaussian assumption in our approach is too simple to cope with the complex situations. Hence, it would be beneficial to extend our approach with more powerful mathematical models. One straightforward solution is to make use of the mixture of Gaussian distributions, which is known as a universal approximator of densities. The advantage is that with more components, we could provide a more precise distribution about the depth given the input image. Moreover, the likelihood might be a more effective loss function to supervise the training procedure.

6.2.2 *Depth Map Generation*

It's well known that single-image depth prediction is an ill-posed problem. Because there are many possible 3D configurations that could produce the same 2D image after projection onto the camera plane. Nonetheless, one natural question is how to sample possible depth maps and evaluate the quality of the samples? We pursue this problem not only for a scientific thrill but mainly because there are several possible real-world applications. For example, we could synthesize diverse novel views of the image by generating multiple depth maps.

Although there has already been numerous research on image generation, the depth map has distinguished properties from the image, and might require special algorithms for generation and evaluation. Therefore it is a meaningful research direction to design generative models and evaluation metrics for depth map generation.

6.2.3 *Relation between Depth and Semantics*

The semantic attributes of the object are important cues for its depth, especially when only a single image is available. There has already been research showing that joint learning of semantic segmentation and single-image depth prediction helps improve the accuracy of the network on both tasks. Moreover, in our experiments we also observe that pretraining on ImageNet is critical to the learning of single-image depth prediction. In summary, the above empirical results show a strong connection between the depth and semantics. However, there is little theoretical analysis about the relation, and it is unknown how to maximize the benefits of the interaction. One possible research direction is to formulate the semantics into the depth perception framework. It might inspire us to design more effective network structures, loss functions and training strategies.

6.2.4 *Fusion with Special Sensors*

In recent years various novel sensors have been proposed to perceive depth more efficiently. The most popular ones include LiDAR and Kinect. It's noteworthy that they are suitable for different scenarios. LiDAR is often equipped on autonomous cars, and can perceive obstacles in the long distance. While Kinect is widely used in indoor devices, such as interactive gaming. Nonetheless, the above sensors also have drawbacks. For example, LiDAR can provide only sparse measurements. While the depth map from Kinect has noises. A meaningful direction is to fuse the predicted depth from images with the depth measurements from special sensors. Because the depth map from a single image is often dense and smooth, but lacks the absolute scale. It is possible to obtain a better depth map by combining the camera with LiDAR or Kinect.

BIBLIOGRAPHY

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. "Image2stylegan++: How to edit the embedded images?" In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8296–8305 (cit. on p. 1).

[2] Ashutosh Agarwal and Chetan Arora. "Attention Attention Everywhere: Monocular Depth Prediction with Skip Attention." In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023, pp. 5861–5870 (cit. on p. 7).

[3] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. "Building rome in a day." In: *Communications of the ACM* 54.10 (2011), pp. 105–112 (cit. on p. 7).

[4] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, Mannat Kaur, and Bingbing Liu. "Bidirectional Attention Network for Monocular Depth Estimation." In: *IEEE International Conference on Robotics and Automation (ICRA)* (2021) (cit. on pp. 53, 64).

[5] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, and Mannat Kaur Bingbing Liu. "Bidirectional attention network for monocular depth estimation." In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 11746–11752 (cit. on pp. 21, 31, 43).

[6] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. "Deep evidential regression." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14927–14937 (cit. on pp. 8, 11).

[7] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization." In: *arXiv preprint arXiv:1607.06450* (2016) (cit. on p. 25).

[8] Abhishek Badki, Alejandro Troccoli, Kihwan Kim, Jan Kautz, Pradeep Sen, and Orazio Gallo. "Bi3d: Stereo depth estimation via binary classifications." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1600–1608 (cit. on p. 68).

[9] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. "IronDepth: Iterative Refinement of Single-View Depth using Surface Normal and its Uncertainty." In: *British Machine Vision Conference (BMVC)*. 2022 (cit. on p. 11).

[10] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*. Vol. 463. 1999. ACM press New York, 1999 (cit. on p. 1).

[11] Timur Bagautdinov, Francois Fleuret, and Pascal Fua. "Probability occupancy maps for occluded depth images." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2829–2837 (cit. on p. 1).

[12] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. "PatchMatch: A randomized correspondence algorithm for structural image editing." In: *ACM Trans. Graph.* 28.3 (2009), p. 24 (cit. on p. 1).

[13] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. "To understand deep learning we need to understand kernel learning." In: *International Conference on Machine Learning*. PMLR. 2018, pp. 541–549 (cit. on p. 3).

[14] James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013 (cit. on p. 66).

[15] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. "Adabins: Depth estimation using adaptive bins." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4009–4018 (cit. on pp. 8, 11, 21, 22, 24, 31, 33, 43–45, 47, 48).

[16] Keshav Bimbraw. "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology." In: *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*. Vol. 1. IEEE. 2015, pp. 191–198 (cit. on p. 65).

[17] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, Jan. 2006 (cit. on pp. 16, 71).

[18] Michael Bleyer, Christoph Rhemann, and Carsten Rother. "Patchmatch stereo-stereo matching with slanted support windows." In: *Bmvc*. Vol. 11. 2011, pp. 1–11 (cit. on p. 65).

[19] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013 (cit. on p. 1).

[20] Garrick Brazil and Xiaoming Liu. "M3d-rpn: Monocular 3D region proposal network for object detection." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9287–9296 (cit. on pp. 58, 60, 61, 63).

[21] Tim Brooks, Aleksander Holynski, and Alexei A Efros. "Instructpix2pix: Learning to follow image editing instructions." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 18392–18402 (cit. on p. 1).

[22] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. "nuscenes: A multimodal dataset for autonomous driving." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631 (cit. on p. 1).

[23] Changjiang Cai, Matteo Poggi, Stefano Mattoccia, and Philippos Mordohai. "Matching-space stereo networks for cross-domain generalization." In: *2020 International Conference on 3D Vision (3DV)*. IEEE. 2020, pp. 364–373 (cit. on p. 69).

