Summer 2022

Learning Depth From Images

Zhenyao Wu

Learning depth from images

by

Zhenyao Wu

Bachelor of Engineering
Tianjin University, 2018

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2022

Accepted by:

Song Wang, Major Professor

Lili Ju, Major Professor

Michael N. Huhns, Committee Member

Yan Tong, Committee Member

Lannan Luo, Committee Member

Tracey L. Weldon, Vice Provost and Dean of the Graduate School

Acknowledgments

I would like to express my sincere gratitude and appreciation to my supervisors, Prof. Song Wang and Prof. Lili Ju, for their guidance, patience, and encouragement during my graduate studies. I am so lucky to have two advisors for pursuing my Ph.D. degree. Their immense knowledge, great patience, and plentiful experience have encouraged me in my academic research and daily life. Without their mentoring and guidance, this dissertation would not be possible.

I would also like to express my sincere appreciation to my dissertation committee members, Prof. Michael N. Huhns, Prof. Yan Tong, and Prof. Lannan Luo for their invaluable suggestions and support of my work.

I want to thank my colleagues including Dr. Dazhou Guo, Dr. Kang Zheng, Dr. Yang Mi, Dr. Hao Guo, Jun Zhou, Yuhang Lu, Xinyi Wu, Lan Fu, Liang Zhao, Yuankai Teng, Rabab Abdelfattah, Canyu Zhang, Xiaoguang Li, Ziyu Zhao, Pingping Cai and other members in our research group for their technical support and encouragement. I would also like to acknowledge Dr. Zhihao Liu, Dr. Yong Zhao, Jin Wan, and Mohammad Hassan Erfani. It was a great time to collaborate with you.

Lastly, I am deeply grateful to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation. I would like to give my warmest thanks to Xinyi for holding my hands and making me the best of me during the past years.

Abstract

Estimating depth from images has become a very popular task in computer vision which aims to restore the 3D scene from 2D images and identify important geometric knowledge of the scene. Its performance has been significantly improved by convolutional neural networks in recent years, which surpass the traditional methods by a large margin. However, the natural scenes are usually complicated, and hard to build the correspondence between pixels across frames, such as the region containing moving objects, illumination changes, occlusions, and reflections. This research explores rich and comprehensive spatial correspondence across images and designs three new network architectures for depth estimation whose inputs can be a single image, stereo pairs, or monocular video.

First, we propose a novel semantic stereo network named SSPCV-Net, which includes newly designed pyramid cost volumes for describing semantic and spatial correspondence on multiple levels. The semantic features are inferred from a semantic segmentation subnetwork while the spatial features are constructed by hierarchical spatial pooling. In the end, we design a 3D multi-cost aggregation module to integrate the extracted multilevel correspondence and perform regression for accurate disparity maps. We conduct comprehensive experiments and comparisons with some recent stereo matching networks on Scene Flow, KITTI 2015 and 2012, and Cityscapes benchmark datasets, and the results show that the proposed SSPCV-Net significantly promotes the state-of-the-art stereo-matching performance.

Second, we present a novel SC-GAN network with end-to-end adversarial training for depth estimation from monocular videos without estimating the camera pose and

pose change over time. To exploit cross-frame relations, SC-GAN includes a spatial correspondence module that uses Smolyak sparse grids to efficiently match the features across adjacent frames and an attention mechanism to learn the importance of features in different directions. Furthermore, the generator in SC-GAN learns to estimate depth from the input frames, while the discriminator learns to distinguish between the ground-truth and estimated depth map for the reference frame. Experiments on the KITTI and Cityscapes datasets show that the proposed SC-GAN can achieve much more accurate depth maps than many existing state-of-the-art methods on monocular videos.

Finally, we propose a new method for single image depth estimation which utilize the spatial correspondence from stereo matching. To achieve the goal, we incorporate a pre-trained stereo network as a teacher to provide depth cues for the features and output generated by the student network which is a monocular depth estimation network. To further leverage the depth cues, we developed a new depth-aware convolution operation that can adaptively choose subsets of relevant features for convolutions at each location. Specifically, we compute hierarchical depth features as the guidance, and then estimate the depth map using such depth-aware convolution which can leverage the guidance to adapt the filters. Experimental results on the KITTI online benchmark and Eigen split datasets show that the proposed method achieves the state-of-the-art performance for single-image depth estimation.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Scene depth plays an important role in computer vision, which is a basic pattern to understand geometric information within a scene. It can provide the 3D relationship between the objects and their environment, thus it has a wide range of applications such as robotic navigation [7], 3D reconstruction [154], autonomous driving [15], and virtual reality [18]. Taking robotic navigation and autonomous driving as an example, scene depth can help the robotics and cars with object detection, collision avoidance, and high-solution map creation.

The activate approaches to obtain the depth in the real world are utilizing the LiDAR and structured light as shown in Figure 1.1. LiDAR is the abbreviation of light detection and ranging. Generally, the LiDAR sensor emits light waves into the scene and measures the time for the reflected light to calculate the distance it traveled. Differently, The structured light scanner projects some light patterns on the target object and captures the light using a camera, and uses the information to recover the 3D geometry.

Figure 1.1 The devices for obtaining 3D depth: a LiDAR sensor (left) and a structured light 3d scanner (right).

However, LiDAR is unable to work in bad weather conditions, such as heavy rain, snow, and fog. The structured light can only work well on a very narrow depth range (0.1m to 2m), and the environment light and moving objects have a negative

impact on accurate depth capturing. Most importantly, these devices are usually very expensive, and obtaining the dense depth map from them also requires numerous computation resources and human power. Compared with the LiDAR sensors and structured light scanners, the cameras are usually much cheaper and more friendly to use. Therefore, estimating depth from images has become one of the mainstream computer vision research. Figure 1.2 shows some samples of depth map corresponding to the RGB image.

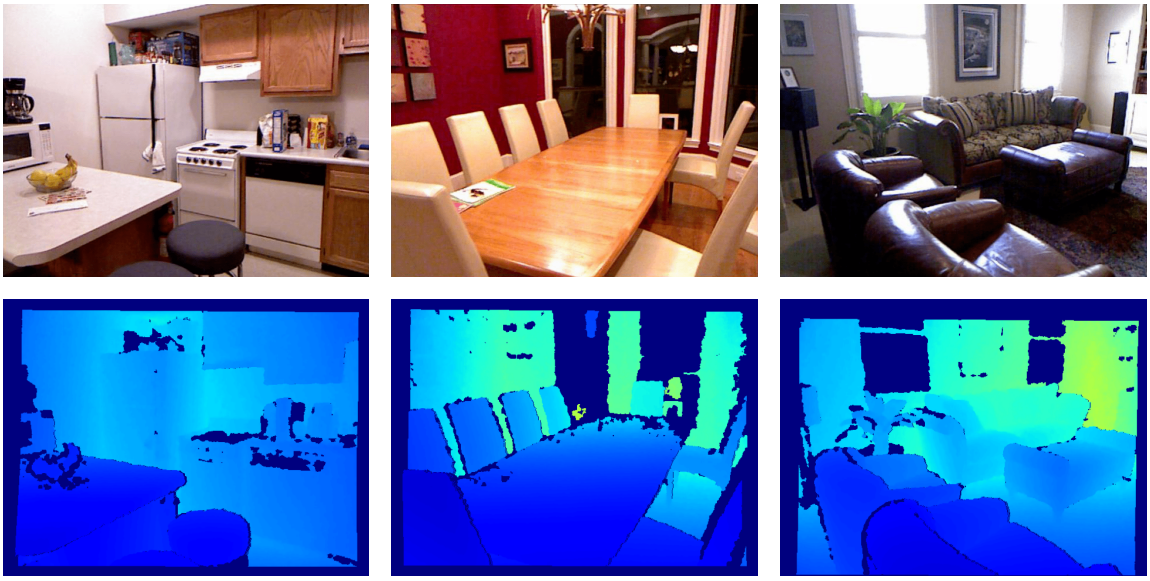

Figure 1.2 The images with the corresponding depth maps from NYU Depth Dataset V2 [115].

In the early period, the depth estimation from the images depends on the depth cues, such as vanishing points [132, 3], and focus-defocus [102]. However, those methods can only be applied in constraint scenes and work for some specific cameras. With the development of computer vision, many hand-made features have been proposed, such as scale-invariant feature transform (SIFT) [69], Conditional Random Field (CRF) [84], and Markov Random Field (MRF) [126], which were adopted to predict depth maps with the machine learning process. However, the results are not

satisfactory.

In the 2010s, Convolutional neural networks [116, 51] have been very successful in many computer vision tasks, *e.g.*, image classification, semantic segmentation, action recognition, and object tracking. For depth estimation tasks, most of the works leverage the large dataset [37, 115, 26] and propose data-driven approaches, such as [33, 151, 71, 14]. Specifically, the depth estimation networks take the RGB images as input and output depth maps. During training, the objective functions are used to penalize the errors between the output depth and the ground truth. Compared with the traditional methods, the deep learning approaches optimized by using numerous data can learn sufficient features and perform better on real-world images. From the perspective of the input types, we usually group the depth estimation as the single image depth estimation, stereo images depth estimation (*i.e.*, stereo matching), and monocular video depth estimation.

Similar to judging the size and distance of any object like a human, we can estimate depth from a single image and use it to reflect the three-dimensional world. The single image depth estimation takes only one RGB image as the input and produces the depth value for all pixels of the input image. The traditional methods [111, 57, 69] estimate single image depth from the predefined depth cues such as line angles and perspective, atmospheric effect, shading, occlusion, object size, real-world environment, and view angle. Recently, the CNN-based approaches [32, 139, 33] build the regression model to calculate the per-pixel depth map after supervised training.

Learning depth from stereo images simulates the way of human eyes by two cameras, and it is also known as stereo matching. The key point of stereo matching is finding corresponding pixels on the two rectified images along the horizontal line. In the stereo system, the two cameras with the same focal length f are parallel. Given the distance between the two cameras b, and the shift (disparity) for corresponding points between the images $(x1-x2)$. The depth value d of this point can be calculated

by:

$$d = \frac{fb}{x2 - x1}, \tag{1.1}$$

as shown in the Figure 1.3.

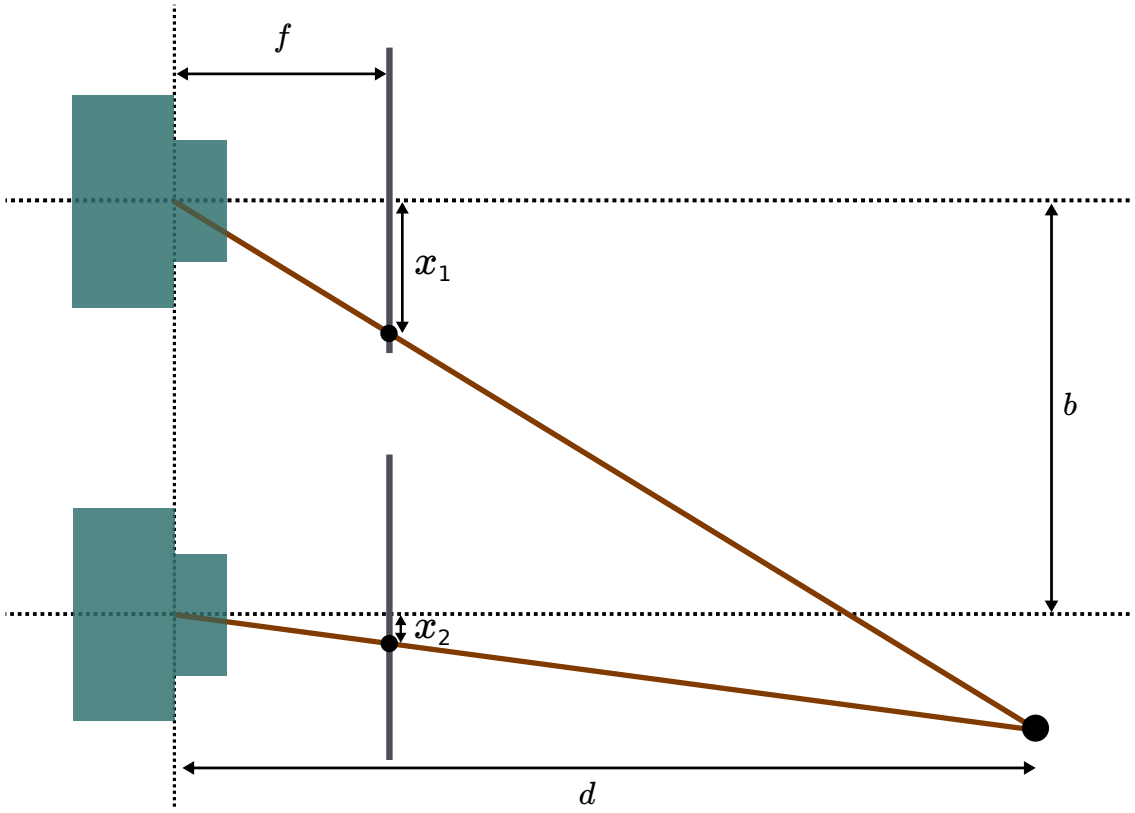

Figure 1.3 An illustration of the depth d calculation from stereo images. b is the distance between the two cameras, and f is the focal length. $x_2 - x_1$ denotes the disparity.

Note that the disparity in the human visual system is the angle between two lines of projection. However, in the computer vision area, the disparity is always usually measured in pixels and represents the coordinate differences of the point between the right and left images instead of a visual angle.

Traditional stereo matching approaches consist of matching cost computation, cost aggregation, disparity optimization, and post-processing. The most important step is the matching cost computation which usually uses the low-level hand-crafted

feature to measure the similarity between the patches from the left image and right image [10, 107]. Recently, as in many other computer vision tasks, convolutional neural networks have been applied to stereo matching with significant success. Most of them [71, 14] formed a 4D cost volume with the concatenated deep features from the image pairs along the horizontal line. After matching cost computation, they use the 3D convolution networks for cost aggregation and estimate the final depth map.

Compared with stereo matching approaches which require at least two fixed cameras, monocular video depth estimation uses only one camera. And it obtains depth from the captured video sequence instead of the paired left-right image. The idea of monocular video depth estimation comes from that humans can know the three-dimensional structure of an object by moving around it, which is known as structure from motion (SfM) as shown in Figure 1.4. So monocular video depth estimation is easy to apply to most the scenarios, while the accuracy relies heavily on high-quality image sequences.

Learning depth from multiple viewpoints presents a similar problem as stereo matching, while the correspondence of the former is more complicated due to the movement of the camera. To find correspondence between images from the different viewpoints, features such as SIFT [69] and SURF [5] can be used for pixel matching. Recently, most existing approaches[122, 161] use high-quality ground truth depth data to train a neural network for precise depth estimation for each image in the video.

1.2 Challenges

A variety of advanced deep neural networks have been developed for depth estimation from different source using different network architectures [14, 73, 45], different objective function[145, 136] and different physics constraint[148, 91]. However, it remains a very challenging task since images may have repeated patterns, object occlusions,

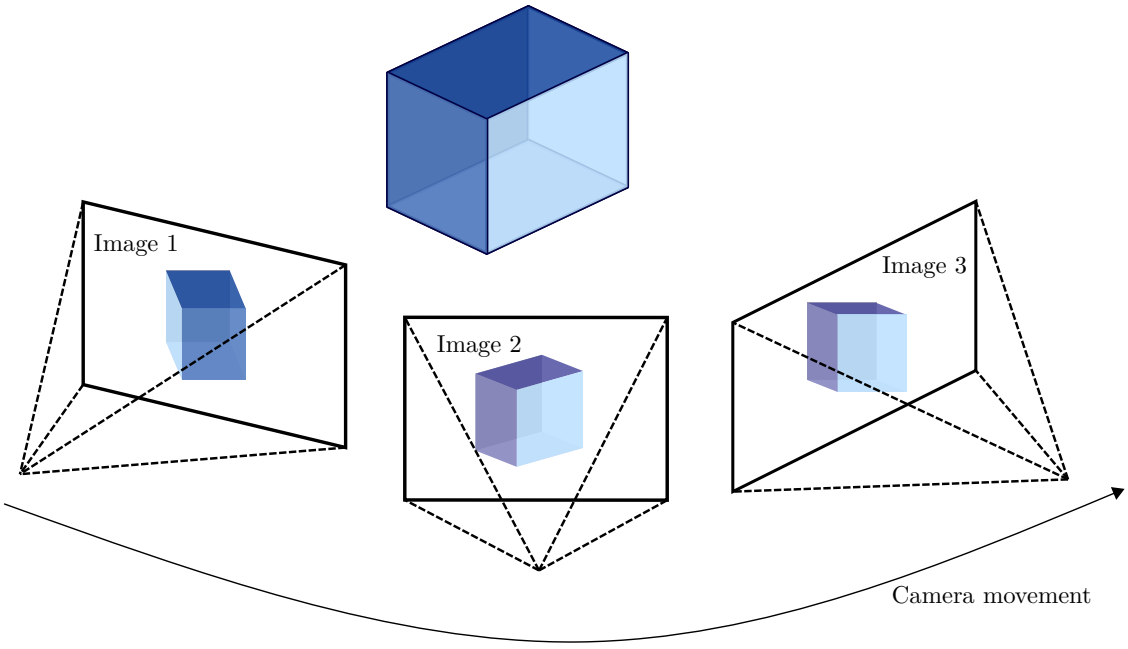

Figure 1.4 An illustration of the structure from motions. With the movement of the human eye or a camera, we can estimate the 3D structure of a scene from a set of 2-D viewpoints.

and textureless regions, which makes the deep learning approaches hard to train. Moreover, the accuracy of depth estimation is extremely important. For instance, when the models are applied in the system of autonomous driving, safety is always the top consideration for this community.

For the stereo matching task, the current approaches always construct the cost volume using the deep features extracted by the respective networks [71, 97, 118, 14]. For these prior works, the cost volume is constructed at a single level without considering multiscale spatial information separately underlying the stereo image pairs. For cost aggregation, a single-scale cost volume may not be sufficient to capture the spatial relationship between stereo images. Since the two images are captured by two cameras from different viewpoints, they will have the occlusion regions, *i.e.*, the objects in the first image may not appear in the second image (as shown in the yellow

box of the Figure 1.5), which makes the matching processing harder. Besides, for the reflective surfaces (as shown in the red box of the Figure 1.5), the pixels from two images might have different appearances, so it is difficult to find accurate corresponding points when applying the intensity-consistency matching.

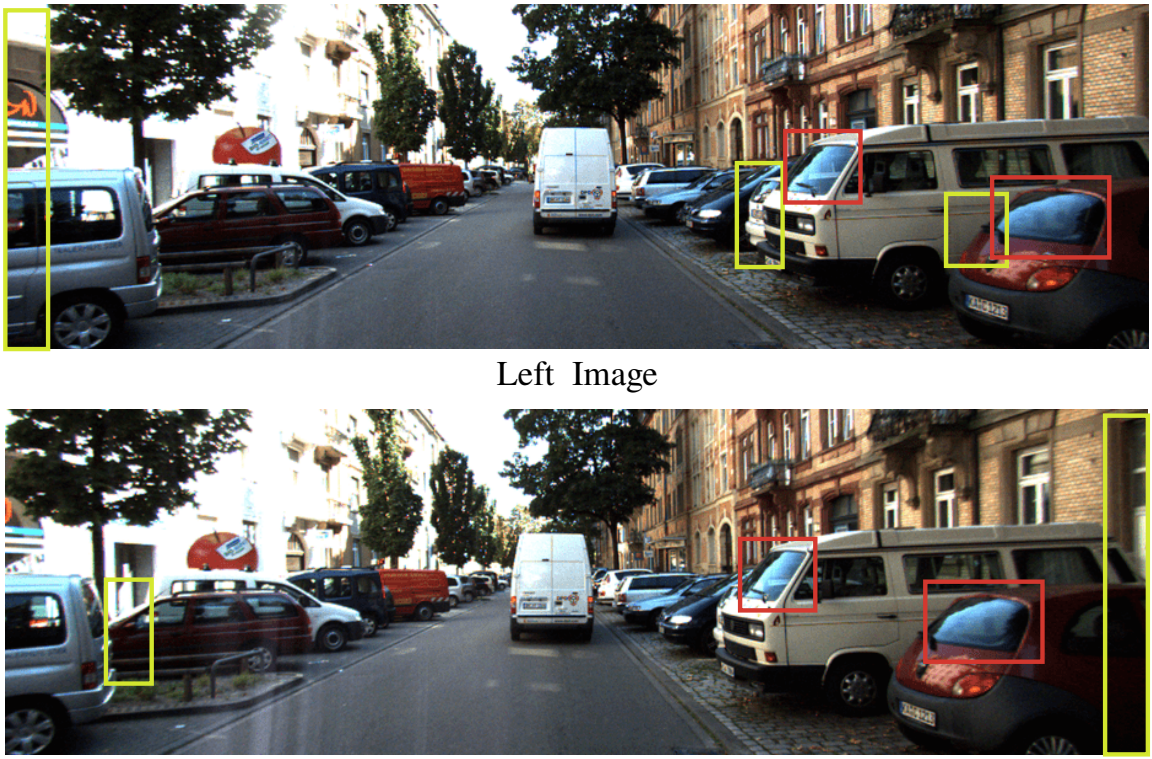

Figure 1.5 An illustration of the occlusion and reflection in the stereo vision. The regions in yellow boxes are occluded in another image, and the regions in red boxes have a reflective surface which has different appearances on the left and right images.

Different from the stereo matching task where the input pair of stereo images are taken by two cameras with a fixed relative pose, the camera pose change between adjacent frames in videos is time-varying, which makes depth estimation from monocular videos a very challenging problem. Most of the available methods address this problem by first estimating the camera pose and pose change over time, usually by

training respective CNNs [108, 130, 162]. For these methods, errors in camera-pose estimation can significantly affect the accuracy of final depth estimation [131].

The most difficult setting is the single image depth estimation since it is an ill-posed problem. As shown in Figure 1.6, the infinite number of 3D scenes can project to the same 2D plane, so the single-image depth estimation still shows a very large performance gap from the depth estimation using a pair of stereo images and video. This is not strange because the former lacks the crucial multi-view geometric information, even if the deep learning techniques can help infer geometric information with data-driven approaches [95, 93].

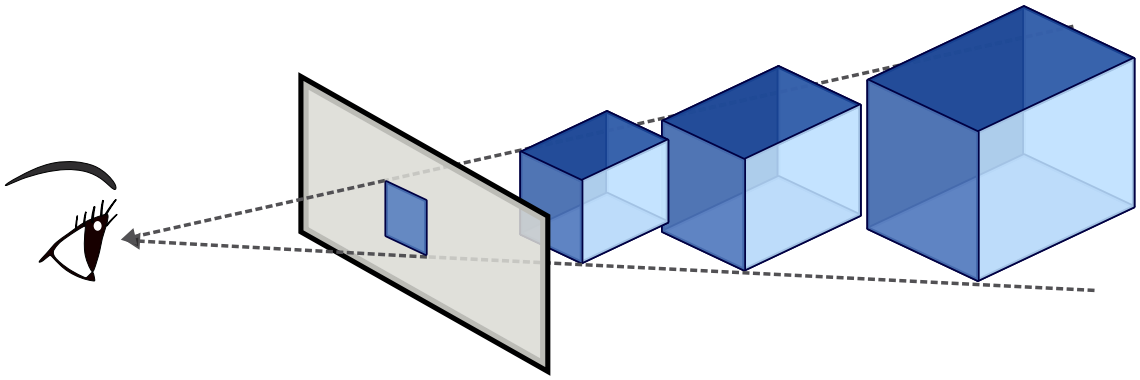

Figure 1.6 An illustration of the ill-pose problem for single-image depth estimation. The infinite number of 3D scenes can project to the same 2D plane.

1.3 Scope of the Proposed Research

To address these three depth estimation tasks (depth from stereo images, video, and single image), this dissertation explores and studies the spatial correspondence between the images for a deep network. Specifically, for the stereo matching task, the spatial correspondence is built between the left and right image, the cross-frame spatial correspondence is constructed for monocular depth estimation, and correspondence knowledge transferring is studied for single image depth estimation.

1.3.1 SPATIAL CORRESPONDENCE FOR THE STEREO MATCHING

To accurately find pixel-level correspondence of the stereo pair, this dissertation studies the pyramid cost volumes for capturing semantic and multiscale spatial information simultaneously.

The use of semantic information aims to capture context cues in a simple manner and learn the similarity of objects' pixels from the left and right semantic segmentation features. As shown in Figure 1.7, the semantic segmentation captures different objects and their boundaries in images and shows much spatial and intensity correlation with the disparity map. In particular, an accurate semantic segmentation can help rectify the disparity values along the object boundaries, which are usually more prone to error in stereo matching

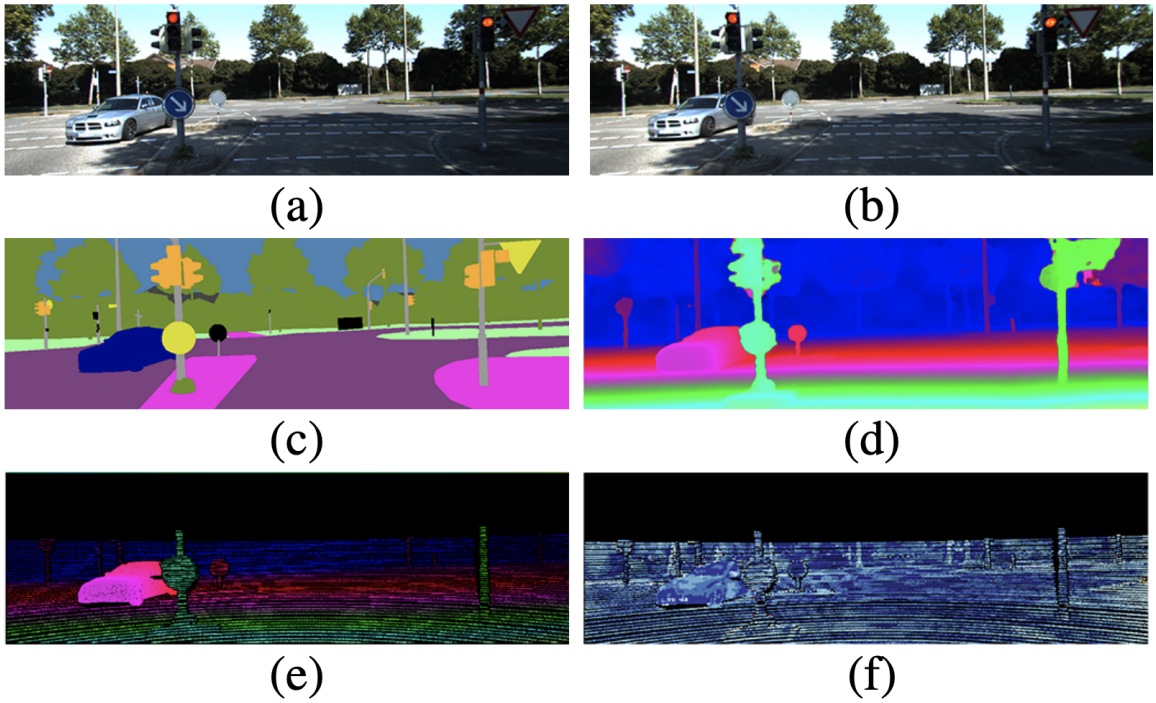

Figure 1.7 (a)&(b) the input stereo pairs (left and right images) from the KITTI dataset; (c) the semantic segmentation; (d) the predicted disparity; (e) the ground-truth of the disparity estimation; (f) the error map of the prediction.

Different from the previous work PSMNet[14], where only a single cost volume is generated from the deep features, this dissertation instead uses multilevel spatial features to build spatial pyramid cost volumes. Besides, we achieve the cost aggregation in a hierarchical way, which enables it to learn both coarse and fine-grained spatial correspondence for stereo matching.

1.3.2 SPATIAL CORRESPONDENCE ACROSS THE FRAMES

This dissertation exploits latent information – spatial correspondence between adjacent frames of a monocular video and estimates the depth by supervised training as shown in Figure 1.8.

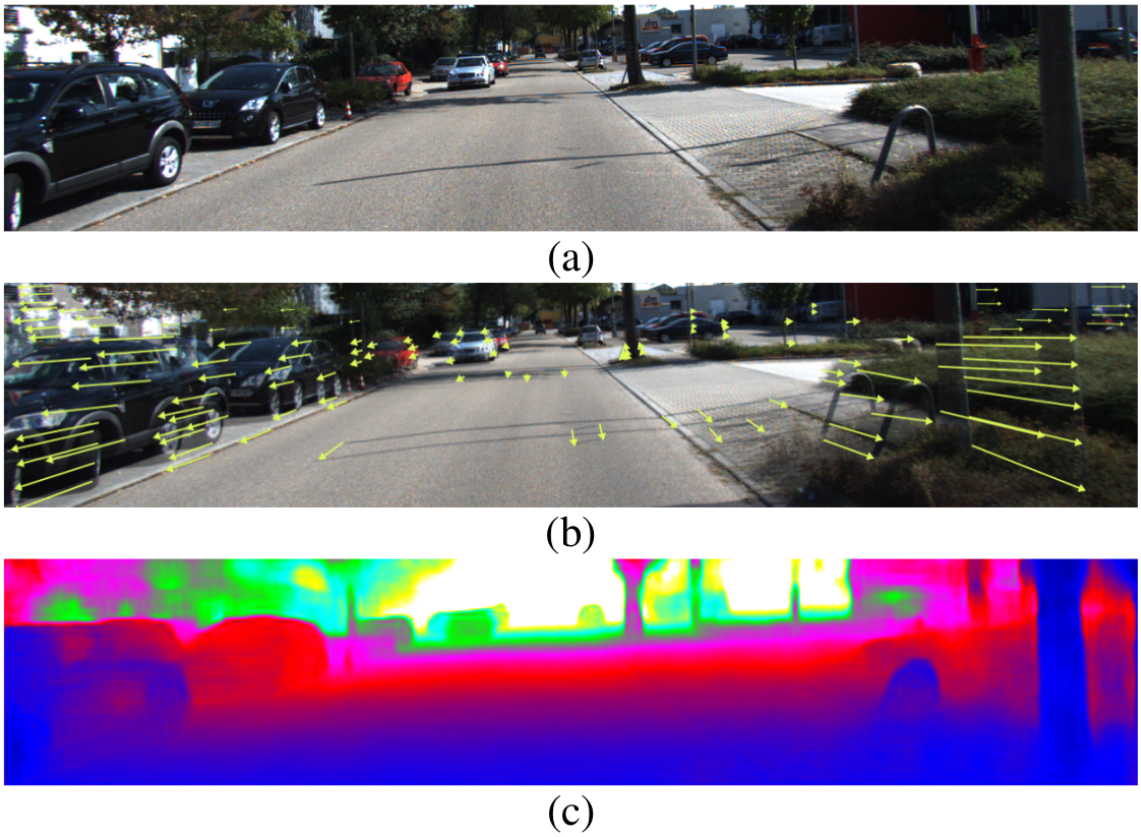

Figure 1.8 An illustration of the spatial relations between adjacent frames and depth estimation. (a) The reference frame; (b) the corresponding features between two adjacent frames; (c) the estimated depth map.

One key issue in building the feature relations between two frames lies in the computational and memory complexity. With large camera-pose change between frames and high image resolution, both of which are common in autonomous driving and virtual reality, the search space of corresponding features between two frames is very large. To address this issue, this dissertation considers down-sample patches of interest in adjacent frames using the Smolyak sparse grid method [117], which brings us both efficiency and accuracy in building cross-frame spatial relations.

1.3.3 TRANSFERRING THE SPATIAL CORRESPONDENCE KNOWLEDGE FOR SINGLE IMAGE DEPTH ESTIMATION

For the single image depth estimation, this dissertation makes use of the feature extracted from the stereo pair to rectify the ill-posed features extracted from a single image by using the knowledge-distillation technique [52], which was initially proposed for model compression (Figure 1.9).

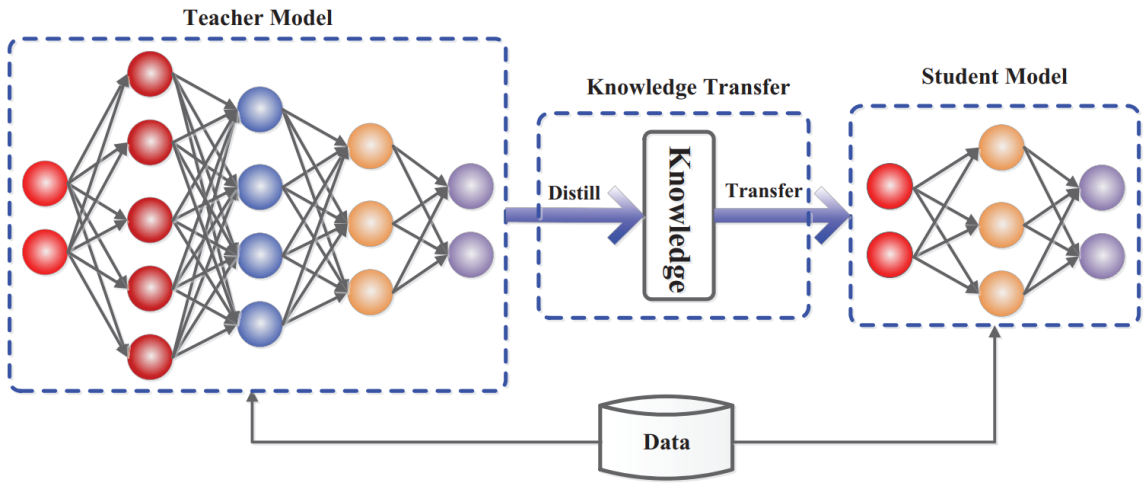

Figure 1.9 An illustration of knowledge-distillation for model compression [44].

Moreover, all existing methods treat the features in different depths equally using traditional convolution operations and these convolution operations may mix the

features from different objects, which might cause inaccurate prediction of depth and abrupt depth change near the border of two adjacent objects in the image. Inspired by the work of [50] which proposes a segmentation-aware CNN by adapting its filters at each pixel based on segmentation cues, we focus on designing a novel depth-aware convolution operation for single-view depth estimation based on depth cues during the knowledge transferring.

1.4 Proposed Approaches

This dissertation proposes three approaches to estimating depth from the images. The first work takes a stereo pair as input and explores the spatial correspondence between left and right images. The second work focuses on estimating the depth from the monocular video using the cross-frame spatial correspondence. The last work studied correspondence knowledge transfer for single image depth estimation.

1.4.1 Semantic Stereo Matching with Pyramid Cost Volumes

In the first work, we propose a new semantic stereo network of SSPCV-Net. One of our major ideas in this work is to develop a new CNN network with multilevel cost volumes, which we call pyramid cost volumes, for better capturing the disparity details in stereo matching. Our work is also partly inspired by the recent work of SegStereo [141] that integrates semantic information to stereo matching through joint learning. The semantic segmentation captures different objects and their boundaries in images and shows much spatial and intensity correlation with the disparity map. In particular, an accurate semantic segmentation can help rectify the disparity values along the object boundaries, which are usually more prone to error in stereo matching [8, 46]. Thus, our network will also integrate both the semantic and the spatial information in multiple levels for constructing pyramid cost volumes, and we find that such an approach can improve the stereo-matching accuracy significantly.

Besides, we propose a 3D multi-cost aggregation module in SSPCV-Net to integrate the extracted multilevel features and perform regression for accurate disparity-map prediction. From the comprehensive experiments and comparisons with some recent stereo matching networks, SSPCV-Net significantly promotes the state-of-the-art performance of stereo matching on the benchmark datasets of Scene Flow, KITTI 2015 and 2012, and Cityscapes.

1.4.2 Spatial Correspondence with Generative Adversarial Network: Learning Depth from Monocular Videos

In the second work, we present a novel SC-GAN network with end-to-end training for depth estimation from monocular videos without estimating the camera pose and pose change over time. To exploit cross-frame relations, SC-GAN includes a newly designed spatial correspondence module and an attention mechanism to learn the importance of features in different directions. We make use of Smolyak sparse grids to greatly reduce the complexity of correlation calculations for the spatial correspondence of adjacent frames. As far as we know, this is the first time to use this method for solving computer vision problems.