[24] Ang Cao, Chris Rockwell, and Justin Johnson. "Fwd: Real-time novel view synthesis with forward warping and depth." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 15713–15724 (cit. on p. 1).

[25] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. "An introduction to total variation for image analysis." In: *Theoretical foundations and numerical methods for sparse recovery* 9.263-340 (2010), p. 227 (cit. on p. 32).

[26] Jia-Ren Chang and Yong-Sheng Chen. "Pyramid stereo matching network." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5410–5418 (cit. on pp. 65, 66, 68, 73, 75).

[27] Tianyu Chang, Xun Yang, Tianzhu Zhang, and Meng Wang. "Domain Generalized Stereo Matching via Hierarchical Visual Transformation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 9559–9568 (cit. on p. 69).

[28] Wenjie Chang, Yueyi Zhang, and Zhiwei Xiong. "Transformer-based Monocular Depth Estimation with Attention Supervision." In: *32nd British Machine Vision Conference (BMVC 2021)*. 2021 (cit. on pp. 11, 21, 22, 43, 44).

[29] Chuangrong Chen, Xiaozhi Chen, and Hui Cheng. "On the Over-Smoothing Problem of CNN Based Disparity Estimation." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on pp. 66, 69).

[30] Duo Chen, Zixin Tang, Zhenyu Xu, Yunan Zheng, and Yiguang Liu. "Gaussian Fusion: Accurate 3D Reconstruction via Geometry-Guided Displacement Interpolation." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5916–5925 (cit. on p. 8).

[31] Shenglun Chen, Baopu Li, Wei Wang, Hong Zhang, Haojie Li, and Zhihui Wang. "Cost affinity learning network for stereo matching." In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 2120–2124 (cit. on p. 78).

[32] Tian Chen, Shijie An, Yuan Zhang, Chongyang Ma, Huayan Wang, Xiaoyan Guo, and Wen Zheng. "Improving Monocular Depth Estimation by Leveraging Structural Awareness and Complementary Datasets." In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 90–108 (cit. on p. 54).

[33] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. "Single-image depth perception in the wild." In: *Advances in neural information processing systems* 29 (2016) (cit. on pp. 3, 10, 32, 34).

[34] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. "Multi-View 3D Object Detection Network for Autonomous Driving." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 58).

[35] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. "MonoPair: Monocular 3D Object Detection Using Pairwise Spatial Relationships." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 (cit. on pp. 62, 63).

[36] Xinjing Cheng, Peng Wang, and Ruigang Yang. "Depth estimation via affinity learned with convolutional spatial propagation network." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 103–119 (cit. on p. 35).

[37] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. "Hierarchical neural architecture search for deep stereo matching." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 22158–22169 (cit. on p. 78).

[38] Kyunghyun Cho, Bart Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." In: *EMNLP*. 2014 (cit. on p. 68).

[39] Jack Choquette. "Nvidia hopper h100 gpu: Scaling performance." In: *IEEE Micro* (2023) (cit. on p. 2).

[40] Jack Choquette, Wishwesh Gandhi, Olivier Giroux, Nick Stam, and Ronny Krashinsky. "Nvidia a100 tensor core gpu: Performance and innovation." In: *IEEE Micro* 41.2 (2021), pp. 29–35 (cit. on p. 2).

[41] Dionysios C Christodouleas, Alex Nemiroski, Ashok A Kumar, and George M Whitesides. "Broadly available imaging devices enable high-quality low-cost photometry." In: *Analytical chemistry* 87.18 (2015), pp. 9170–9178 (cit. on p. 1).

[42] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009 (cit. on p. 72).

[43] Marco Cuturi. "Sinkhorn distances: Lightspeed computation of optimal transport." In: *Advances in neural information processing systems* 26 (2013) (cit. on p. 68).

[44] C.M. da Fonseca and J. Petronilho. "Explicit inverses of some tridiagonal matrices." In: *Linear Algebra and its Applications* 325.1 (2001), pp. 7–21. URL: https://www.sciencedirect.com/science/article/pii/S0024379500002895 (cit. on p. 16).

[45] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. "Image retrieval: Ideas, influences, and trends of the new age." In: *ACM Computing Surveys (Csur)* 40.2 (2008), pp. 1–60 (cit. on p. 1).

[46] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255 (cit. on pp. 2, 23, 42).

[47] Thomas Deselaers, Daniel Keysers, and Hermann Ney. "Features for image retrieval: an experimental comparison." In: *Information retrieval* 11 (2008), pp. 77–107 (cit. on p. 1).

[48] Raul Diaz and Amit Marathe. "Soft labels for ordinal regression." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4738–4747 (cit. on pp. 53, 64).

[49] Tom van Dijk and Guido de Croon. "How Do Neural Networks See Depth in Single Images?" In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on p. 51).

[50] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." In: *arXiv preprint arXiv:2010.11929* (2020) (cit. on p. 45).

[51] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." In: *International Conference on Learning Representations*. 2021 (cit. on pp. 2, 68).

[52] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, et al. "DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality." In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 2020, pp. 829–843 (cit. on p. 31).

[53] Richard O. Duda and Peter E. Hart. "Use of the Hough Transformation to Detect Lines and Curves in Pictures." In: *Commun. ACM* 15.1 (Jan. 1972), pp. 11–15 (cit. on pp. 5, 52, 55).

[54] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. "DeepPruner: Learning Efficient Stereo Matching via Differentiable PatchMatch." In: *ICCV*. 2019 (cit. on p. 68).

[55] David Eigen, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network." In: *Advances in neural information processing systems* 27 (2014) (cit. on pp. 3, 22, 23, 25, 26, 31, 41, 42, 44, 53, 62, 63).

[56] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge." In: *International journal of computer vision* 88 (2010), pp. 303–338 (cit. on p. 2).