Furthermore, we use the generator in SC-GAN learns to estimate depth from the input frames, while a discriminator is applied to distinguish between the ground-truth and estimated depth map for the reference frame. The proposed SC-GAN significantly promotes the state-of-the-art performance of the monocular depth estimation on the KITTI and Cityscapes datasets.

1.4.3 Learning Depth from Single Image using Depth-Aware Convolution and Stereo Knowledge

In the last work, we introduce a pre-trained stereo network to provide additional supervision on both intermediate features and the output of the student through

knowledge distillation. We also design a novel depth-aware convolution operation in DACNN to learn the depth with the help of spatial correspondence. Specifically, the depth-aware convolution operation can adaptively choose subsets of relevant features for convolutions at each location, and we compute hierarchical depth features as the guidance, and then estimate the depth map using such depth-aware convolution which can leverage the guidance to adapt the filters. Experimental results on the KITTI online benchmark and Eigen split datasets show that the proposed method achieves the state-of-the-art performance for single-image depth estimation.

1.5 Structure of the Dissertation

The remainder of this dissertation is organized as follows. In Chapter 2, we overview the relevant knowledge used in this research. In Chapter 3, a literature review for related works is conducted. Chapter 4 elaborates on the proposed method of using semantic information for stereo matching. Chapter 5 explores spatial correspondence for depth estimation from monocular videos. Chapter 6 elaborates a new method that can estimate depth from a single image with knowledge distillation. Finally, Chapter 7 concludes the dissertation.

Chapter 2

Background

This chapter provides some background to this dissertation, including a brief introduction to CNN networks, the spatial correspondence in optical flow estimation, some advanced convolutional layers, and the benchmarks for depth estimation tasks. Specifically, the CNN networks are the fundamental knowledge used in all three proposed methods, the optical flow estimation is associated with the second work – depth estimation from monocular video, and the introduction of the advanced convolutional layers is related to the third work – single image depth estimation.

2.1 A brief introduction of CNN networks

With the advances in computation resources and massive datasets, CNNs are the most successful architectures in the deep learning community, especially for data-driven computer vision tasks. CNNs mainly consist of convolutional layers, normalization layers, and activation layers.

In the convolutional layer, the learnable local kernel is involved to extract features from images. Taking 3×3 convolutional layer with dilation rate 1 as an example, the receptive field $\mathcal{R}$ can be defined as

$$\mathcal{R} = \{(-1,-1), (-1,0), \ldots, (0,1), (1,1)\}. \tag{2.1}$$

For each location $\mathbf{p}_0$ on the output feature map $\mathbf{y}$, we have

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n), \tag{2.2}$$

where $\mathbf{p}_n$ enumerates the locations in $\mathcal{R}$. The normalization layers stabilize the data which can improve the learning speed and avoid overfitting. The activation layers enable the non-linear functions modeling in CNNs.

Based on those layers, many well-known CNN-based archtecture are proposed in last ten years, *e.g.*AlexNet [75], VGGNet [116], GoogLeNet [121], U-Net [110], and ResNet [51], and the network going deeper and deeper. AlexNet [75] only has five

convolution layers as shown in Figure 2.1, while VGGNet[116] and GoogLeNet [121] have 19 and 22 layers, respectively.

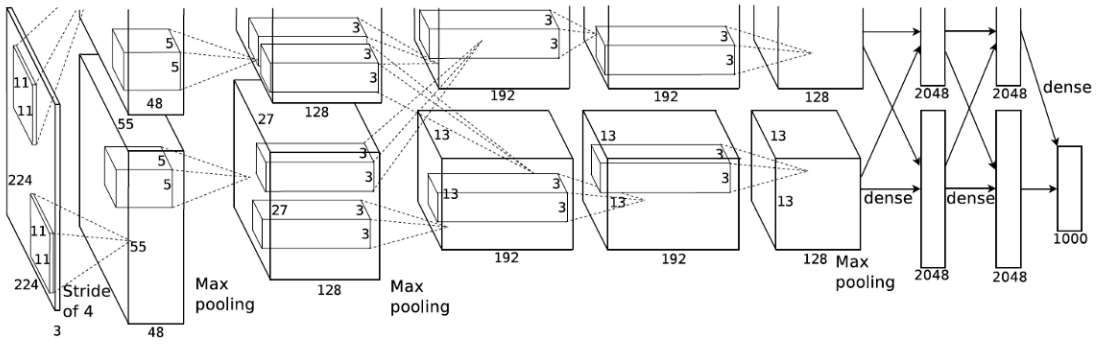

Figure 2.1 The architecture of Alexnet. From [75]. It consists of five convolutional layers, three max-pooling layers, and three fully connected layers.

However, training deep networks will cause the vanishing gradient problem. To solve this issue, He *et al.*proposes ResNet [51] in 2016. The residual block can be defined as:

$$y = \mathcal{F}(x, \{W_i\}) + x, \tag{2.3}$$

where x and y are the input and output, and the function $\mathcal{F}(x, \{W_i\})$ represents the residual mapping with the learnable parameters W_i. With this skip connection operation as shown in Figure 2.2, ResNet makes it possible to train up to hundreds of layers and still achieves compelling performance.

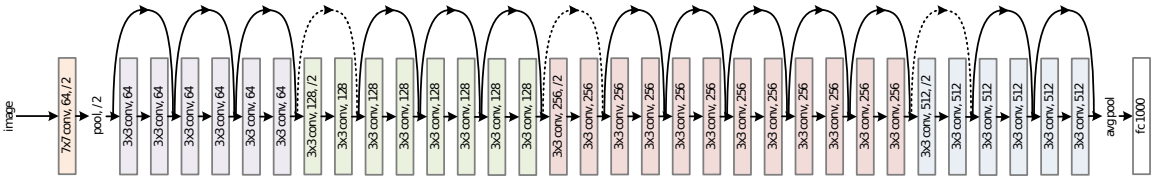

Figure 2.2 The architecture of ResNet. From [51]. It utilize skip connections (shortcuts) to jump the feature over the layers.

The detailed structure of ResNet is shown in Table 2.1. With the remarkable results, ResNet become the most popular backbone to extract deep features in lots of computer vision works *e.g.*, image classification, object detection, and human re-identification.

Table 2.1 Architectures for ResNet [51] with different numbers of blocks stacked.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

For the image segmentation task, Ronneberger *et al.*propose a new network structure U-Net [110] as illustrated in Figure 2.3. The U-Net consists of a contracting path and an expansive path, and it has 23 convolutional layers in total. In the contracting path, the image is down-sampled with convolution layers and max-pooling layers. While in the expansive path, the features go through the up-convolution layer for upsampling. The U-shape network architecture not only extracts features after the convolution layer but also restores the predictions to the same size as the original input images. So it is widely used in many dense prediction tasks (*e.g.*, semantic segmentation, optical flow estimation, depth estimation) and low-level vision tasks (*e.g.*, low-light enhancement, shadow removal, image denoising).

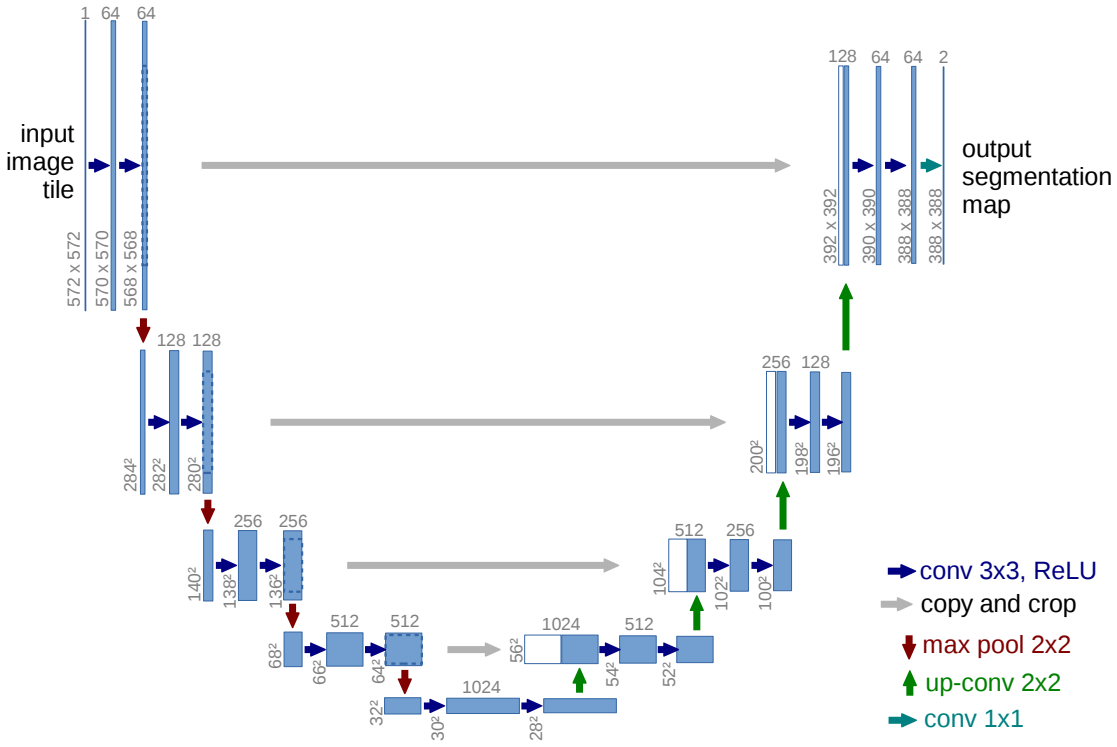

Figure 2.3 The architecture of U-Net for semantic segmentation. From [110].

2.2 Spatial correspondence in optical flow Estimation

The optical flow is a pattern used to describe the pixel's motion in the image sequence. Recently, it is widely used in many video-based computer vision tasks, *e.g.*, action recognition, video semantic segmentation, and tracking. The optical flow estimation needs precise per-pixel displacement, *i.e.*, finding correspondences between two input images. The traditional approaches [58] for optical flow estimation usually optimize a complex energy function to find the best matching between the pixels of two images. However, it is computationally expensive and always falls in a region that has occlusion, illumination varying, or noise.

In 2015, Dosovitskiy *et al.* [30] firstly adopt CNNs to learn optical flow and propose the FlowNet for data-driven optical flow estimation. The architecture of FlowNet is shown in Figure 2.4.

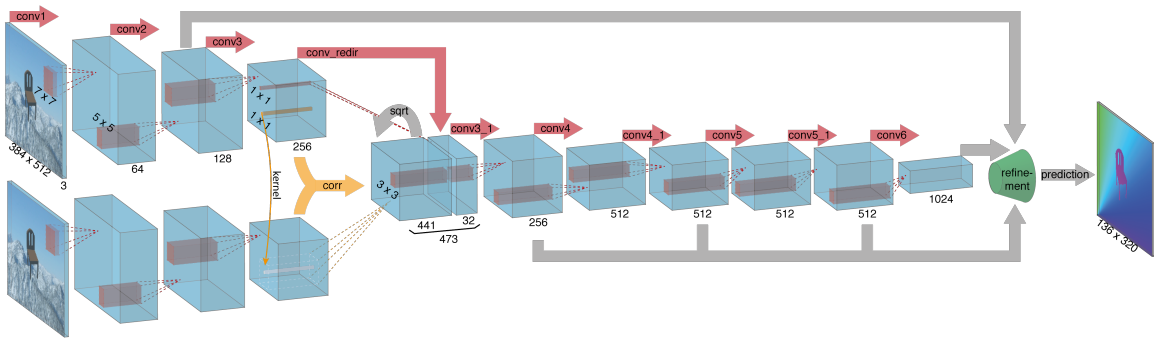

Figure 2.4 The architecture of FlowNet. From [30]. The correlation layer is used to perform the patch-wise comparisons between feature maps from two images.

Firstly, the FlowNet takes two image as input and extract the features separately. Secondly, it combine the features with a correlation layer. More specifically, let us define f_1 and f_2 are two feature from two images. Given a maximum displacement d, the correlation of two patches x_1 in f_1 and x_1 in f_2 can be calculated as

$$c(x_1, x_2) = \sum_{o \in [-d,d] \times [-d,d]} < f_1(x_1), f_x(x_2 + o) > \tag{2.4}$$

in a square patch of size $2d+1$. Finally, in order to provide dense per-pixel predictions, the FlowNet refines the coarse representation and up-sample it to original size of input image.

Recently, more works are proposed for building the spatial correspondence for optical flow estimation. For instance, RAFT [123] builds the multi-scale 4D correlation volumes, and then aggregate them in a pre-defined flow field, shown in the Figure 2.5.

2.3 Advanced convolutional layers

With the great success of the classic CNNs in many computer vision tasks, many researchers have been devoted to modifying the convolutional layers for further performance improvements [27, 50, 65, 87, 119, 128, 165]. Different from the traditional

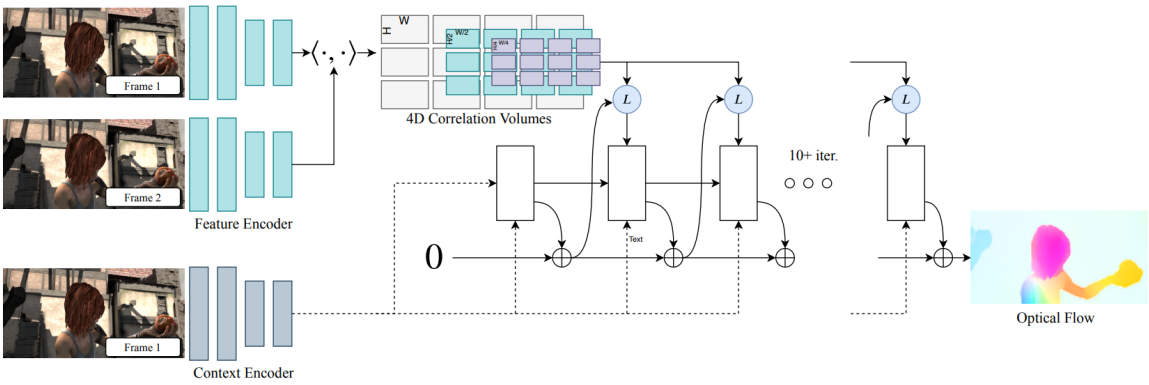

Figure 2.5 The architecture of RAFT. From [123].

convolutional layers where convolutional filters are invariant to input images after training, Xu *et al.* [65] introduces a dynamic filter network (DFN), in which one branch is used to generate filters conditioned on the input while the other branch uses the predicted filters to compute the final output. Similarly, Dai *et al.*develop a deformable convolutional network (DCN) [27] to dynamically change the receptive field in filtering the current features. As shown in Figure 2.6, the receptive field $\mathcal{R}$ in DCN is augmented with offsets $\{\Delta \mathbf{p}_n | n = 1, ..., N\}$, where $N = |\mathcal{R}|$. Then the convolution operation becomes:

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta \mathbf{p}_n). \tag{2.5}$$

Later in [165] the performance of DCN is further improved by learning the weight at each position. In [119], pixel-adaptive convolution (PAC) is proposed to modify the original filter weights by using local features computed from a guidance branch with a fixed parametric kernel function:

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{K}(F_{\mathbf{p}_0}, F_{\mathbf{p}_0 + \mathbf{p}_n}) \cdot \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n), \tag{2.6}$$

where K is a pre-defined kernel function such as a Gaussian kernel, and F is another feature map extracted from the image. Based on [119], Wannenwetsch *et al.* [133]

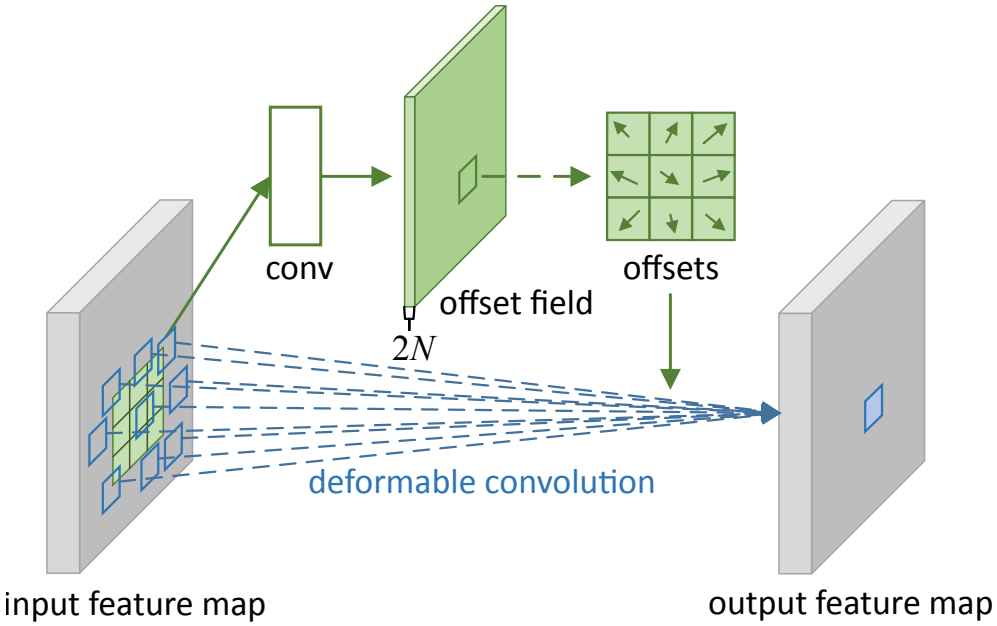

Figure 2.6 An illustration of the deformable convolution. From [27]. The receptive field is changed by using an additional convolutional layer.

propose a probabilistic pixel-adaptive convolution to provide location information and handle boundary artifacts. More interestingly, Harley *et al.* [50] integrate segmentation cues into a convolutional neural network by first using the segmentation labels to supervise the process of feature embedding, and then using the embeddings to construct masks for the positions on the feature maps.

2.4 Depth Estimation Benchmark

KITTI [36] is the most famous dataset for depth estimation which is collected using a vehicle equipped with a sparse Velodyne LiDAR scanner and high resolution video cameras as shown in Figure 2.7. It is captured by driving in rural areas and on highways in Karlsruhe, and there are up to 15 cars and 30 people in each image.

KITTI dataset is proposed for the tasks of stereo matching, depth estimation, semantic segmentation, optical flow, visual odometry (SLAM), 3D tracking, and 3D

Figure 2.7 A vehicle equipped with a Velodyne LiDAR scanner, a stereo rig and GPS devices for collecting the KITTI dataset [36].

object detection. Specifically, the stereo 2012 benchmark consists of 194 training scenes and 195 test scenes, while The stereo 2015 benchmark consists of 200 training image pairs and 200 test image pairs. The depth prediction part contains over 93,000 depth maps with corresponding RGB images.

Cityscapes [24] is a very large dataset and focuses on semantic understanding of real urban street scenes. For the depth estimation task, it provided pre-computed depth maps using SGM [53]. It has 5,000 pairs in total, and they are split into 2,975/500/1,525 for training/validation/testing. Note that the Cityscapes also provides the stereo pairs which can be used for stereo matching.

Compared with the street scenes collected by KITTI [36] and Cityscapes [24], Make3D [111] provides diverse outdoor scenes in the city of Palo Alto. It includes a total of 534 pairs of images and depth maps. The resolution of the image and depth map are 2272×1704 and 55×305, respectively.

NYU Depth Dataset V2 [115] was introduced for indoor depth estimation. It uses

a Microsoft Kinect RGBD camera to collect the depth maps. There are 1,449 pairs of aligned RGB and depth images including commercial and residential buildings from three different cities in the United States. In total, it has 464 different indoor scenes.

Recently, Vasiljevic *et al.* [129] proposed new dataset named as DIODE. It contains thousands of color images with accurate, dense depth maps for both indoor and outdoor scenes. The data was collected with the FARO S350 laser scanner to record 360° panoramic scans. The resolution of images and depth maps is 1,024 × 768. This dataset also provides the surface normal map for each image, which can benefit the 3D reconstruction.

Chapter 3

Literature review

This chapter provides a literature review of the works related to this dissertation, including stereo matching, depth estimation from monocular video, and single image depth estimation.

3.1 Stereo Matching

Almost all recent state-of-the-art performances of stereo matching are achieved by using CNN-based architecture. For example, in [89, 39], disparity value is discretized and disparity estimation is reduced to classification with CNNs. In [91], CNN is used for computing disparity map and optical flow simultaneously. This result can be refined iteratively based on error maps [97].

In [113], the disparity is estimated by patch matching. In GC-Net [71], cost volumes are regularized by 3D convolutions before being used for disparity estimation. Specifically, as shown in 3.1, they first extract features of the left and right images with a weight-sharing encoder and from the cost volume by concatenating the left and right feature maps across each disparity level. Then, the cost volumes are aggregated with 3D convolution layers. Finally, the disparity values are regressed from the 3d cost volume using a soft argmin operation.

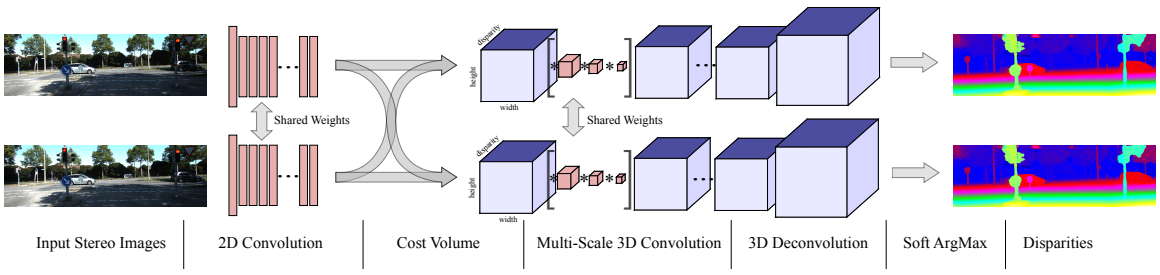

Figure 3.1 The architecture of GCNet. From [71].

Based on GC-Net [71], more and more end-to-end networks for stereo matching are proposed. In Stereonet [73], the use of low-resolution cost volumes leads to sub-

pixel matching accuracy and real-time speed. In [22], a new 3D convolutional module, as well as a sparse depth map, is used for improving stereo matching. PSMNet [14] extracts multiscale image information for constructing a single cost volume, which is then taken for regularization and disparity estimation. The pipeline of PSMNet is shown in Figure 3.2.

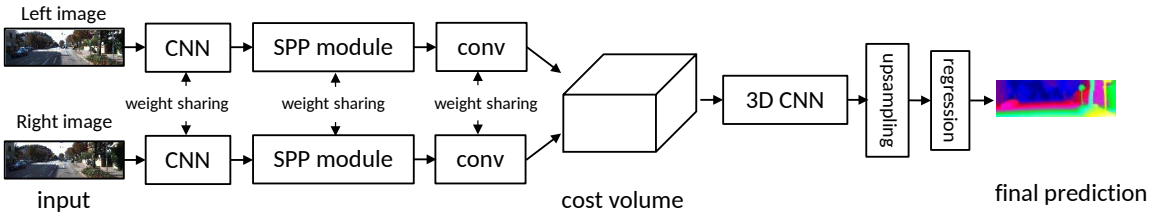

Figure 3.2 The architecture of PSMNet. From [14]. The multi-scale image information is constructed by SPP module.

In EdgeStereo [118], edge detection is incorporated to accurately estimate depth change across object boundaries. GWcNet [47] aggregate the 3d information with two cost volumes, where the first one is the concatenation of the left and right features, and the second one is constructed by group-wise correlation. For accurate stereo matching, HD^3-Stereo [147] estimates the disparity map and the model-inherent uncertainty map simultaneously. GANet [156] proposes two novel components during 3d cost volume aggregation, where the first one builds the semi-global matching, and the second one focuses on the local region.

AANet [140] uses the deformable convolution [27] to learn content-adaptive weights for each pixels in cost aggregation. CFNet [114] also uses uncertainty learning for robust stereo matching and fuses multiple low-resolution cost volumes to enlarge the receptive field. DSMNet[155] adds a normalization layer and a non-local layer which regulates the feature's distribution for better domain generalization ability. By applying the neural architecture search, LEAStereo [23] almost outperforms all state-of-the-art deep stereo matching approaches.

3.2 Depth Estimation from Monocular Video

Depth estimation from monocular videos has attracted much interest in recent years. In [69], handcrafted features were matched between frames for depth estimation and optical flow is also used to improve the depth estimation accuracy. Zhou *et al.* [162] trained a network to estimate the relative camera pose between adjacent frames and then fed it to another network for depth estimation. As shown in Figure 3.3, two networks (Pose CNN and Depth CNN) are involved in the joint training framework.

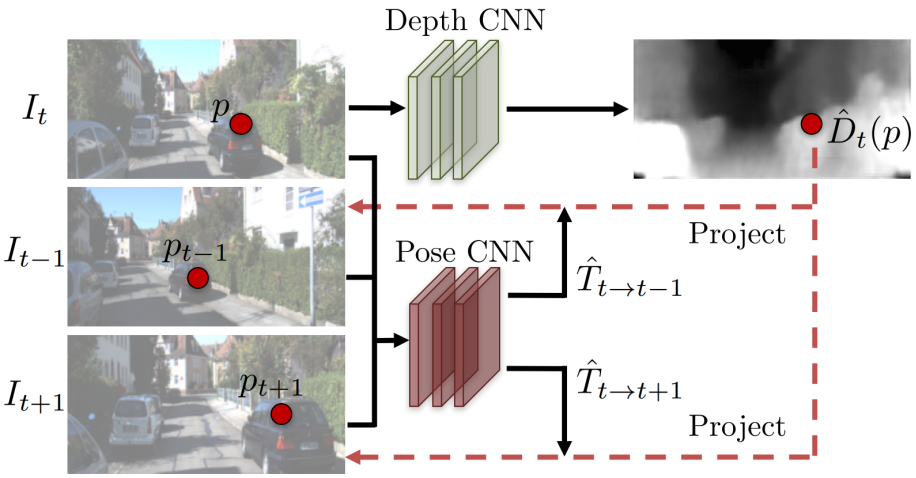

Figure 3.3 An illustration of the monocular depth and camera motion estimation from unstructured video sequences. From [162].

DeepV2D [122] estimates the relative camera poses between a keyframe and a set of nearby frames and finally generates a fused depth map on the keyframe. DeepTAM [161] estimates the relative camera pose and uses it to propagate the known depth map of a keyframe to other frames. Mahjourian *et al.* [90] combined camera-pose estimation and depth estimation in a single network by enforcing the 3D geometry consistency. Yin *et al.* [148] considered camera pose estimation, depth estimation, and optical flow in a unified network named GeoNet shown in Figure 3.4. We can see that all these methods need to estimate camera-pose change between frames.

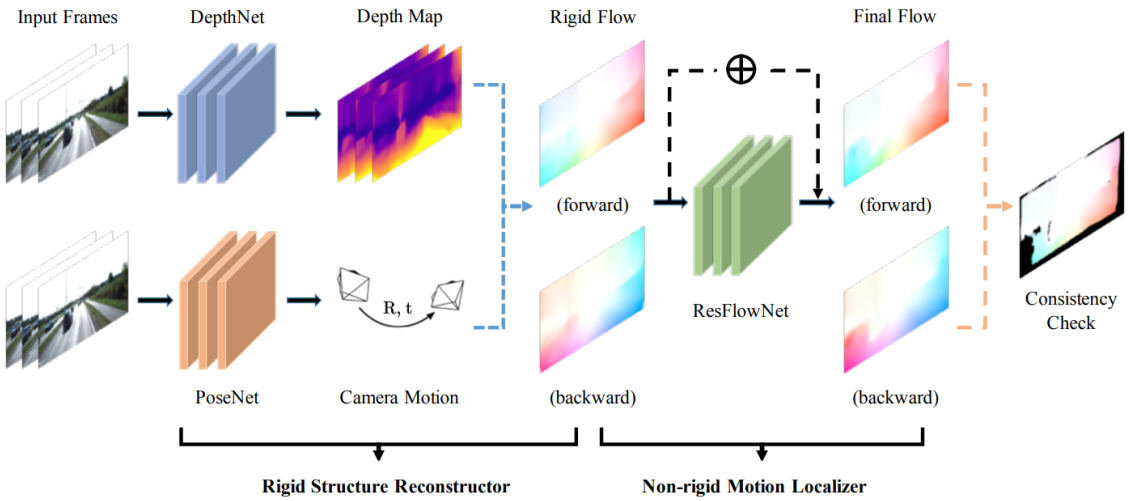

Figure 3.4 The architecture of GeoNet. From [148].

Recently, there are many methods [41, 134, 135, 136] focus on self-supervised approaches, however, the performance is still lower than the fully supervised approaches. Specifically, MonoDepth [41] treats depth estimation as an image reconstruction problem during training, where the left-right consistency of stereo images is used as a new constraint. Taking monocular video as input during training, MonoDepth2 [42] proposes a new objective function to handle occlusions across the frames and an auto-masking approach to ignore the noise and boost training. Watson *et al.* [134] observes that the re-projection is reliable for the regions with repeating patterns and textures-less areas, so they proposed to use depth hints obtained from other off-the-shelf stereo algorithms as labels for supervision. ManyDepth [135] makes use of multiple frames during testing and propose a multi-frame depth estimation model which combines the strengths of monocular and multi-view depth estimation. DistDepth [136] proposes to distill the 3d structure knowledge from a supervised model into a self-supervised depth estimator.

3.3 Single Image Depth Estimation

As large-scale datasets (e.g. KITTI [37]) are available, more and more supervised approaches have been developed for monocular depth estimation. Eigen *et al.* [32] propose a multi-scale deep network to learn the depth of a single image by global coarse prediction and local refinement. Following this paper, several further works have been developed to extract features using deep learning techniques. Xu *et al.* [139] designs a new framework using continuous conditional random fields (CRFs) to fuse the multi-scale byproducts of the network.

Fu *et al.*propose DORN [33] by discretizing the depth and converting the regression problem into a multi-class classification problem, which achieves the state-of-the-art performance. The detailed structure of DORN is illustrated in the Figure 3.5

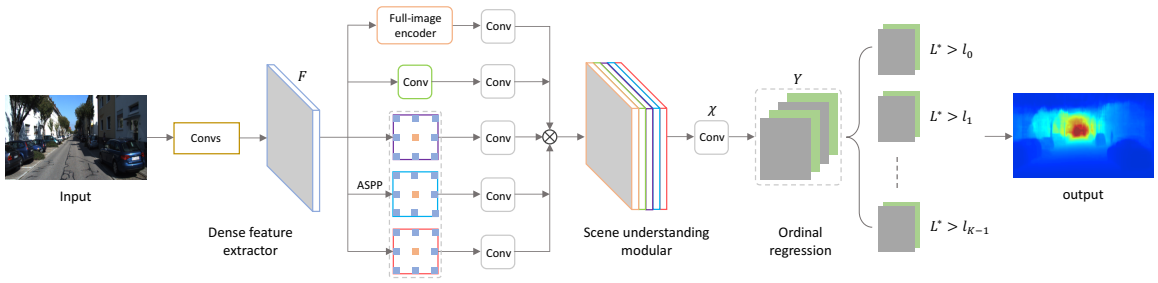

Figure 3.5 The architecture of DORN. From [33].

Jiao *et al.* [66] propose an attention-driven loss and a synergy network to mutually improve the depth estimation and semantic labeling tasks. Chen *et al.* [19] develop a conditional GAN framework with a novel patch-wise loss function to predict depth values at the patch level by incorporating more global information. Gan *et al.* [34] explicitly model the relationship across different pixels using an additional affinity layer to model the depth relation of neighboring pixels. Recently, Yin *et al.* [145] propose virtual normal directions to incorporate geometric constraints in the 3D space to improve the depth prediction accuracy.

In AdaBins [6], Bhat *et al.*divide the depth range into adaptive bins and the final depth map comes from a linear combination of the bin centers MiDaS [105] introduces a new loss function which is invariant to depth range and scale, which enables the training of the model on diverse training sets from different sources. To address the unknown depth shift, Wei *et al.*[146] improve the estimated depth by training models on synthetic 3D data or data for 3D scene shape priors. BoostingDepth [94] proposes a new loss function which focuses on boundary accuracy, and it improves the depth estimation for high-resolution images. DPT [104] applies the self-attention blocks for depth estimation with more globally coherent predictions.

Chapter 4

Semantic Stereo Matching with Pyramid Cost Volumes

4.1 Motivation

Stereo matching is indispensable for many computer vision applications, such as autonomous driving [15], 3D reconstruction [154], augmented realities [18], and robot navigation [7]. By finding pixel-level correspondence between two images, stereo algorithms aim to construct a disparity map from a pair of rectified stereo images. In traditional methods, hand-crafted reliable features are used to identify cross-image matching pixels or patches for computing the disparity map [10, 107]. Recently, as in many other computer vision tasks, convolutional neural networks (CNNs) have been applied to stereo matching with significant success.

When applying CNNs for stereo matching, many of the existing works construct a cost volume for computing the correspondence cost at each position by traversing a set of possible disparity values. A regression layer is then used to infer the optimal disparity map based on the cost volume. While early works calculate the cost in the original image domain [54, 55, 124], recent works construct the cost volume using the deep features extracted by the respective networks [71, 97, 118, 14]. For these prior works, the cost volume is constructed at a single level without considering multiscale spatial information separately underlying the stereo image pairs. However, for the considered feature map, a single-scale cost volume may not be sufficient to capture the spatial relationship between stereo images. One of our major ideas in this work is to develop a new CNN network with multilevel cost volumes, which we call *pyramid cost volumes*, for better capturing the disparity details in stereo matching.

Multiscale information has been used in many CNN-based computer vision applications. For example, PSPNet [160] and DeepLab [17, 16] embed multiscale features of scenes to improve semantic segmentation. SPyNet [106] calculates optical flow by warping images in multiple scales. PWC-Net [120] uses multiscale features to compute optical flow with a single branch. Different from these works, we here introduce the multiscale information into stereo matching, as in PSMNet [14]. But as

discussed above, PSMNet constructs a single cost volume using multiscale features, while we construct multilevel cost volumes directly, resulting in much better disparity estimation.

Our work is also partly inspired by the recent work of SegStereo [141] that integrates semantic information to stereo matching through joint learning. Semantic information has been found to be useful when integrated to solve many important computer vision problems. For example, in [21] an integrated SegFlow model is developed to address optical flow and video segmentation together, leading to a win-win result. In [67, 158, 70], two tasks of monocular depth estimation and semantic segmentation are solved simultaneously by using weight-sharing sub-networks or joint CNN learning. SegStereo [141] combines semantic and image features into a single cost volume for disparity estimation.

The semantic segmentation captures different objects and their boundaries in images and shows much spatial and intensity correlation with the disparity map. In particular, an accurate semantic segmentation can help rectify the disparity values along the object boundaries, which are usually more prone to error in stereo matching [8, 46]. Thus, our network will also integrate both the semantic and the spatial information in multiple levels for constructing pyramid cost volumes, and we find that such an approach can improve the stereo-matching accuracy significantly.

More specifically, we design a new semantic stereo network named SSPCV-Net for stereo matching. In this network, after several initial convolutional layers, we take the extracted deep features as input for two separate branches. One of them performs the traditional spatial pooling, but with hierarchical multilevel processing. The other branch is a semantic segmentation subnetwork. We then build pyramid cost volumes by combining the outputs of these two branches from input stereo pairs such that these new pyramid cost volumes well represent both semantic and spatial information in multiple levels. Next, we design a 3D multi-cost aggregation module

to integrate the extracted multilevel features and perform regression for predicting disparity maps.

We employ a two-step strategy to train the SSPCV-Net: 1) supervised training of the semantic segmentation subnetwork; and 2) joint training of the whole network with supervision on both semantic segmentation and disparity estimation. We conduct comprehensive experiments, including a series of ablation studies and comparison tests of SSPCV-Net with existing state-of-the-art methods on Scene Flow, KITTI 2015 and KITTI 2012 benchmark datasets. It is observed that the proposed SSPCV-Net clearly outperforms many existing state-of-the-art stereo-matching methods.