[57] Rui Fan, Li Wang, Mohammud Junaid Bocus, and Ioannis Pitas. "Computer stereo vision for autonomous driving." In: *arXiv preprint arXiv:2012.03194* (2020) (cit. on p. 65).

[58] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. "Deep ordinal regression network for monocular depth estimation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2002–2011 (cit. on pp. 10, 21, 22, 31, 43, 44, 53, 64).

[59] Yasutaka Furukawa, Carlos Hernández, et al. "Multi-view stereo: A tutorial." In: *Foundations and Trends® in Computer Graphics and Vision* 9.1-2 (2015), pp. 1–148 (cit. on p. 7).

[60] Yasutaka Furukawa and Jean Ponce. "Accurate, dense, and robust multiview stereopsis." In: *IEEE transactions on pattern analysis and machine intelligence* 32.8 (2009), pp. 1362–1376 (cit. on p. 31).

[61] Yarin Gal et al. "Uncertainty in deep learning." In: () (cit. on pp. 11, 15, 16).

[62] Divyansh Garg, Yan Wang, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. "Wasserstein distances for stereo disparity estimation." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 22517–22529 (cit. on pp. 66, 69).

[63] William C Gartner. "Image formation process." In: *Journal of travel & tourism marketing* 2.2-3 (1994), pp. 191–216 (cit. on p. 1).

[64] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3354–3361 (cit. on pp. 1, 3, 21–23, 25, 34, 42, 43, 52, 58, 62, 68, 76).

[65] Ross Girshick. "Fast R-CNN." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015 (cit. on p. 58).

[66] Ross Girshick. "Fast R-CNN." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015 (cit. on p. 76).

[67] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013 (cit. on p. 16).

[68] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. Vol. 1. MIT Press, 2016 (cit. on p. 2).

[69] "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 2).

[70] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. "Cascade cost volume for high-resolution multi-view stereo and stereo matching." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2495–2504 (cit. on pp. 68, 73).

[71] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. "Group-wise Correlation Stereo Network." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3273–3282 (cit. on p. 68).

[72] Shir Gur and Lior Wolf. "Single image depth estimation trained via depth from defocus cues." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7683–7692 (cit. on p. 51).

[73] Mohammad Mahdi Haji-Esmaeili and Gholamali Montazer. "Playing for depth." In: *arXiv preprint arXiv:1810.06268* (2018) (cit. on p. 7).

[74] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. "Enhanced computer vision with microsoft kinect sensor: A review." In: *IEEE transactions on cybernetics* 43.5 (2013), pp. 1318–1334 (cit. on p. 2).

[75] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. "A survey on vision transformer." In: *IEEE transactions on pattern analysis and machine intelligence* 45.1 (2022), pp. 87–110 (cit. on p. 2).

[76] Qi Han, Kai Zhao, Jun Xu, and Ming-Ming Cheng. "Deep Hough Transform for Semantic Line Detection." In: *ECCV*. 2020, pp. 750–766 (cit. on pp. 56, 57, 59, 63).

[77] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003 (cit. on pp. 1, 2, 51).

[78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 31, 56, 58, 63, 75).

[79] Lukas Hewing, Juraj Kabzan, and Melanie N Zeilinger. "Cautious model predictive control using gaussian process regression." In: *IEEE Transactions on Control Systems Technology* 28.6 (2019), pp. 2736–2743 (cit. on p. 8).

[80] Geoffrey E Hinton and Ruslan R Salakhutdinov. "Reducing the dimensionality of data with neural networks." In: *science* 313.5786 (2006), pp. 504–507 (cit. on p. 2).

[81] Heiko Hirschmuller. "Stereo processing by semiglobal matching and mutual information." In: *IEEE Transactions on pattern analysis and machine intelligence* 30.2 (2007), pp. 328–341 (cit. on pp. 2, 65, 68).

[82] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cit. on p. 2).

[83] Sepp Hochreiter and Jürgen Schmidhuber. "LSTM can solve hard long time lag problems." In: *Advances in neural information processing systems* 9 (1996) (cit. on p. 2).

[84] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. "Learning to learn using gradient descent." In: *Artificial Neural Networks—ICANN 2001: International Conference Vienna, Austria, August 21–25, 2001 Proceedings 11*. Springer. 2001, pp. 87–94 (cit. on p. 3).

[85] William Hoff and Narendra Ahuja. "Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection." In: *IEEE transactions on pattern analysis and machine intelligence* 11.2 (1989), pp. 121–136 (cit. on p. 65).

[86] Derek Hoiem, Alexei A Efros, and Martial Hebert. "Automatic photo pop-up." In: *ACM SIGGRAPH 2005 Papers*. Association for Computing Machinery, 2005, pp. 577–584 (cit. on pp. 7, 31).

[87] Berthold K. P. Horn. "Obtaining shape from shading information." In: *Shape from Shading*. Cambridge, MA, USA: MIT Press, 1989, pp. 123–171 (cit. on pp. 2, 51).

[88] Julia Hornauer and Vasileios Belagiannis. "Gradient-based Uncertainty for Monocular Depth Estimation." In: *European Conference on Computer Vision*. Springer. 2022, pp. 613–630 (cit. on p. 8).

[89] Yi-Zeng Hsieh and Shih-Syun Lin. "Robotic arm assistance system based on simple stereo matching and Q-learning optimization." In: *IEEE Sensors Journal* 20.18 (2020), pp. 10945–10954 (cit. on p. 65).

[90] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries." In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 1043–1051 (cit. on pp. 11, 34).

[91] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 60).

[92] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. "Deepmvs: Learning multi-view stereopsis." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2821–2830 (cit. on p. 31).

[93] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. "Guiding Monocular Depth Estimation Using Depth-Attention Volume." In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 581–597 (cit. on p. 54).

[94] Michel Jaboyedoff, Thierry Oppikofer, Antonio Abellán, Marc-Henri Derron, Alex Loye, Richard Metzger, and Andrea Pedrazzini. "Use of LIDAR in landslide investigations: a review." In: *Natural hazards* 61 (2012), pp. 5–28 (cit. on p. 2).