4.2 Method

The architecture of the proposed SSPCV-Net is shown in Figure 4.1. We can see that the main pipeline includes: (a) feature extraction: using ResNet50 [51]; (b) spatial pooling: using average pooling, the resulted multilevel feature maps are fed into the semantic segmentation network; (c) multi-cost aggregation: a new pyramid cost volumes are built to incorporate semantic information and multilevel spatial context information. In addition, a 3D multi-cost aggregation module is added for cost-volume aggregation; (d) disparity regression: disparity map is estimated from the cost volumes using 3D convolution.

4.2.1 Network architecture

We first use ResNet-50 [51] with the dilated network strategy [16, 150] to extract features from the input pair of images, and then adopt adaptive average pooling to compress features into three scales, followed by a 1×1 convolution layer to change the dimension of the feature maps. The resulting spatial features are simultaneously fed into two branches of the network – one branch produces spatial pyramid cost

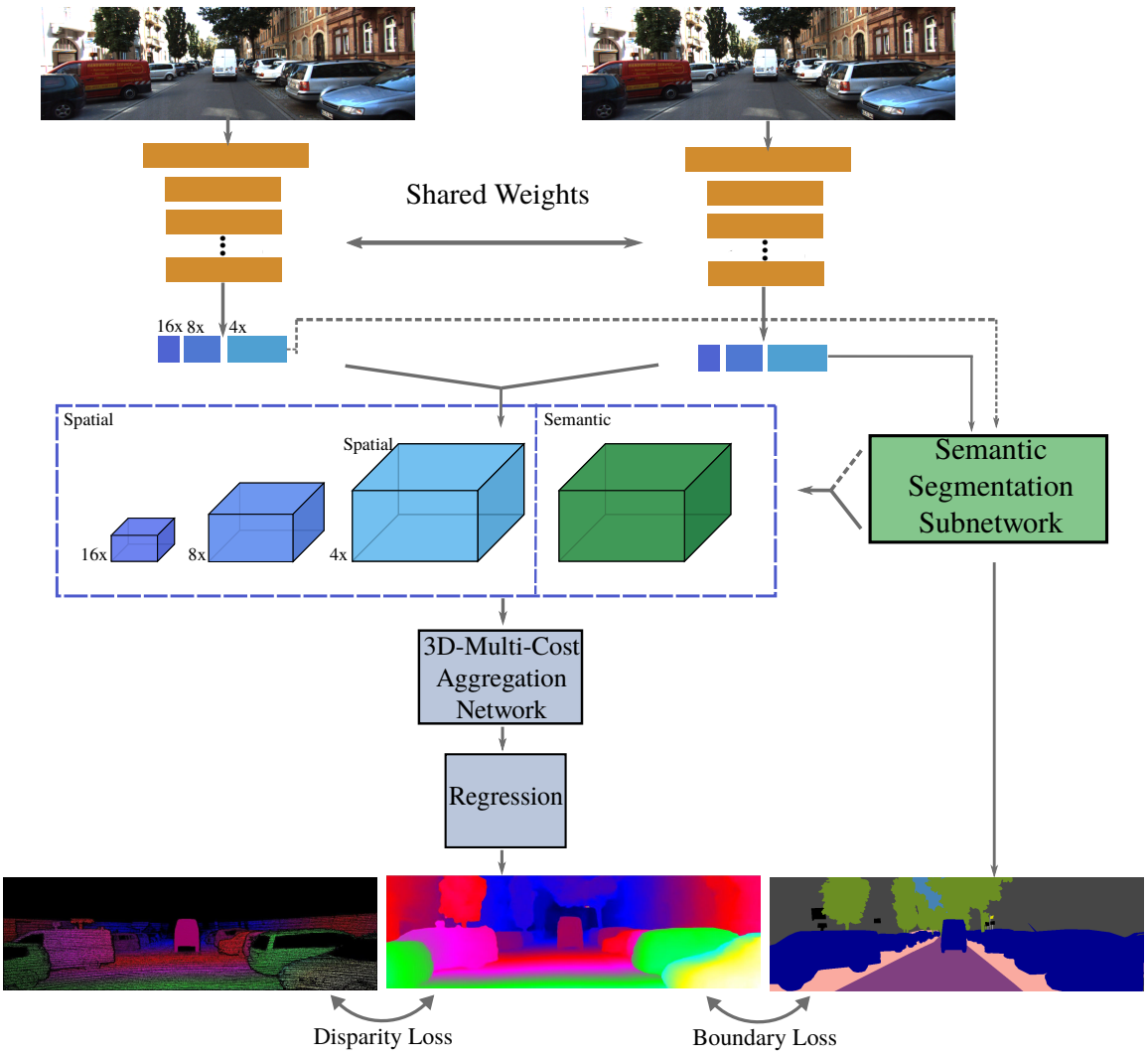

Figure 4.1 Architecture of the proposed semantic stereo network for disparity estimation. It consists of feature extraction, spatial pooling, semantic segmentation sub-network, multi-cost aggregation, and disparity regression.

volumes directly and the other branch is a semantic segmentation subnetwork, which generates a semantic cost volume. The obtained semantic cost volume and the spatial cost volumes make up pyramid cost volumes, as shown in the box of *Pyramid Cost Volumes* in Figure 4.1. All these cost volumes are then fed into a 3D multi-cost aggregation module for aggregation and regularization. At the end, a regression layer produces the final disparity map. The pyramid cost volumes and the 3D multi-cost

aggregation module are elaborated in the following sections.

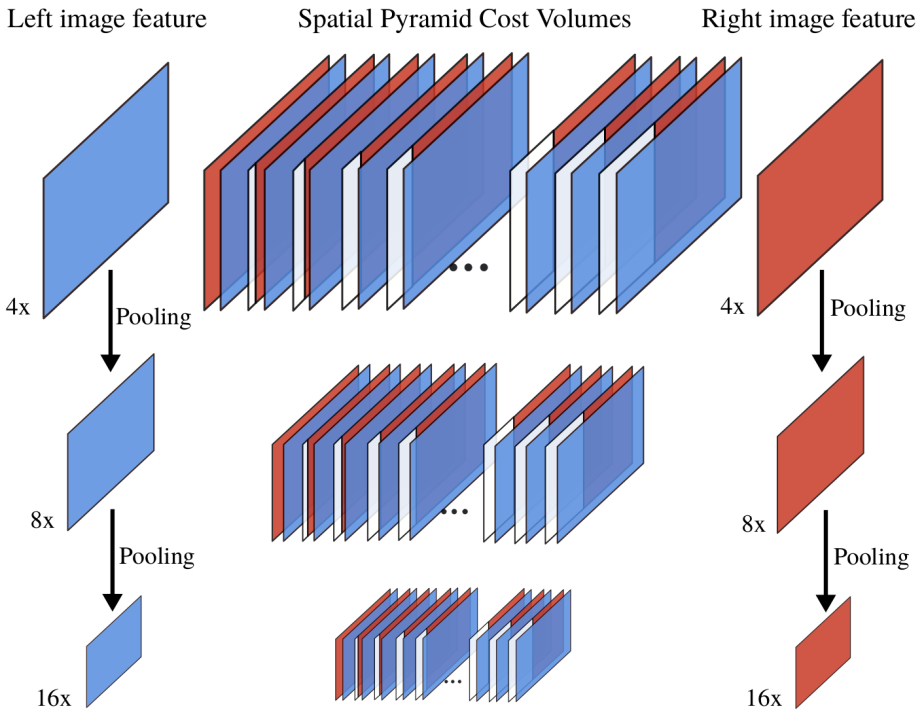

Figure 4.2 The construction process of spatial pyramid cost volumes.

4.2.2 Spatial pyramid cost volumes

We propose to use the idea of pyramid cost volumes to learn the relationship between an object and its neighbors in space. Different from PSMNet, where only a single cost volume is generated from the pyramid features by first upsampling them to the same dimension and then performing concatenation, we instead use multilevel spatial features to build spatial pyramid cost volumes.

We use hierarchical scales of spatial features after different adaptive average pooling layers in feature extraction to form levels of cost volumes. Following the idea of GC-Net [71], for each level of the spatial feature maps, we form a cost volume by concatenating the corresponding unaries from the left and right image features and

then packing them into a 4D volume, which contains all spatial context information for inferring disparity from this level.

As shown in Figures 4.1 and 4.2, three hierarchical levels of feature maps are particularly used in our SSPCV-Net to form spatial pyramid cost volumes to represent different level of information, and the spatial pyramid cost volumes have sizes of $C \times \alpha W \times \alpha H \times \alpha D$ with $\alpha \in \{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}\}$ respectively at each level, where C is number of channels, W and H are the width and height of original images respectively, and D is the maximum disparity.

4.2.3 Semantic cost volume

For the semantic branch, the semantic segmentation sub-network follows PSPNet [160]. With the extracted feature maps, the sub-network upsamples the low-dimensional feature maps to the same size and concatenates all the feature maps. In the end, it is followed by a convolution layer to generate the final prediction of the semantic segmentation map.

To form the single semantic cost volume, we use the features before the classification layer. The use of semantic cost volume aims to capture context cues in a simple manner and learn the similarity of objects' pixels from the left and right semantic segmentation features. By concatenating each unary semantic feature with their corresponding unary from the opposite stereo image across each disparity level, and packing them into a 4D volume, we obtain a semantic cost volume with the size of $C \times \frac{1}{4}W \times \frac{1}{4}H \times \frac{1}{4}D$, which is the same size as the largest spatial cost volume.

4.2.4 3D multi-cost aggregation module

As shown in Figure 4.3, both the spatial pyramid cost volumes and the semantic cost volume are fed into the 3D multi-cost aggregation module.

We use a "Hourglass" module (Figure 4.4) and a 3D feature fusion module (FFM)

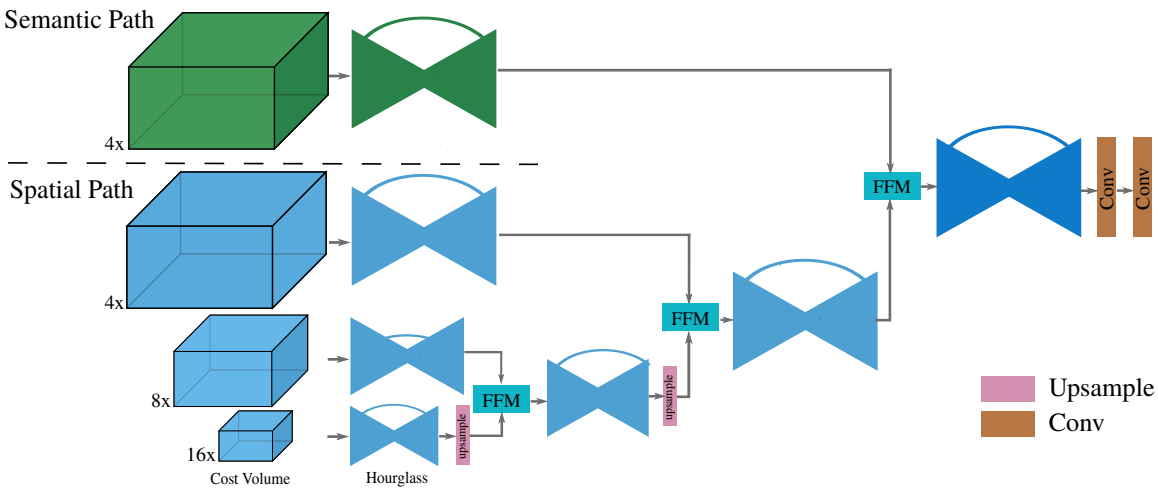

Figure 4.3 Details of the 3D multi-cost aggregation module with the hourglass and the 3D feature fusion.

to learn different levels of spatial context information through the encoding/decoding process.

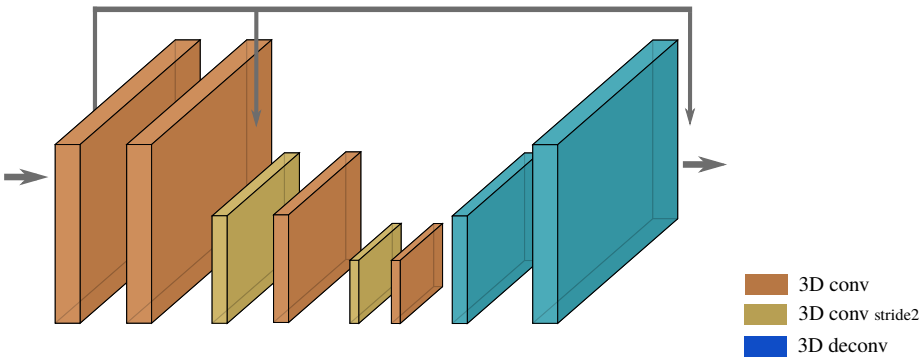

Figure 4.4 The structure of “Hourglass" module.

As for the strategy, inspired by the MSCI (multiscale context intertwining) scheme in [82] and RefineNet [83], we fuse the 4D spatial cost volumes from the lowest level to the higher ones in a recursive way: we first upsample the lower level volume to the same size as its immediately higher level one and feed them into FFM, then the fused cost volume is further fused with the next higher level cost volume after the

hourglass module. Finally, the last level fused spatial cost volume is fused with the semantic cost volume and the result is then upsampled to the original image size $1 \times W \times H \times D$ via the bilinear interpolation.

Instead of concatenating the features as in BiSeNet [149], which includes a 2D feature fusion module to help the context information fusion, we develop a 3D feature fusion module specifically for fusing two cost volumes: first the two 3D cost volumes are summed up following the residual block structure in [51], next the adaptive average pooling is used to transform the concatenated features to a feature vector and then a weight vector is computed through a fc-ReLU-fc-sigmoid structure [59], finally, the upsampled one of the two cost volumes is multiplied by the weight vector and added with the other cost volume to form the output of the FFM module. The details of FFM are shown in Figure 4.5)

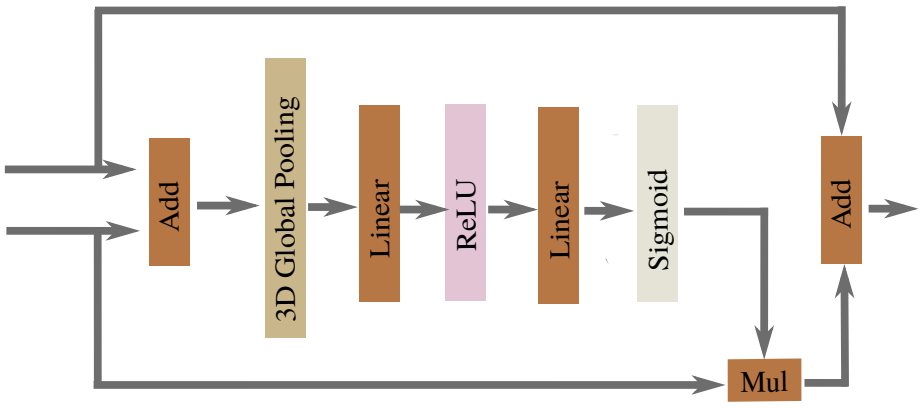

Figure 4.5 The structure of FFM module.

4.2.5 Disparity regression and loss function

We take the disparity regression proposed in [71, 14] to estimate the continuous disparity map. The softmax operation $\sigma(\cdot)$ is first used to normalize the finally fused cost volume C_d to output a probability $P(d)$ for each disparity d, which is regarded to as a soft attention mechanism and often more robust than classification-based

approaches. The predicted disparity $\hat{d}$ is then calculated as the sum of each disparity d weighted by its probability as

$$\hat{d} = \sum_{d=0}^{D_{max}} d \times P(d) \tag{4.1}$$

where D_{max} denotes the maximum disparity.

To train the proposed architecture, we rely on the following multi-task loss function.

$$L = \alpha L_{disp} + (1 - \alpha) L_{bdry} \tag{4.2}$$

which consists of the weighted sum ($0 \leq \alpha \leq 1$ is the weight) of two terms, namely the disparity loss (L_{disp}) and the boundary loss (L_{bdry}).

We use the $smooth_{L_1}$ as the basic loss function to train our proposed SSPCV-Net which has been widely used in many regression tasks [40, 71]. The disparity loss is defined as

$$L_{disp}(d^*, \hat{d}) = \frac{1}{N} \sum_{(i,j)} smooth_{L_1}(d^*_{i,j}, \hat{d}_{i,j}) \tag{4.3}$$

where N is the number of all the labeled pixels, d^* is the disparity ground-truth.

Since the disparity discontinuity point is always on the semantic boundaries [103], we accordingly deploy the following boundary-loss function as

$$L_{bdry} = \frac{1}{N} \sum_{(i,j)} \left(|\varphi_x(sem_{i,j})| e^{-|\varphi_x(\hat{d}_{i,j})|} + |\varphi_y(sem_{i,j})| e^{-|\varphi_y(\hat{d}_{i,j})|} \right) \tag{4.4}$$

where sem is the semantic segmentation ground-truth label, and φ_x and φ_y are the intensity gradients between neighboring pixels along the x and y directions, respectively.

4.3 Experiment

4.3.1 Datasets and evaluation metrics

In this section, we use the following stereo datasets for performance evaluation and comparison of SSPCV-Net with several recent state-of-the-art networks for stereo

matching:

Scene Flow [91]: This is a synthetic dataset consists of 35,454 training and 4,370 testing image pairs that can be used for evaluating optical flow and stereo matching performance. This dataset has dense and elaborate disparity maps as ground-truth for training.

KITTI 2015 & KITTI 2012 [92, 36]: These are two real-world datasets. KITTI 2015 contains 200 training stereo image pairs with sparse ground-truth disparities and another 200 testing image pairs without ground-truth disparities. The left (reference) images of the stereo image pairs have semantic labels. KITTI 2012 contains 194 training stereo image pairs with sparse ground-truth disparities and another 195 testing image pairs without ground-truth disparities. All these images have no semantic labels.

Cityscapes [24]: This is a large dataset of stereo image pairs focusing on urban street scenes. It contains 1,525 stereo image pairs for testing with ground-truth disparities precomputed using SGM.

Some metrics are used to evaluate the stereo matching performance. The measure of averaged end-point error (EPE) is defined by

$$EPE(d^* - \hat{d}) = ||d^* - \hat{d}||_2. \tag{4.5}$$

A pixel is considered to be an erroneous pixel when its disparity error is larger than t pixels, and the percentages of erroneous pixels in non-occluded and all areas are calculated. The percentages of erroneous pixel averaged over background & foreground regions and all ground-truth pixels are measured separately. Specifically, $t = 3$ is used for Scene Flow, Cityscapes and KITTI 2015, and $t \in \{2, 3, 4, 5\}$ for KITTI 2012. For all error metrics, the lower the better.

4.3.2 Model specification

We implemented the proposed SSPCV-Net based on PyTorch, and the training was done on two Nvidia 1080 GPUs with Adam (momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$). The stereo image pairs were randomly cropped into two kinds of size $(256 \times 512, 256 \times 792)$ before the training stage. The maximum disparity D_{max} was set to 256 for Scene Flow and 192 for KITTI 2015 & 2012.

For Scene Flow dataset, we trained our model from scratch using the training split with a constant learning rate of 0.001 and a batch size of 2 with $\alpha = 0.9$. The semantic segmentation subnetwork within SSPCV-Net was first trained for 40 epochs, where segmentation labels were transformed from object labels, then we did the joint training of the whole network for 40 epochs.

For KITTI 2015 & 2012, the model trained with SceneFlow was used (as pretrain) for further fine-tuning on the KITTI training dataset. The learning rate for both KITTI dataset trainings began at 0.01 and was reduced at a rate of 50% every 100 epochs. The semantic segmentation subnetwork was first trained with the KITTI 2015 dataset for 300 epochs. Then we did the joint training of the whole network for 400 epochs with $\alpha = 0.9$ for KITTI 2015, but with $\alpha = 1$ (i.e., the boundary loss term L_{bdry} was excluded from the loss function) for KITTI 2012 because of no semantic ground-truth available for use in KITTI 2012 dataset.

The overall training process took about 120 hours for the Scene Flow dataset and 70 hours for each of two KITTI datasets.

4.3.3 Comparisons with some existing networks

We compared the performance of SSPCV-Net with some state-of-the-art networks for stereo matching, including MC-CNN [151], DispNet v2 [46], iResNet-i2 [81], GC-Net [71], CRL [97], PSMNet [14], EdgeStereo [118], and Segstereo [141].

On Scene Flow – As reported in Table 4.1 for performance evaluation results on the Scene Flow dataset, SSPCV-Net obtained the best averaged EPE (0.87) and 3-pixel error in all pixels (D1-all) for all regions (3.1) and significantly outperformed all comparison methods in term of accuracy.

Table 4.1 Results of the performance comparison on Scene Flow dataset.

Method	GC-Net	iResNet-i2	CRL	PSMNet	EdgeStereo	SegStereo	SSPCV-Net
Averaged EPE	1.84	1.40	1.32	1.09	1.11	1.45	0.87
D1-all	9.7	5.0	6.7	4.2	-	3.5	3.1

The predicted disparity maps and corresponding errors of two examples by SSPCV-Net are illustrated together with the disparity maps by PSMNet in Figure 4.6, which visually demonstrates that SSPCV-Net can reach more accurate disparity maps especially at the edge of the objects. We also provide the 3d visualization on the Scene dataset in Figure 4.7.

On KITTI 2015 – Table 4.2 reports the performance evaluation results on the KITTI 2015 online leaderboard (by the KITTI evaluation server), in which the 3-pixel errors in estimated pixels (D1-est), background (D1-bg), foreground (D1-fg) and all pixels (D1-all) for all regions (ALL) and non-occluded regions (NOC) are computed. Clearly, SSPCV-Net achieved the best performance in terms of almost all error metrics except for the NOC D1-fg metric among all comparison methods. The leaderboard ranks the overall performance based on the ALL

D1-all metric, and SSPCV-Net obtained 2.11%, which is much better than other stereo matching networks. Moreover, we evaluated the semantic sub-network on KITTI 2015 and got an average IoU of 56.43% for each class and 82.21% for each category. For visual illustration, Figure 4.8 presents three examples of the disparity maps estimated by SSPCV-Net, PSMNet and GC-Net with the corresponding error maps. In Figure 4.9, we visualize the 3d reconstruction results of one sample with

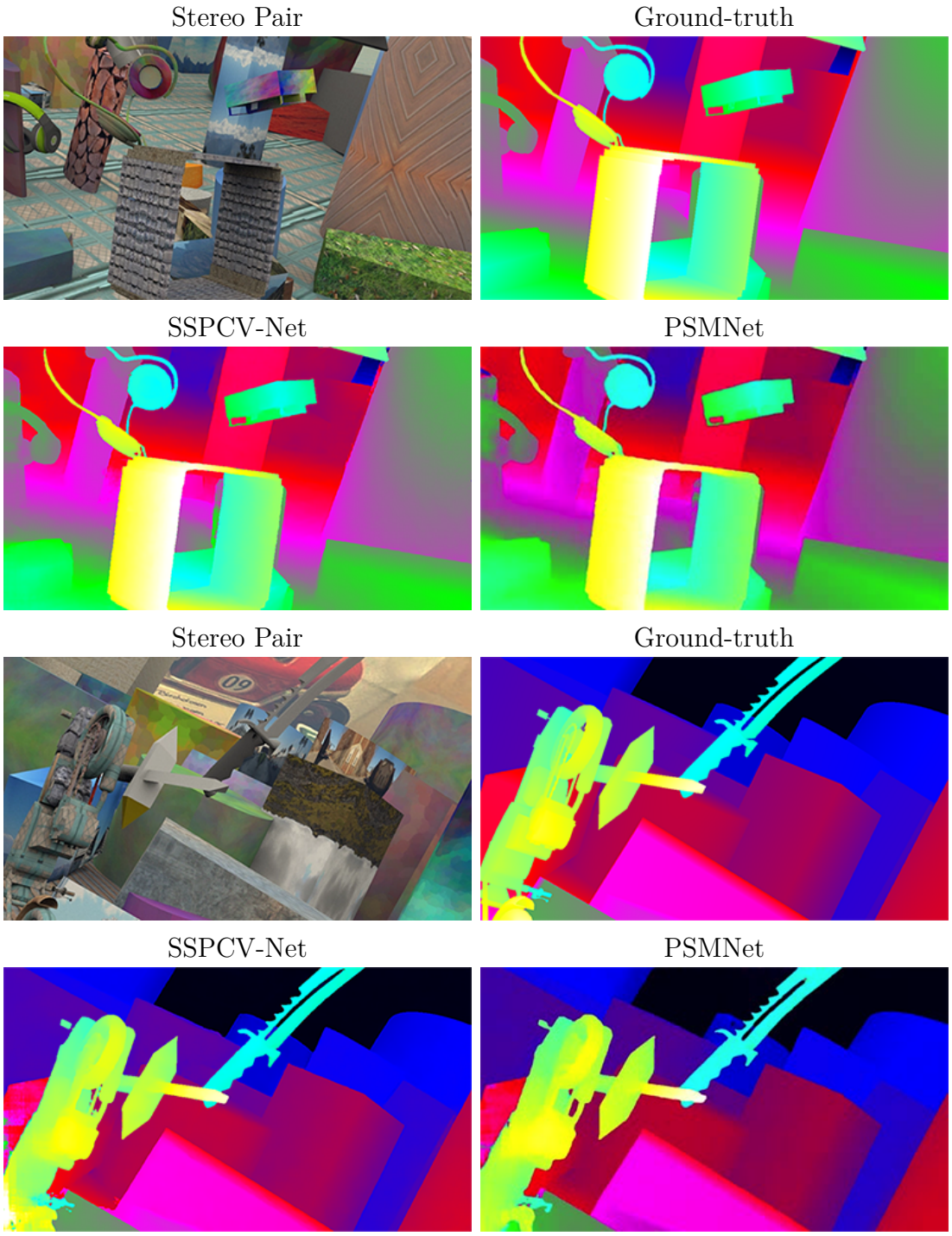

Figure 4.6 Two testing results from Scene Flow dataset. From left to right: the left input image of stereo image pair, the ground-truth disparity, the predicted disparity map by SSPCV-Net, and the predicted disparity map by PSMNet.

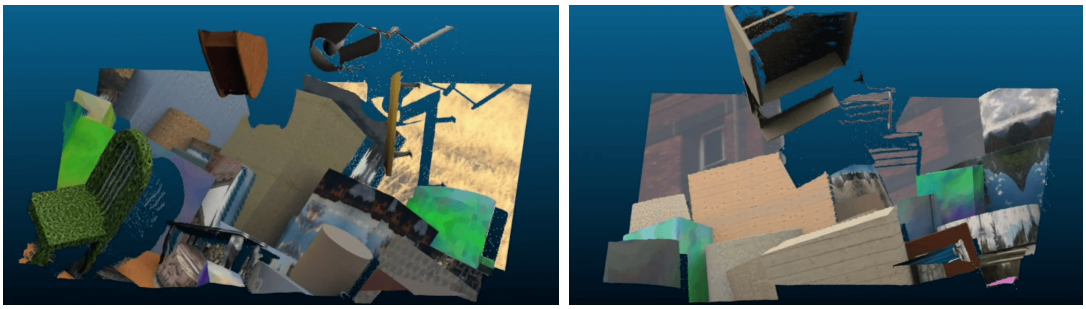

Figure 4.7 The 3d visualizations on the Sceneflow dataset. The 3d point clouds are reconstructed using the original image and the estimated depth.

Table 4.2 Results of the performance comparison on the KITTI 2015 dataset.

Method	ALL			NOC		
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all
MC-CNN [151]	2.89	8.88	3.89	2.48	7.64	3.33
DispNet v2 [46]	3.00	5.56	3.43	2.73	4.95	3.09
GC-Net [71]	2.21	6.16	2.87	2.02	5.58	2.61
CRL [81]	2.48	3.59	2.67	2.32	3.12	2.45
EdgeStereo [118]	2.27	4.18	2.59	2.12	3.85	2.40
PSMNet [14]	1.86	4.62	2.32	1.71	4.31	2.14
SegStereo [141]	1.88	4.07	2.25	1.76	3.70	2.08
SSPCV-Net	1.75	3.89	2.11	1.61	3.40	1.91

both RGB values and semantic categories.

On KITTI 2012 – Table 4.3 reports the performance evaluation results on the KITTI 2012 online leaderboard, in which the 2, 3, 4 and 5-pixel errors in all regions (Out-All) and non-occluded regions (Out-Noc) are evaluated.

Although the boundary loss term was excluded from the loss function for joint training, in this case, SSPCV-Net still achieved the best performance in five error metrics out of a total of eight among all comparison methods, and did just very slightly worse than PSMNet in two and EdgeStereo in one of the remaining three

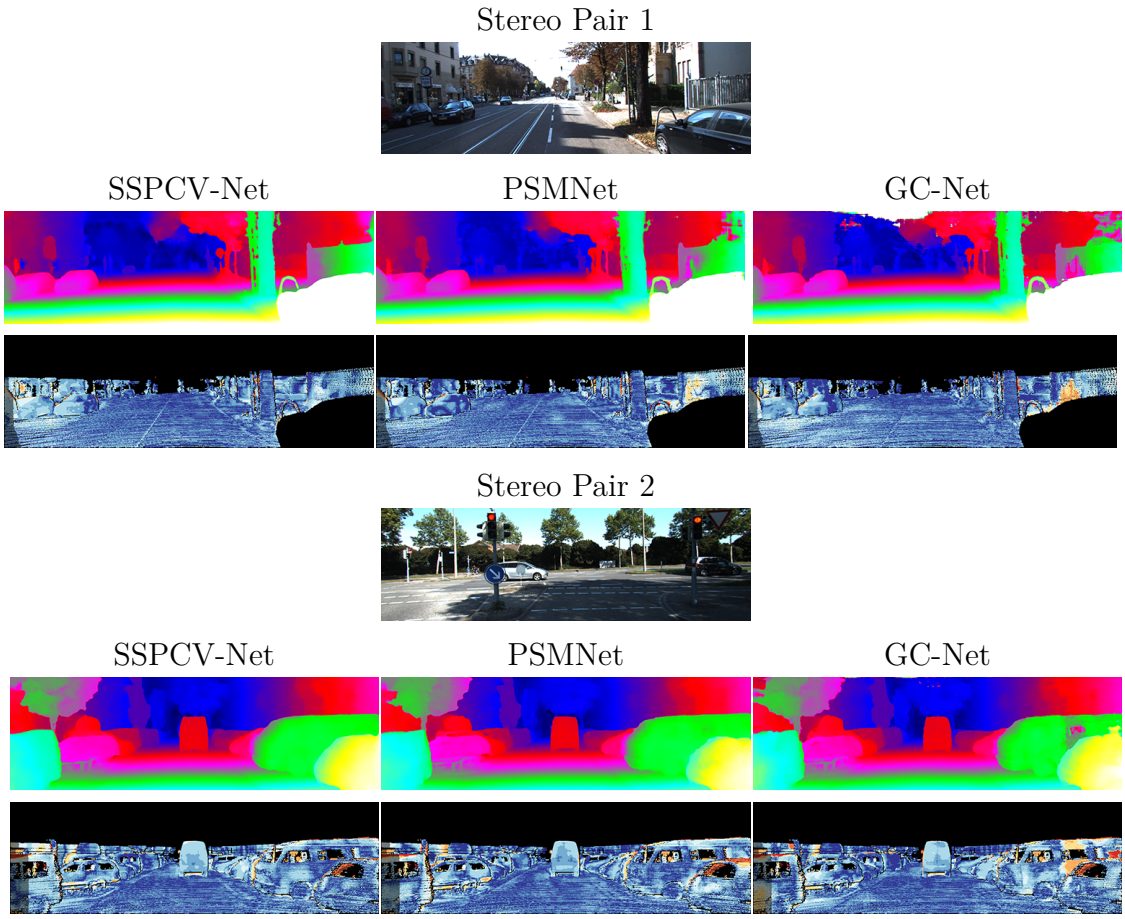

Figure 4.8 Two testing results from KITTI 2015 dataset. For each input image pair, the predicted disparity and corresponding error maps obtained by SSPCV-Net, PSMNet and GC-Net are presented.

error metrics. Figure 4.10 visually illustrates two examples of the predicted disparity maps produced by SSPCV-Net, PSMNet and GC-Net, and it again shows SSPCV-Net can give more reliable and accurate results, especially in ambiguous regions. The Figure 4.11 shows the 3d reconstruction results.

On Cityscapes – To evaluate the generalization ability, we used the test split of Cityscapes to test the models which were all trained on Scene Flow and KITTI 2015 (without any training on the Cityscapes dataset). Note that the channel of cost volumes for all compared methods was set to be 16 in experiments.

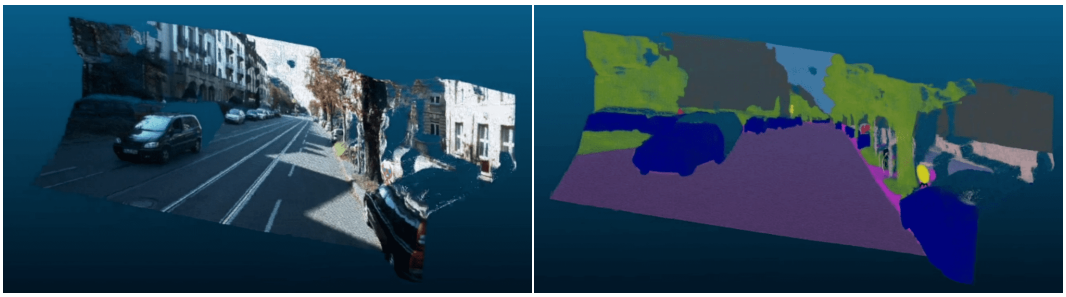

Figure 4.9 The 3d visualization of one sample on the KITTI 2015 dataset. The 3d point clouds are reconstructed using the original image, semantic segmentation map and the estimated depth.

Table 4.3 Results of the performance comparison on KITTI 2012 dataset.

Method	2px		3px		4px		5px	
	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All
MC-CNN [151]	3.90	5.45	2.43	3.63	1.90	2.85	1.64	2.39
GC-Net [71]	2.71	3.46	1.77	2.30	1.36	1.77	1.12	1.46
PSMNet [14]	2.44	3.01	1.49	1.89	1.12	1.42	0.90	1.15
EdgeStereo [118]	2.79	2.43	1.73	2.18	1.30	1.64	1.04	1.32
SegStereo [141]	2.66	3.19	1.68	2.03	1.25	1.52	1.04	1.32
SSPCV-Net	2.47	3.09	1.47	1.90	1.08	1.41	0.87	1.14

Figure 4.12 shows two examples of the disparity maps estimated by SSPCV-Net, PSMNet, and GC-Net for visual comparison, which show SSPCVT-Net significantly outperformed PSMNet and GC-Net on the generalization ability. Predictions by the proposed SSPCV-Net are able to capture the global layout and object details (shape & edge) quite well. The 3d reconstruction results in Figure 4.13 also verify the generalization ability of the proposed method.