[95] Nishant Jain, Suryansh Kumar, and Luc Van Gool. "Robustifying the Multi-Scale Representation of Neural Radiance Fields." In: *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. URL: https://bmvc2022.mpi-inf.mpg.de/0578.pdf (cit. on p. 7).

[96] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. "Look Deeper into Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018 (cit. on p. 53).

[97] Sing Bing Kang, J.A. Webb, C.L. Zitnick, and T. Kanade. "A multibaseline stereo system with active illumination and real-time image acquisition." In: *Proceedings of IEEE International Conference on Computer Vision*. 1995, pp. 88–93 (cit. on p. 65).

[98] Abhishek Kar, Shubham Tulsiani, João Carreira, and Jitendra Malik. "Category-Specific Object Reconstruction from a Single Image." In: *Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on p. 1).

[99] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. "Generalization in deep learning." In: *arXiv preprint arXiv:1710.05468* 1.8 (2017) (cit. on p. 3).

[100] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. "Uncertainty-aware deep multi-view photometric stereo." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12601–12611 (cit. on pp. 7, 31).

[101] Alex Kendall and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 4, 7, 8, 11, 14, 16, 23).

[102] Alex Kendall, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7482–7491 (cit. on p. 4).

[103] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. "End-To-End Learning of Geometry and Context for Deep Stereo Regression." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017 (cit. on pp. 3, 65, 66, 68).

[104] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. "End-to-end learning of geometry and context for deep stereo regression." In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 66–75 (cit. on p. 73).

[105] Wan-Soo Kim, Dae-Hyun Lee, Yong-Joo Kim, Taehyeong Kim, Won-Suk Lee, and Chang-Hyun Choi. "Stereo-vision-based crop height estimation for agricultural robots." In: *Computers and Electronics in Agriculture* 181 (2021), p. 105937 (cit. on p. 65).

[106] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on pp. 23, 42).

[107] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015,*

Conference Track Proceedings. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980 (cit. on pp. 59, 63).

[108] Andreas Klaus, Mario Sormann, and Konrad Karner. "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure." In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 3. IEEE. 2006, pp. 15–18 (cit. on p. 65).

[109] Maria Klodt and Andrea Vedaldi. "Supervising the new with the old: learning sfm from sfm." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 698–713 (cit. on p. 8).

[110] Vladimir Kolmogorov and Ramin Zabih. "Computing visual correspondence with occlusions using graph cuts." In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. IEEE. 2001, pp. 508–515 (cit. on p. 65).

[111] Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002 (cit. on pp. 66, 72).

[112] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems* 25 (2012) (cit. on p. 31).

[113] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In: *Communications of the ACM* 60.6 (2017), pp. 84–90 (cit. on p. 2).

[114] Suryansh Kumar, Yuchao Dai, and Hongdong Li. "Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames." In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4649–4657 (cit. on pp. 7, 31).

[115] Suryansh Kumar, Yuchao Dai, and Hongdong Li. "Superpixel soup: Monocular dense 3d reconstruction of a complex dynamic scene." In: *IEEE transactions on pattern analysis and machine intelligence* 43.5 (2019), pp. 1705–1717 (cit. on pp. 7, 31).

[116] Mathieu Labbé and François Michaud. "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation." In: *Journal of Field Robotics* 36.2 (2019), pp. 416–446 (cit. on p. 31).

[117] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." In: *Advances in neural information processing systems* 30 (2017) (cit. on pp. 8, 11, 14, 16, 18, 25).

[118] Miguel Lázaro-Gredilla, Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. "Sparse spectrum Gaussian process regression." In: *The Journal of Machine Learning Research* 11 (2010), pp. 1865–1881 (cit. on p. 14).

[119] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series." In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995 (cit. on pp. 2, 68).

[120] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." In: *nature* 521.7553 (2015), pp. 436–444 (cit. on pp. 2, 3, 52, 56).

[121] Jae-Han Lee, Minhyeok Heo, Kyung-Rae Kim, and Chang-Su Kim. "Single-image depth estimation based on fourier domain analysis." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 330–339 (cit. on p. 54).

[122] Jae-Han Lee and Chang-Su Kim. "Monocular depth estimation using relative depth maps." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9729–9738 (cit. on p. 35).

[123] Jae-Han Lee and Chang-Su Kim. "Multi-loss rebalancing algorithm for monocular depth estimation." In: *European Conference on Computer Vision*. Springer. 2020, pp. 785–801 (cit. on p. 36).

[124] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. "From big to small: Multi-scale local planar guidance for monocular depth estimation." In: *arXiv preprint arXiv:1907.10326* (2019) (cit. on pp. 10, 21, 22, 31, 35, 42–44).

[125] Sihaeng Lee, Janghyeon Lee, Byungju Kim, Eojindl Yi, and Junmo Kim. "Patch-wise attention network for monocular depth estimation." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 1873–1881 (cit. on pp. 21, 31, 43).

[126] Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Second. New York, NY, USA: Springer-Verlag, 1998 (cit. on p. 66).

[127] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. "Depth and Surface Normal Estimation From Monocular Images Using Regression on Deep Features and Hierarchical CRFs." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015 (cit. on pp. 10, 35).

[128] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. "Practical stereo matching via cascaded recurrent network with adaptive correlation." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 16263–16272 (cit. on pp. 68, 78).

[129] Jun Li, Reinhard Klein, and Angela Yao. "A two-streamed network for estimating fine-scaled depth maps from single rgb images." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3372–3380 (cit. on p. 35).

[130] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. "RTM3D: Real-Time Monocular 3D Detection from Object Keypoints for Autonomous Driving." In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 644–660 (cit. on pp. 53, 63).

[131] Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. "Lsdir: A large scale dataset for image restoration." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 1775–1787 (cit. on p. vii).

[132] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X. Creighton, Russell H. Taylor, and Mathias Unberath. "Revisiting Stereo Depth Estimation From a Sequence-to-Sequence Perspective With Transformers." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 6197–6206 (cit. on p. 68).