4.3.4 Ablation studies

We first conducted ablation studies to compare a number of different model variants for SSPCV-Net on the Scene Flow dataset and the KITTI 2015 dataset (*without*

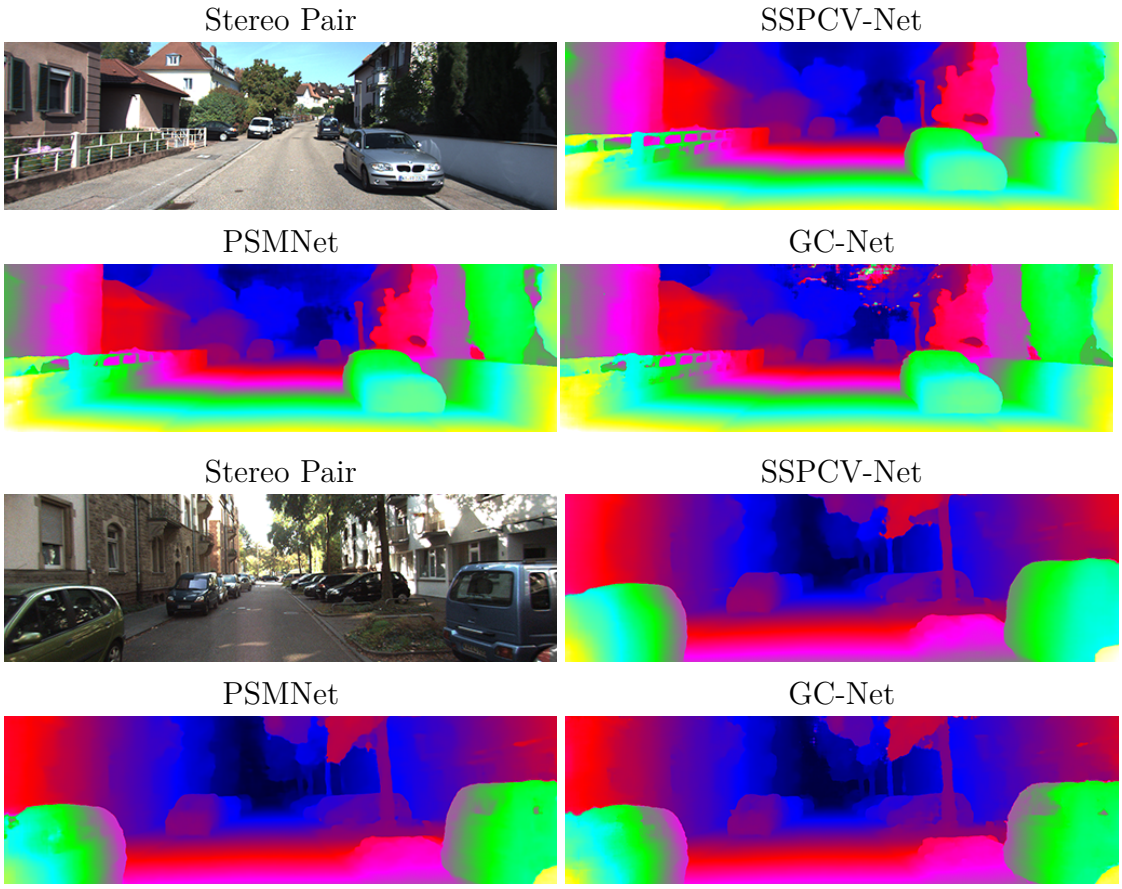

Figure 4.10 Two testing results from KITTI 2012 dataset. For each input image pair, the disparity maps obtained by SSPCV-Net, PSMNet and GCNet are presented.

pretraining from Scene Flow), respectively.

For KITTI 2015, we divided the origin training set into a training split (80%) and a validation split (20%) since the original testing set has no disparity ground truth provided. The importance of three key ideas in SSPCV-Net was evaluated: 1) adding a semantic branch, 2) using pyramid cost volumes and 3) dilated convolution in feature extraction. The results are reported in Table 4.4 and clearly justify our design choices for SSPCV-Net: pyramid cost volumes and the semantic information can promote the accuracy of disparity estimation, and the feature extraction has been improved when the dilated convolution strategy was used in the network.

Figure 4.11 Two 3d visualizations on the KITTI 2012 dataset. The 3d point clouds are reconstructed using the original image and the estimated depth.

Table 4.4 Comparison of a number of different model variants for justification of SSPCV-Net on SceneFlow validation dataset and KITTI 2015 validation datasets. The percentage of pixels with errors is used for KITTI 2015 evaluation and the averaged end-point error is used for Scene Flow evaluation.

	Semantic branch	Pyramid cost volumes	Dilated convoution	Scene Flow validation	KITTI 2015 validation
Single spatial cost volume				2.12	2.63
+Semantic branch	✓			1.76	2.42
+Semantic branch (Joint-train)	✓			1.78	2.37
+Spatial pyramid cost volumes		✓		1.21	2.11
+3D multiple cost volumes	✓	✓		1.04	1.99
SSPCV-Net (w/o FFM)	✓	✓	✓	1.07	2.10
SSPCV-Net (w/o L_{bdry})	✓	✓	✓	1.01	1.93
SSPCV-Net	✓	✓	✓	0.98	1.85

Some disparity maps regressed from SSPCV-Net by excluding certain cost volumes of different branches or levels are illustrated in the Figure 4.14, (a) Without the lowest-level spatial cost volume; (b) without the highest-level spatial cost volume; (c) without the semantic cost volume; (d) from the full-version SSPCV-Net. The lowest-level spatial cost volume helps improve the accuracy in small objects region and the highest-level spatial cost volume contains more context information and helps detect more scenes. The semantic cost volume helps produce better edge and better shape

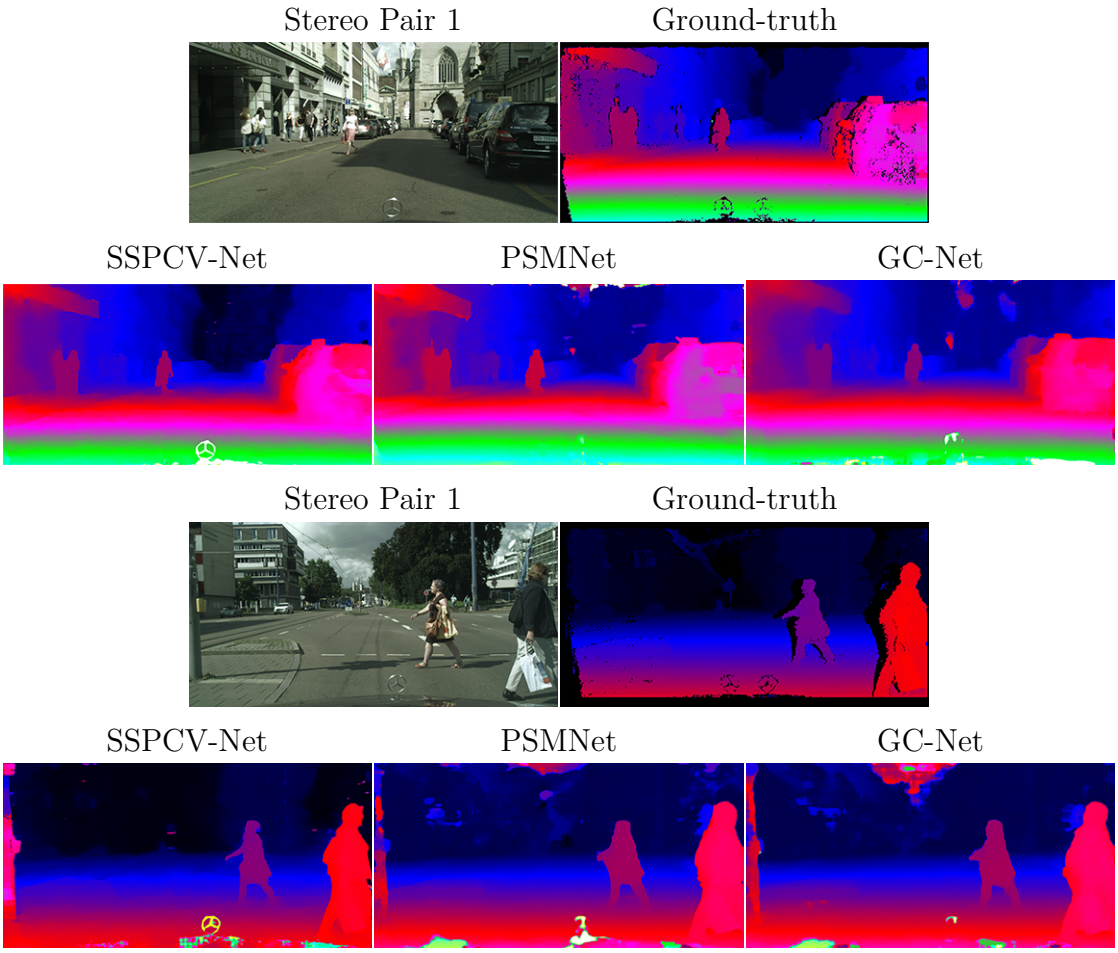

Figure 4.12 Two testing results from Cityscapes dataset by SSPCV-Net, PSMNet and GC-Net on the generalization ability.

cues. Finally, SSPCV-Net possesses all advantages of the semantic cost volume and spatial pyramid cost volumes.

4.4 Chapter Summary

In this work, we developed a new semantic stereo network of SSPCV-Net, in which pyramid cost volumes are constructed for describing semantic and spatial information on multiple levels and a 3D multi-cost aggregation module is proposed to integrate the extracted multilevel features. Comprehensive experiments on Scene Flow, KITTI

Figure 4.13 Two 3d visualizations on the Cityscapes dataset. The 3d point clouds are reconstructed using the original image and the estimated depth.

2015 and 2012 datasets demonstrated that the proposed SSPCV-Net can significantly improve the accuracy of stereo matching over many existing state-of-the-art neural networks.

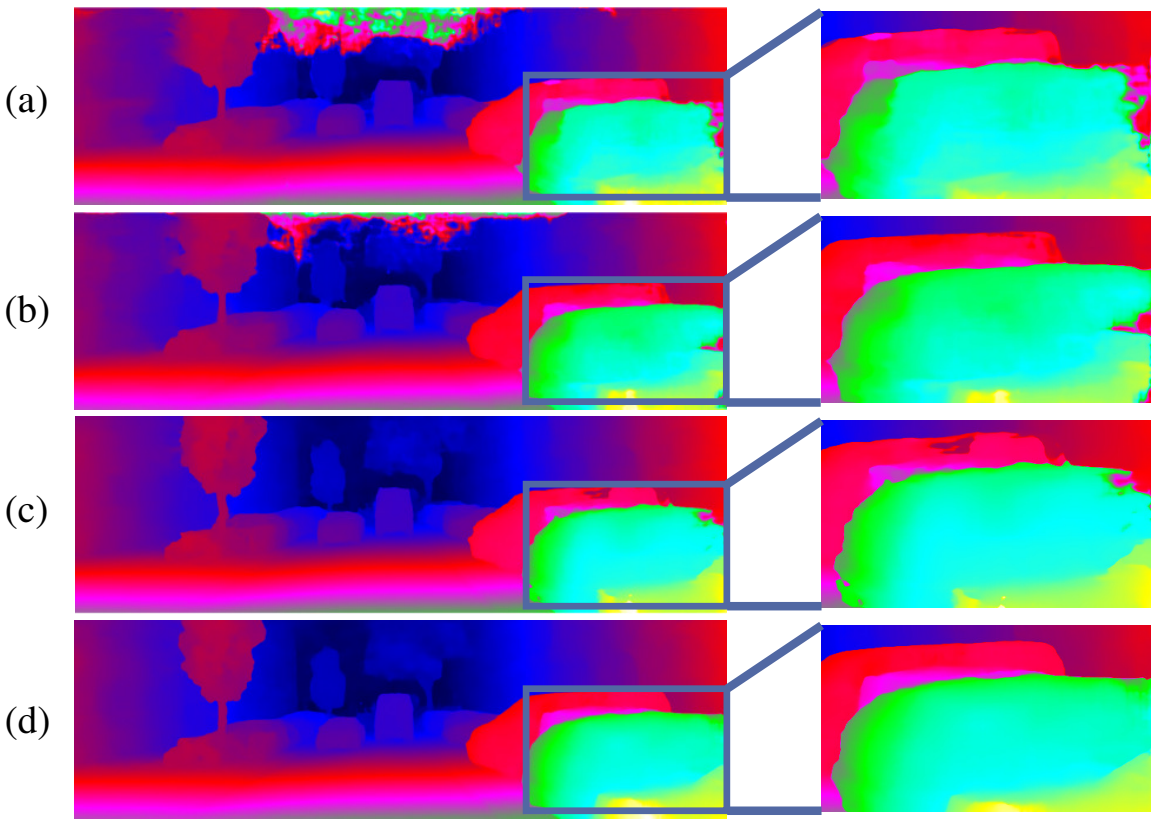

Figure 4.14 Disparity maps resulting from SSPCV-Net by excluding certain cost volume of different branches or levels.

Chapter 5

Spatial Correspondence with Generative Adversarial Network: Learning Depth from Monocular Videos

5.1 Motivation

Depth estimation from 2D images or videos is critical for many computer-vision applications, including robotics [7], autonomous driving [15, 36], 3D reconstruction [154] and augmented realities [18].

As in many other computer vision tasks, convolutional neural networks (CNNs) have been widely applied to depth estimation with significant success in recent years, such as estimating depth from single images [31, 32, 33, 86, 2, 153], stereo images [14, 71], multi-view images [61, 127, 131, 144] and monocular videos [69, 162, 161, 122, 90, 148]. Among them, depth estimation from monocular videos has drawn more and more interests in recent years since 1) it only requires one monocular camera as in many real scenarios, and 2) spatial relations between adjacent frames provide important information for depth estimation. The goal of our work focuses on developing new CNN models for depth estimation from monocular videos.

Different from stereo matching where the input pair of stereo images are taken by two cameras with a fixed relative pose, the camera pose change between adjacent frames in videos are time-varying and unknown priorly, which makes depth estimation from monocular videos a very challenging problem. Most of the available methods address this problem by first estimating the camera pose and pose change over time, usually by training respective CNNs [108, 130, 162].

There are also many works learning depth from multi-view stereo [61, 127, 131, 144]. These methods can be applied to estimate depth maps from monocular videos if treating multiple adjacent frames as multi-view images. However, many of these methods require camera poses to be given [62, 64, 144] and others need to estimate camera poses [127, 61] just like the video-based depth estimation methods mentioned above. For these methods, errors in camera-pose estimation can significantly affect the accuracy of final depth estimation [131].

In this work, we develop a novel network of SC-GAN (Spatial Correspondence

with Generative Adversarial Network) to exploit latent information between adjacent frames of a monocular video and estimate the depth by supervised training. We first propose a spatial correspondence (SC) module to match the features between adjacent frames. Inspired by simple observation that spatial features along different directions make different amounts of contributions in estimating the depth map, we introduce a direction-based attention (DBA) mechanism in the SC module to learn the importance of features along different directions. One key issue in building the feature relations between two frames lies in the computational and memory complexity. With large camera-pose change between frames and high image resolution, both of which are common in autonomous driving and virtual reality, the search space of corresponding features between two frames is very large. To address this issue, we down-sample patches of interest in adjacent frames using the Smolyak sparse grid method [117], which brings us both efficiency and accuracy in building cross-frame spatial relations.

Recently, Generative Adversarial Network (GAN) [43] which has drawn broad attention in style transfer [29, 68], image-to-image translation [63, 164], image editing [163] and cross-domain image generation [9, 28]. In [101], a GAN network was proposed to refine the estimated disparity map in image-based stereo matching. In [19, 25, 1], the classical GAN was adapted to estimate the depth from a single image. In [100], the cycled generative networks are deployed to estimate depth from stereo pair in an unsupervised manner.

In this research, we employ an end-to-end adversarial training for SC-GAN, where the generator learns to estimate depth from the input frames, while the discriminator learns to distinguish between the ground-truth and estimated depth maps for the reference frame. In the experiments, we carry out a series of ablation and comparison studies on the KITTI and Cityscapes benchmark datasets, and find that the proposed SC-GAN can achieve significantly better performance than many existing state-of-the-art monocular depth estimation methods.

5.2 Method

The proposed SC-GAN network consists of a generator and a discriminator that compete against each other. Figure 5.1 presents the detailed architecture of SC-GAN. The inputs are triple adjacent frames in a video – frames -1, 0, and 1. Among them, frame 0 is the reference frame from which we seek to estimate the depth map. While this architecture can be extended to consider more or less adjacent frames, in this work we focus on triple-frame inputs for simplicity.

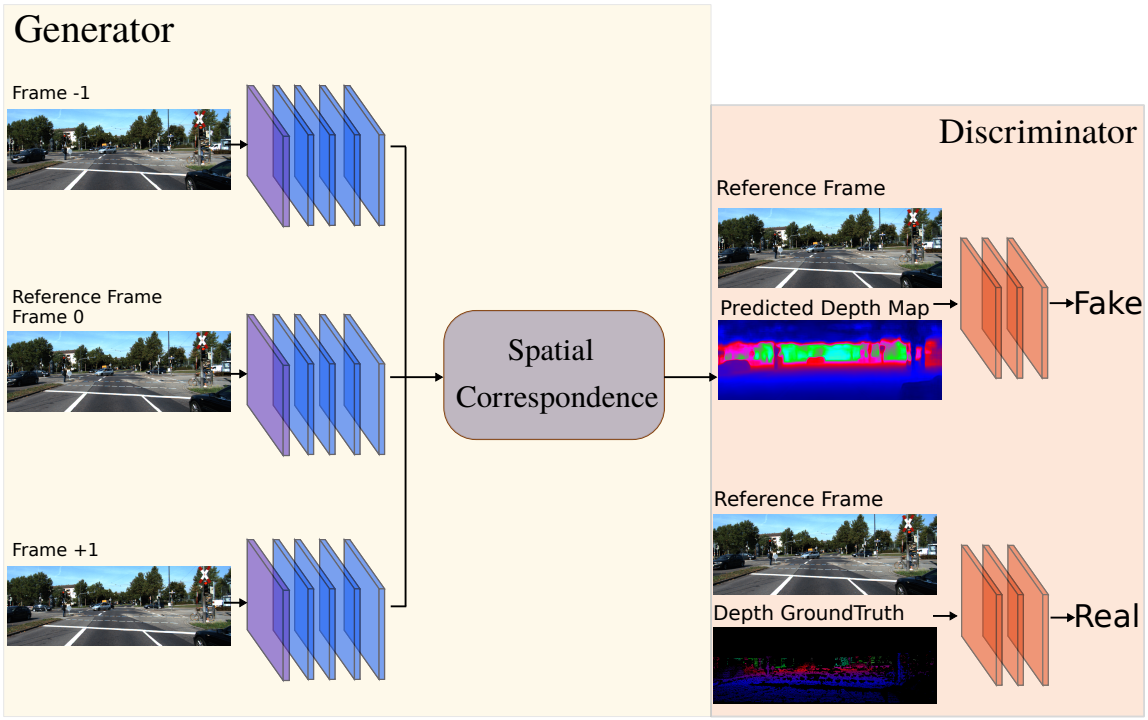

Figure 5.1 Architecture of the proposed SC-GAN consisting of a generator and a discriminator.

5.2.1 Network architecture

The generator network of SC-GAN, as as shown in Figure 5.2 includes a spatial correspondence module, a direction-based attention mechanism and a depth map

refinement module, which takes the group of triple adjacent frames as input and outputs the depth map in an end-to-end manner.