[133] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. "DepthFormer: Exploiting Long-Range Correlation and Local Information for Accurate Monocular Depth Estimation." In: *arXiv preprint arXiv:2203.14211* (2022) (cit. on p. 7).

[134] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. "BinsFormer: Revisiting Adaptive Bins for Monocular Depth Estimation." In: *arXiv preprint arXiv:2204.00987* (2022) (cit. on p. 7).

[135] Julian Lienen, Eyke Hüllermeier, Ralph Ewerth, and Nils Nommensen. "Monocular Depth Estimation via Listwise Ranking Using the Plackett-Luce Model." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 14595–14604 (cit. on pp. 10, 34).

[136] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125 (cit. on p. 62).

[137] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. "Focal Loss for Dense Object Detection." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017 (cit. on p. 58).

[138] Yancong Lin, Silvia L Pintea, and Jan C van Gemert. "Deep Hough-Transform Line Priors." In: *European Conference on Computer Vision*. Springer. 2020, pp. 323–340 (cit. on pp. 5, 52, 55, 57, 59, 60, 63).

[139] Lahav Lipson, Zachary Teed, and Jia Deng. "Raft-stereo: Multilevel recurrent field transforms for stereo matching." In: *2021 International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 218–227 (cit. on pp. 65, 66, 68).

[140] Biyang Liu, Huimin Yu, and Yangqi Long. "Local similarity pattern and cost self-reassembling for deep stereo matching networks." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 2. 2022, pp. 1647–1655 (cit. on pp. 69, 78).

[141] Biyang Liu, Huimin Yu, and Guodong Qi. "Graftnet: Towards domain generalized stereo matching with a broad-spectrum and task-oriented feature." In: *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition. 2022, pp. 13012–13021 (cit. on p. 69).

[142] Ce Liu, Shuhang Gu, Luc Van Gool, and Radu Timofte. "Deep line encoding for monocular 3d object detection and depth prediction." In: *32nd British Machine Vision Conference*. BMVA Press. 2021, p. 354 (cit. on pp. vii, 7, 21, 43, 51).

[143] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. "Single Image Depth Prediction Made Better: A Multivariate Gaussian Take." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17346–17356 (cit. on pp. vii, 2, 7).

[144] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. "VA-DepthNet: A Variational Approach to Single Image Depth Prediction." In: *The Eleventh International Conference on Learning Representations*. 2022 (cit. on pp. vii, 31).

[145] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. "VA-DepthNet: A Variational Approach to Single Image Depth Prediction." In: *The Eleventh International Conference on Learning Representations (ICLR)*. 2023 (cit. on p. 7).

[146] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, Yao Yao, and Luc Van Gool. "Stereo Risk: A Continuous Modeling Approach to Stereo Matching." In: *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024 (cit. on pp. vii, 65).

[147] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. "Learning depth from single monocular images using deep convolutional neural fields." In: *IEEE transactions on pattern analysis and machine intelligence* 38.10 (2015), pp. 2024–2039 (cit. on pp. 10, 35).

[148] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. "An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution." In: *Advances in Neural Information Processing Systems*. 2018 (cit. on pp. 57, 59).

[149] Rui Liu, Chengxi Yang, Wenxiu Sun, Xiaogang Wang, and Hongsheng Li. "StereoGAN: Bridging Synthetic-to-Real Domain Gap by Joint Optimization of Domain Translation and Stereo Matching." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 (cit. on p. 69).

[150] Xingtong Liu, Ayushi Sinha, Masaru Ishii, Gregory D Hager, Austin Reiter, Russell H Taylor, and Mathias Unberath. "Dense depth estimation in monocular endoscopy with self-supervised learning methods." In: *IEEE transactions on medical imaging* 39.5 (2019), pp. 1438–1447 (cit. on pp. 7, 31).

[151] Yuxuan Liu, Yuan Yixuan, and Ming Liu. "Ground-aware monocular 3D object detection for autonomous driving." In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 919–926 (cit. on pp. 58–64).

[152] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. "Swin transformer: Hierarchical vision transformer using shifted windows." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022 (cit. on pp. 18, 23, 25, 39, 42, 45).

[153] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. "A ConvNet for the 2020s." In: *arXiv preprint arXiv:2201.03545* (2022) (cit. on p. 45).

[154] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Wei Li, Christian Theobalt, Ruigang Yang, and Wenping Wang. "Adaptive surface normal constraint for depth estimation." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12849–12858 (cit. on pp. 11, 21, 35, 43).

[155] H Christopher Longuet-Higgins. "A computer algorithm for reconstructing a scene from two projections." In: *Nature* 293.5828 (1981), pp. 133–135 (cit. on p. 31).

[156] I. Loshchilov and F. Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts." In: *International Conference on Learning Representations (ICLR) 2017 Conference Track*. Apr. 2017 (cit. on pp. 59, 63).

[157] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization." In: *International Conference on Learning Representations*. 2019 (cit. on p. 76).

[158] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012 (cit. on p. 34).

[159] Chenchi Luo, Yingmao Li, Kaimo Lin, George Chen, Seok-Jun Lee, Jihwan Choi, Youngjun Francis Yoo, and Michael O Polley. "Wavelet synthesis net for disparity estimation to synthesize dslr calibre bokeh effect on smartphones." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2407–2415 (cit. on p. 65).

[160] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. "Rethinking Pseudo-LiDAR Representation." In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 311–327 (cit. on pp. 53, 63).

[161] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4040–4048 (cit. on pp. 67, 68, 76, 81).

[162] William McCluney. 2014 (cit. on p. 1).

[163] Moritz Menze and Andreas Geiger. "Object scene flow for autonomous vehicles." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3061–3070 (cit. on pp. 68, 76).

[164] Andreas Meuleman, Hakyeong Kim, James Tompkin, and Min H Kim. "FloatingFusion: Depth from ToF and Image-stabilized Stereo Cameras." In: *European Conference on Computer Vision*. Springer. 2022, pp. 602–618 (cit. on p. 65).

[165] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. "Nerf: Representing scenes as neural radiance fields for view synthesis." In: *Communications of the ACM* 65.1 (2021), pp. 99–106 (cit. on p. 31).

[166] Thomas Mollenhoff, Emanuel Laude, Michael Moeller, Jan Lellmann, and Daniel Cremers. "Sublabel-accurate relaxation of nonconvex energies." In: *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition. 2016, pp. 3948–3956 (cit. on p. 32).

[167] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. "3D bounding box estimation using deep learning and geometry." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7074–7082 (cit. on p. 53).

[168] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. "Instant neural graphics primitives with a multiresolution hash encoding." In: *arXiv preprint arXiv:2201.05989* (2022) (cit. on p. 31).

[169] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012 (cit. on p. 8).

[170] Vaishnavh Nagarajan and J Zico Kolter. "Uniform convergence may be unable to explain generalization in deep learning." In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 3).

[171] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation." In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer. 2016, pp. 483–499 (cit. on pp. 68, 75).

[172] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. "Exploring generalization in deep learning." In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 3).

[173] David Nistér. "An efficient solution to the five-point relative pose problem." In: *IEEE transactions on pattern analysis and machine intelligence* 26.6 (2004), pp. 756–770 (cit. on p. 31).

[174] Silvia Noschese, Lionello Pasquini, and Lothar Reichel. "Tridiagonal Toeplitz matrices: properties and novel applications." In: *Numerical linear algebra with applications* 20.2 (2013), pp. 302–326 (cit. on p. 16).

[175] Jiahao Pang, Wenxiu Sun, Chengxi Yang, Jimmy Ren, Ruichao Xiao, Jin Zeng, and Liang Lin. "Zoom and learn: Generalizing deep stereo matching to novel domains." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2070–2079 (cit. on p. 65).

[176] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. "Rethinking Depth Estimation for Multi-View Stereo: A Unified Representation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 (cit. on p. 69).

[177] Patrick Pérez, Michel Gangnet, and Andrew Blake. "Poisson image editing." In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 577–582 (cit. on p. 1).

[178] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. "On the uncertainty of self-supervised monocular depth estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3227–3237 (cit. on p. 8).

[179] Marija Popović, Florian Thomas, Sotiris Papatheodorou, Nils Funk, Teresa Vidal-Calleja, and Stefan Leutenegger. "Volumetric occupancy mapping with probabilistic depth completion for robotic navigation." In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5072–5079 (cit. on p. 1).

[180] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Second. Cambridge, USA: Cambridge University Press, 1992 (cit. on p. 14).

[181] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. "Geonet: Geometric neural network for joint depth and surface normal estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 283–291 (cit. on pp. 11, 21, 32, 35, 43).

[182] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. "Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3997–4008 (cit. on pp. 7, 21, 43).

[183] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors." In: *CVPR 2011*. IEEE. 2011, pp. 777–784 (cit. on p. 1).

[184] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain. "A survey on LiDAR scanning mechanisms." In: *Electronics* 9.5 (2020), p. 741 (cit. on p. 2).

[185] Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. "Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 14648–14657 (cit. on p. 35).

[186] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. "Vision transformers for dense prediction." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12179–12188 (cit. on pp. 7, 8, 21, 22, 24, 43, 44).

[187] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020) (cit. on pp. 23, 44).

[188] Zhibo Rao, Yuchao Dai, Zhelun Shen, and Renjie He. "Rethinking Training Strategy in Stereo Matching." In: *IEEE Transactions on Neural Networks and Learning Systems* (2022), pp. 1–14 (cit. on p. 78).

[189] Carl Edward Rasmussen. "Gaussian processes in machine learning." In: *Summer school on machine learning*. Springer. 2003, pp. 63–71 (cit. on pp. 8, 14).

[190] Stephan Reichelt, Ralf Häussler, Gerald Fütterer, and Norbert Leister. "Depth cues in human visual perception and their realization in 3D displays." In: *Three-Dimensional Imaging, Visualization, and Display 2010 and Display Technologies and Applications for Defense, Security, and Avionics IV*. Vol. 7690. SpIE. 2010, pp. 92–103 (cit. on p. 1).

[191] M. Renardy and R.C. Rogers. *An Introduction to Partial Differential Equations*. Texts in Applied Mathematics. Springer New York, 2004. URL: https://books.google.ch/books?id=0RmDjrF03CUC (cit. on p. 16).

[192] Stephan R Richter, Hassan Abu Al Haija, and Vladlen Koltun. "Enhancing photorealism enhancement." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022) (cit. on p. 7).

[193] Gernot Riegler and Vladlen Koltun. "Stable view synthesis." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12216–12225 (cit. on p. 7).

[194] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. "Dense depth priors for neural radiance fields from sparse input views." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12892–12901 (cit. on p. 7).

[195] O. Ronneberger, P.Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Vol. 9351. LNCS. Springer, 2015, pp. 234–241 (cit. on p. 63).

[196] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors." In: *nature* 323.6088 (1986), pp. 533–536 (cit. on p. 3).

[197] Ashutosh Saxena, Sung Chung, and Andrew Ng. "Learning depth from single monocular images." In: *Advances in neural information processing systems* 18 (2005) (cit. on p. 10).

[198] Ashutosh Saxena, Min Sun, and Andrew Y Ng. "Learning 3d scene structure from a single still image." In: *2007 IEEE 11th international conference on computer vision*. IEEE. 2007, pp. 1–8 (cit. on pp. 10, 31).

[199] Ashutosh Saxena, Min Sun, and Andrew Y Ng. "Make3D: Depth Perception from a Single Still Image." In: *Aaai*. Vol. 3. 2008, pp. 1571–1576 (cit. on pp. 1, 10, 31).

[200] Daniel Scharstein and Richard Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms." In: *International journal of computer vision* 47 (2002), pp. 7–42 (cit. on pp. 3, 5, 65, 67, 68, 76, 81).

[201] Yoav Y Schechner and Nahum Kiryati. "Depth from defocus vs. stereo: How different really are they?" In: *International Journal of Computer Vision* 39 (2000), pp. 141–162 (cit. on p. 2).

[202] Jürgen Schmidhuber. "Deep learning in neural networks: An overview." In: *Neural networks* 61 (2015), pp. 85–117 (cit. on p. 2).

[203] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. "A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on pp. 68, 76).

[204] Zhelun Shen, Yuchao Dai, and Zhibo Rao. "Cfnet: Cascade and fused cost volume for robust stereo matching." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13906–13915 (cit. on pp. 77–79, 82).

[205] Zhelun Shen, Yuchao Dai, Xibin Song, Zhibo Rao, Dingfu Zhou, and Liangjun Zhang. "Pcw-net: Pyramid combination and warping cost volume for stereo matching." In: *European Conference on Computer Vision*. Springer. 2022, pp. 280–297 (cit. on pp. 68, 77–79, 82, 83).

[206] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. "Indoor segmentation and support inference from rgbd images." In: *European conference on computer vision*. Springer. 2012, pp. 746–760 (cit. on pp. 3, 21, 23–25, 27, 33, 34, 42–44, 62).

[207] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015 (cit. on p. 56).

[208] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." In: *arXiv preprint arXiv:1409.1556* (2014) (cit. on p. 31).

[209] Jan Smisek, Michal Jancosek, and Tomas Pajdla. "3D with Kinect." In: *Consumer depth cameras for computer vision: Research topics and applications* (2013), pp. 3–25 (cit. on p. 2).

[210] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. "Sun rgb-d: A rgb-d scene understanding benchmark suite." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 567–576 (cit. on pp. 21–23, 42).

[211] Xiao Song, Guorun Yang, Xinge Zhu, Hui Zhou, Zhe Wang, and Jianping Shi. "AdaStereo: A Simple and Efficient Approach for Adaptive Stereo Matching." In: *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition (CVPR). June 2021, pp. 10328–10337 (cit. on p. 69).

[212] Ashley W Stroupe, Martin C Martin, and Tucker Balch. "Merging gaussian distributions for object localization in multi-robot systems." In: *Experimental Robotics VII*. Springer, 2001, pp. 343–352 (cit. on p. 8).

[213] Murali Subbarao and Gopal Surya. "Depth from defocus: A spatial domain approach." In: *International Journal of Computer Vision* 13.3 (1994), pp. 271–294 (cit. on p. 2).

[214] Guolei Sun, Zhaochong An, Yun Liu, Ce Liu, Christos Sakaridis, Deng-Ping Fan, and Luc Van Gool. "Indiscernible Object Counting in Underwater Scenes." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 13791–13801 (cit. on p. vii).

[215] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022 (cit. on p. 65).

[216] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. "Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14362–14372 (cit. on p. 68).

[217] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. "Cnn-slam: Real-time dense monocular slam with learned depth prediction." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6243–6252 (cit. on p. 7).

[218] Zachary Teed and Jia Deng. "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 16558–16569 (cit. on p. 31).

[219] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. "Unsupervised Adaptation for Deep Stereo." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017 (cit. on p. 69).

[220] Alessio Tonioni, Oscar Rahnama, Thomas Joy, Luigi Di Stefano, Thalaiyasingam Ajanthan, and Philip HS Torr. "Learning to adapt for stereo." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9661–9670 (cit. on p. 69).

[221] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. "Real-Time Self-Adaptive Deep Stereo." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on p. 69).

[222] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. "SMD-Nets: Stereo Mixture Density Networks." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 8942–8952 (cit. on pp. 66, 69).

[223] Shimon Ullman. "The interpretation of structure from motion." In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 203.1153 (1979), pp. 405–426 (cit. on p. 7).

[224] Tom Van Dijk and Guido De Croon. "How Do Neural Networks See Depth in Single Images?" In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 2183–2191 (cit. on p. 3).

[225] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. "A review on the long short-term memory model." In: *Artificial Intelligence Review* 53.8 (2020), pp. 5929–5955 (cit. on p. 2).

[226] Vladimir Vapnik. "Principles of risk minimization for learning theory." In: *Advances in neural information processing systems* 4 (1991) (cit. on pp. 66, 80, 86).

[227] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017 (cit. on p. 68).

[228] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008 (cit. on p. 69).

[229] Jiayi Wang, Raymond KW Wong, and Xiaoke Zhang. "Low-rank covariance function estimation for multidimensional functional data." In: *Journal of the American Statistical Association* 117.538 (2022), pp. 809–822 (cit. on p. 14).

[230] Xiaoyan Wang, Chunping Hou, Liangzhou Pu, and Yonghong Hou. "A depth estimating method from a single image using FoE CRF." In: *Multimedia Tools and Applications* 74.21 (2015), pp. 9491–9506 (cit. on p. 10).

[231] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8445–8453 (cit. on p. 53).

[232] Yan Wang, Zihang Lai, Gao Huang, Brian H Wang, Laurens Van Der Maaten, Mark Campbell, and Kilian Q Weinberger. "Anytime stereo image depth estimation on mobile devices." In: *2019 international conference on robotics and automation (ICRA)*. IEEE. 2019, pp. 5893–5900 (cit. on p. 68).

[233] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. "CroCo v2: Improved Cross-view Completion Pre-training for Stereo Matching and Optical Flow." In: *ICCV*. 2023 (cit. on p. 78).

[234] N.A. Weiss, P.T. Holmes, and M. Hardy. *A Course in Probability*. Pearson Addison Wesley, 2006. URL: https://books.google.ch/books?id=Be9fJwAACAAJ (cit. on p. 18).

[235] Andrew E Welchman, Arne Deubelius, Verena Conrad, Heinrich H Bülthoff, and Zoe Kourtzi. "3D shape perception from combined depth cues in human visual cortex." In: *Nature neuroscience* 8.6 (2005), pp. 820–827 (cit. on p. 1).

[236] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. "Synsin: End-to-end view synthesis from a single image." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7467–7477 (cit. on p. 1).

[237] Robert Woodham. "Photometric Method for Determining Surface Orientation from Multiple Images." In: *Optical Engineering* 19 (Jan. 1992) (cit. on p. 2).

[238] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. "Unsupervised Learning of Probably Symmetric Deformable 3D Objects From Images in the Wild." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 (cit. on p. 1).

[239] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. "Structure-guided ranking loss for single image depth prediction." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 611–620 (cit. on pp. 10, 34).

[240] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. "Multi-Scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 10).

[241] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5354–5362 (cit. on p. 53).

[242] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. "Structured attention guided convolutional neural fields for monocular depth estimation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3917–3925 (cit. on p. 10).

[243] Gangwei Xu, Junda Cheng, Peng Guo, and Xin Yang. "Attention concatenation volume for accurate and efficient stereo matching." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12981–12990 (cit. on pp. 68, 77–79, 82, 83).

[244] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. "Iterative Geometry Encoding Volume for Stereo Matching." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and*

Pattern Recognition. 2023, pp. 21919–21928 (cit. on pp. 66–68, 77–79, 81, 82).

[245] Haofei Xu and Juyong Zhang. "AANet: Adaptive Aggregation Network for Efficient Stereo Matching." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1959–1968 (cit. on p. 68).

[246] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. "Efficient joint segmentation, occlusion labeling, stereo and flow estimation." In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 756–771 (cit. on p. 65).

[247] Kohei Yamashita, Shohei Nobuhara, and Ko Nishino. "3D-GMNet: Single-View 3D Shape Recovery as A Gaussian Mixture." In: *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020 (cit. on p. 8).

[248] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. "Transformer-based attention networks for continuous pixel-wise prediction." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16269–16279 (cit. on pp. 11, 21, 22, 43, 44).

[249] Jiayu Yang, Jose M Alvarez, and Miaomiao Liu. "Non-parametric depth distribution modelling based depth inference for multi-view stereo." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8626–8634 (cit. on p. 69).

[250] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 1281–1292 (cit. on p. 31).

[251] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. "MVS-Net: Depth Inference for Unstructured Multi-view Stereo." In: *European Conference on Computer Vision (ECCV)* (2018) (cit. on p. 3).

[252] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. "Recurrent MVSNet for High-resolution Multi-view Stereo Depth Inference." In: *Computer Vision and Pattern Recognition (CVPR)* (2019) (cit. on p. 3).

[253] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. "Enforcing geometric constraints of virtual normal for depth prediction." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5684–5693 (cit. on pp. 11, 21, 22, 25, 26, 31, 32, 35, 43, 44).

[254] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. "Enforcing geometric constraints of virtual normal for depth prediction." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5684–5693 (cit. on pp. 53, 64).

[255] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. "Deep layer aggregation." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2403–2412 (cit. on p. 62).

[256] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. "Neural Window Fully-Connected CRFs for Monocular Depth Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 3916–3925 (cit. on pp. 7, 8, 10, 11, 21, 22, 24, 26, 27, 31, 33, 35, 43–45, 47, 48).

[257] Jure Zbontar and Yann LeCun. "Computing the Stereo Matching Cost With a Convolutional Neural Network." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015 (cit. on pp. 3, 68).

[258] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning (still) requires rethinking generalization." In: *Communications of the ACM* 64.3 (2021), pp. 107–115 (cit. on p. 3).

[259] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. "GA-Net: Guided Aggregation Net for End-to-end Stereo Matching." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 185–194 (cit. on pp. 3, 65, 66, 68).

[260] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. "Domain-invariant stereo matching networks." In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer. 2020, pp. 420–439 (cit. on p. 69).

[261] Jiawei Zhang, Xiang Wang, Xiao Bai, Chen Wang, Lei Huang, Yimin Chen, Lin Gu, Jun Zhou, Tatsuya Harada, and Edwin R Hancock. "Revisiting domain generalized stereo matching networks from a feature consistency perspective." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13001–13011 (cit. on p. 69).

[262] Ruo Zhang, Ping-Sing Tsai, J.E. Cryer, and M. Shah. "Shape-from-shading: a survey." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.8 (1999), pp. 690–706 (cit. on p. 51).

[263] Zhengyou Zhang. "Microsoft kinect sensor and its effect." In: *IEEE multimedia* 19.2 (2012), pp. 4–10 (cit. on pp. 1, 2).

[264] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. "Pattern-affinitive propagation across depth, surface normal and semantic segmentation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4106–4115 (cit. on p. 53).

[265] Haoliang Zhao, Huizhou Zhou, Yongjun Zhang, Jie Chen, Yitong Yang, and Yong Zhao. "High-Frequency Stereo Matching Network." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 1327–1336 (cit. on pp. 66–68, 77–79, 81, 82).

[266] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. "Pyramid scene parsing network." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2881–2890 (cit. on pp. 68, 75).

[267] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Places: A 10 million image database for scene recognition." In: *IEEE transactions on pattern analysis and machine intelligence* 40.6 (2017), pp. 1452–1464 (cit. on p. 2).

[268] Rui Zhou, Jiaxi Ying, and Daniel P Palomar. "Covariance Matrix Estimation Under Low-Rank Factor Model With Nonnegative Correlations." In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 4020–4030 (cit. on p. 14).

[269] Xiaosu Zhu, Jingkuan Song, Lianli Gao, Feng Zheng, and Heng Tao Shen. "Unified Multivariate Gaussian Mixture for Efficient Neural Image Compression." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 17612–17621 (cit. on p. 8).

[270] Daniel Zoran, Phillip Isola, Dilip Krishnan, and William T Freeman. "Learning ordinal relationships for mid-level vision." In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 388–396 (cit. on pp. 10, 34).