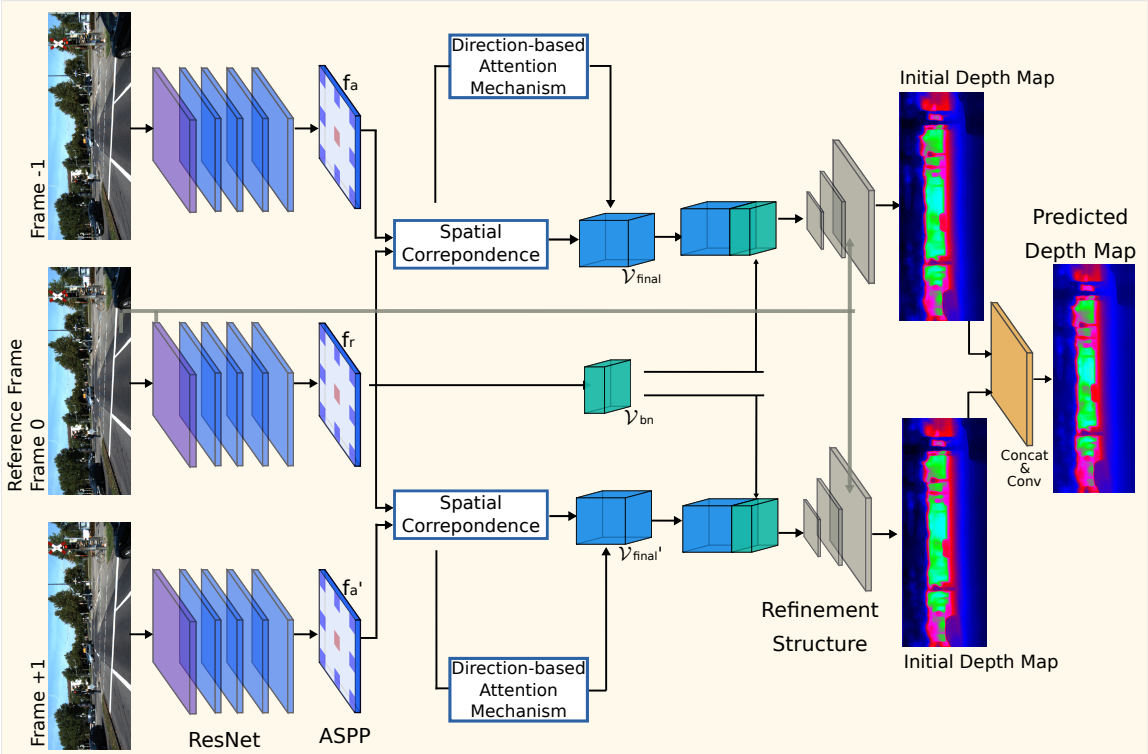

Figure 5.2 Architecture of the generator network of SC-GAN.

Firstly, it begins by using ResNet-50 [51] to extract features from the input frames (all with the same size $W \times H$) and then for each frame the atrous spatial pyramid pooling (ASPP) [17] module is employed to extract features from multiple large receptive fields via dilated convolutional operations with dilation rates $(6, 12, 18)$. The output feature map for each frame is then a $\frac{W}{4} \times \frac{H}{4} \times \tilde{C}$ tensor where $\tilde{C}$ represents the number of channels. Note that ResNet-50 and the ASPP module are weight sharing among all three branches. Secondly, the spatial correspondence module combined with the direction-based attention mechanism is used to form correlation features for each pair of the reference frame and one of its adjacent frames. Thirdly, the correlation features and the batch normalized features of the reference frame are fed

together into the weight-sharing refinement subnetwork to get the respective initial depth map for each pair. Finally, the concatenation of all initial depth maps from all pairs is the input of a 3×3 convolution layer to predict the final depth map of the reference frame.

During the training phase, we use the Markovian discriminator (PatchGAN [63]) which consists of 4×4 Convolution-InstanceNorm-LeakyReLU layers. The discriminator is used to distinguish the pair of the predicted depth map and the reference frame with the pair of the ground-truth depth map and the reference frame, and then to provide feedback to the generator.

5.2.2 Spatial correspondence

Inspired by Flownet [30] which introduces a “correlation layer" that performs multiplicative patch comparisons between two feature maps, we propose a spatial correspondence module to match the features. An illustration of the spatial correspondence is shown in Figure 5.3.

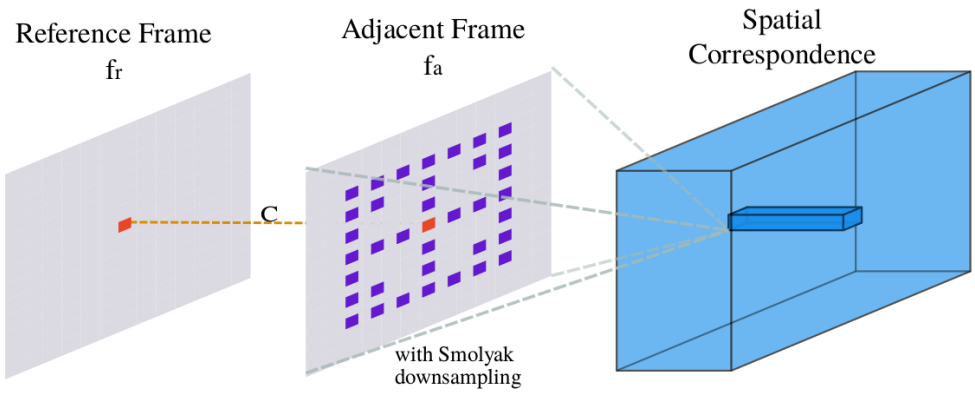

Figure 5.3 Architecture of the spatial correspondence module. The grey squares represent the feature maps from the reference frame and one of its adjacent frames, respectively, and the volume on the right indicates the obtained correlation features V defined in Eq. (5.1).

Let k be the maximum displacement when corresponding the features between the

feature map f_r of the reference frame and feature map f_a of one of its adjacent frames. For the feature at each position (i, j) of f_r, the search space for its corresponding feature position in f_a is a $(2k + 1) \times (2k + 1)$ square patch centered at (i, j). One common way is to define their spatial correlation features as a $\frac{W}{4} \times \frac{H}{4} \times C$ tensor V with $C = (2k + 1)^2$ and each of its entry is given by

$$V(i, j, c) = \mathcal{C}(f_r(i, j), f_a(i + o_1, j + o_2)) \tag{5.1}$$

for $o_1, o_2 \in [-k, k]$ where $c = (o_1 + k)(2k + 1) + (o_2 + k)$ and the function $\mathcal{C}$ denotes a 1×1 convolution (equivalent to the dot-product operation in this case). Thus the spatial correlation features V outputs C values for each position (i, j) of the reference f_r, which could be quite large when the maximum displacement k is large, especially for high-resolution input images.

A typical way to reduce the search space is to down-sample the $(2k+1) \times (2k+1)$ square patch and only search for the corresponding features from a sparse set of positions. Uniform sampling is certainly a choice – as shown in Figure 5.4-(a), for a given sampling rate r, the third dimension (i.e., the number of channels) of the correlation feature tensor V can be reduced to $C = \lceil \frac{2k+1}{r} \rceil^2$ by uniform sampling. In this work, we propose to use Smolyak sparse grids [117] for non-uniform sampling.

The sparse grid technique is an effective numerical method with high computational efficiency for representation, interpolation and integration of functions in multi-dimensional spaces, and was first proposed in [117] by Smolyak based on sparse tensor products. Since then it has been widely used in approximation theory [4], uncertainty quantification and high-dimensional integrations [11], global optimization [96], data compression [38] and etc.

The set of Smolyak sparse grids $\mathcal{S}$ in a square domain in two dimensions is defined as

$$\mathcal{S}_l = \bigcup_{\alpha_1 + \alpha_2 \leqslant l} (\Theta^x_{\alpha_1} \bigotimes \Theta^y_{\alpha_2}), \tag{5.2}$$

where l denotes the level of the sparse grids, α_1 and α_2 are nonnegative integers, and Θ_α^j is the one-dimensional interpolation abscissas, which can be $2^\alpha + 1$ uniformly distributed points (uniform-type) or the roots (need to be accordingly scaled by the domain size) of the $(2^\alpha - 1)$-order Chebyshev polynomial $\{\cos(\frac{(n-1/2)\pi}{2^\alpha - 1})\}_{n=1}^{2^\alpha - 1}$ plus two end points (Chebyshev-type).

Since the distribution of Smolyak sparse grid points (especially Chebyshev-type) is highly non-uniform, we project all grid points to their nearest integer-coordinate points in the frame then remove all duplicated ones in order to avoid extra interpolations cost in practice. Such set of approximate Smolyak sparse grid-points at each level l is denoted as $\tilde{\mathcal{S}}_l$. Consequently, the third dimension of the feature tensor V becomes

$$C = |\tilde{\mathcal{S}}_l| \tag{5.3}$$

with the use of $\tilde{\mathcal{S}}_l$ as the sampling points, which can be much smaller than $(2k+1)^2$ and significantly reduces the amount of calculations while still maintaining good approximation accuracies of the correlation information. Sampled points by using two types of Smolyak sparse grids at different levels on a 49×49 square patch are illustrated in Figure 5.4-(b) and (c), respectively.

In the later experiment, we will conduct ablation studies to compare the performance of using uniform down-sampling, and Smolyak sparse grids of uniform-type and Smolyak sparse grids of Chebyshev-type for down-sampling.

5.2.3 Direction-based attention mechanism

To enable the spatial correspondence module to selectively leverage the spatial correlation features aggregated along different directions, we propose a direction-based attention mechanism (DBA), inspired by the DSC method [60] and the Squeeze-and-Excitation Blocks [59]. An illustration of the DBA mechanism is shown in Figure 5.5, which consists of an adaptive average pooling layer, two fully connected (FC) layers,

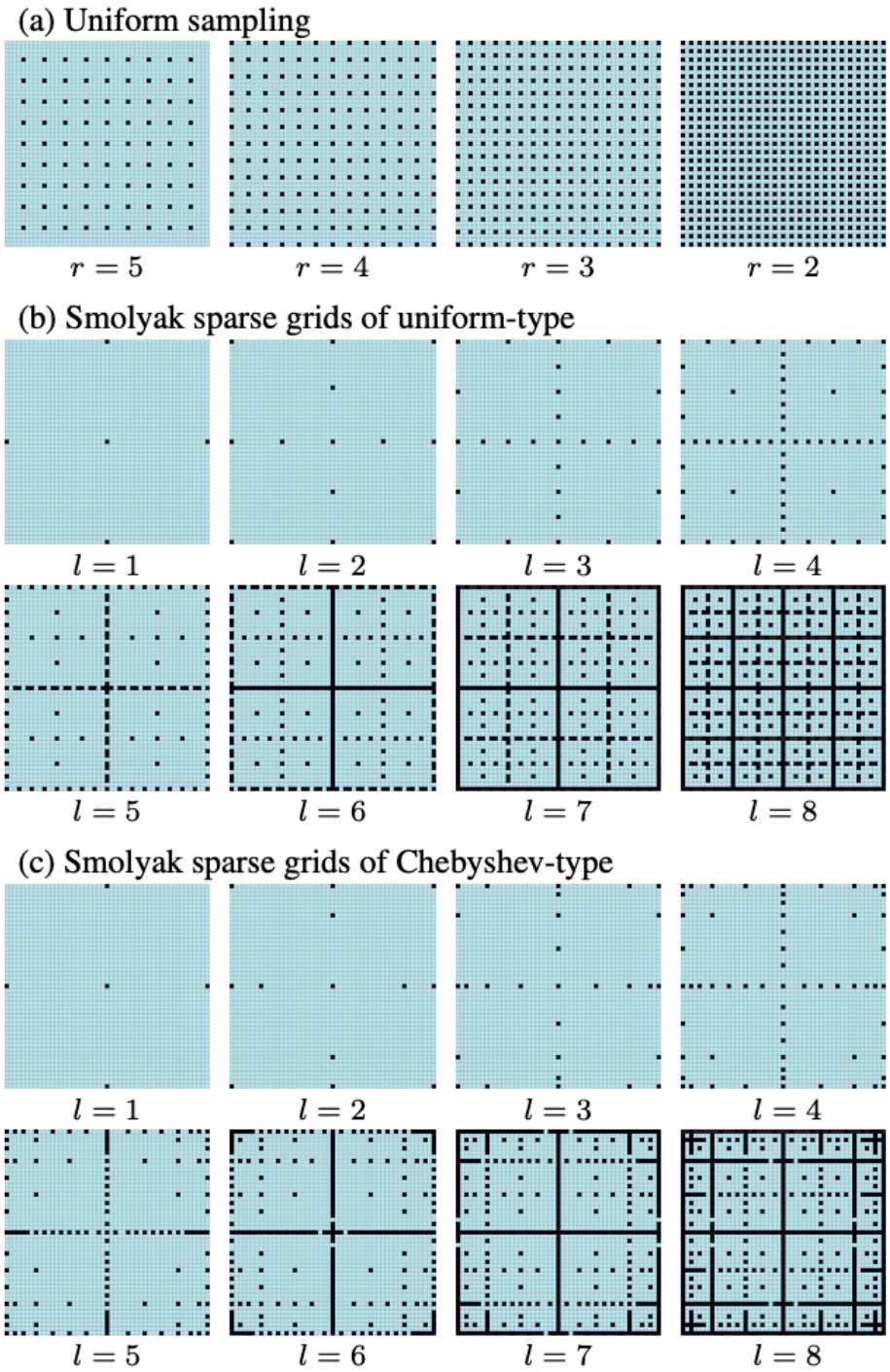

Figure 5.4 Sampled points by using different down-sampling methods on a 49×49 square patch. (a) Uniform samplings; (b) Smolyak sparse grids of uniform-type; (c) Smolyak sparse grids of Chebyshev-type.

and a ReLU layer and generates a direction-wise attention vector for each pair of feature maps (f_r and f_a) of the reference frame and one of its adjacent frames.

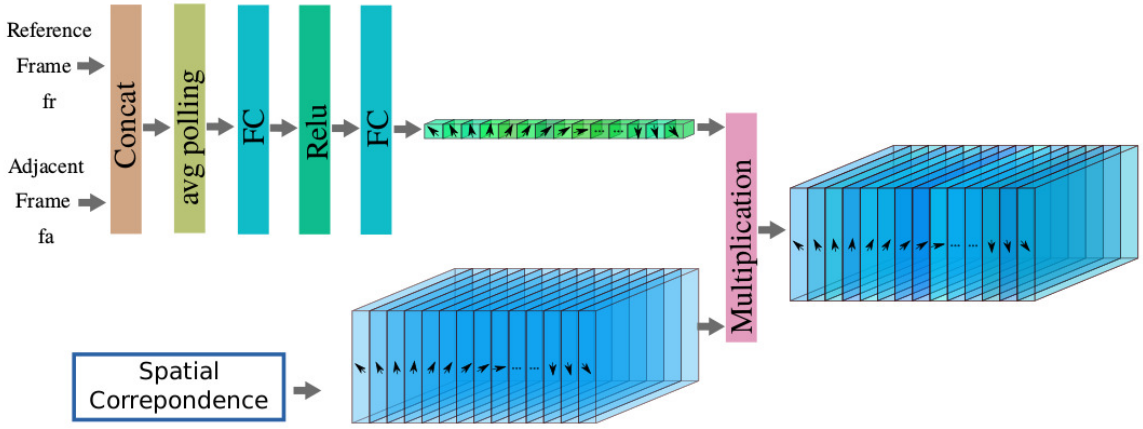

Figure 5.5 Architecture of the direction-based attention (DBA) mechanism.

The DBA mechanism starts from computing a vector $\boldsymbol{w} \in \mathbb{R}^{2\tilde{C}}$ from features of the reference and adjacent frames:

$$\boldsymbol{w} = \frac{16}{H \times W} \sum_{i=1}^{H/4} \sum_{j=1}^{W/4} \langle f_r(i,j), f_a(i,j) \rangle, \tag{5.4}$$

where $\langle \cdot, \cdot \rangle$ denotes the concatenation. Then the direction-based vector $\boldsymbol{w}_{DBA} \in \mathbb{R}^C$ is defined as:

$$\boldsymbol{w}_{DBA} = \boldsymbol{W}_2 \cdot \mathcal{R}(\boldsymbol{W}_1 \cdot \boldsymbol{w}), \tag{5.5}$$

where $\mathcal{R}$ stands for the ReLU function and $\boldsymbol{W}_1 \in \mathbb{R}^{C \times 2\tilde{C}}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{C \times C}$ are the weight matrices for two fully connected layers. The final direction-based correlation feature tensor V_{final} is then defined by

$$V_{final}(i,j,c) = V(i,j,c) \cdot \boldsymbol{w}_{DBA}(c), \tag{5.6}$$

for $c = 0, 1 \cdots, C-1$.

Note that, our DBA mechanism is different from the prior DSC method [60] and the Squeeze-and-Excitation Blocks [59]. By including the pooling layer, the DBA mechanism uses global information to learn the weight of all possible directions, while the DSC method [60]) uses local information only. The Squeeze-and-Excitation Blocks [59] learns the inter-dependencies between channels from a single image, while the proposed DBA mechanism learns a vector to represent the weight to reflect the importance of each of the directions.

5.2.4 Depth map refinement

In order to provide dense depth predictions with high resolution, we build a weight-sharing refinement subnetwork to refine the predicted depth map for each group of input frames. The input of the refinement subnetwork is the concatenation of V_{final} and V_{bn}, where V_{final} is the final correlation feature map defined in Eq. (5.6) and V_{bn} is the result of batch normalization on f_r.

The refinement subnetwork is depicted in Figure 5.6, which includes a series of deconvolution, concatenation, and convolution operations, as in [152, 30]. Following these operations, we can leverage both high-level and low-level information from three parts including the features generated from the ASPP module, features obtained after Conv0 layer in ResNet, and the original reference frame. The kernel size in convolution blocks is 3×3 and each deconvolution layer doubles the resolution of the result. And finally, obtain an initial estimate of the depth map with the same resolution as the original frame.

5.2.5 Loss function

SC-GAN contains a generator G to estimate the depth map $G(R)$ for the reference frame R as described above, and a discriminator D to distinguish the ground-truth depth map M_d and the predicted depth map $G(R)$ of the reference frame. Follow-

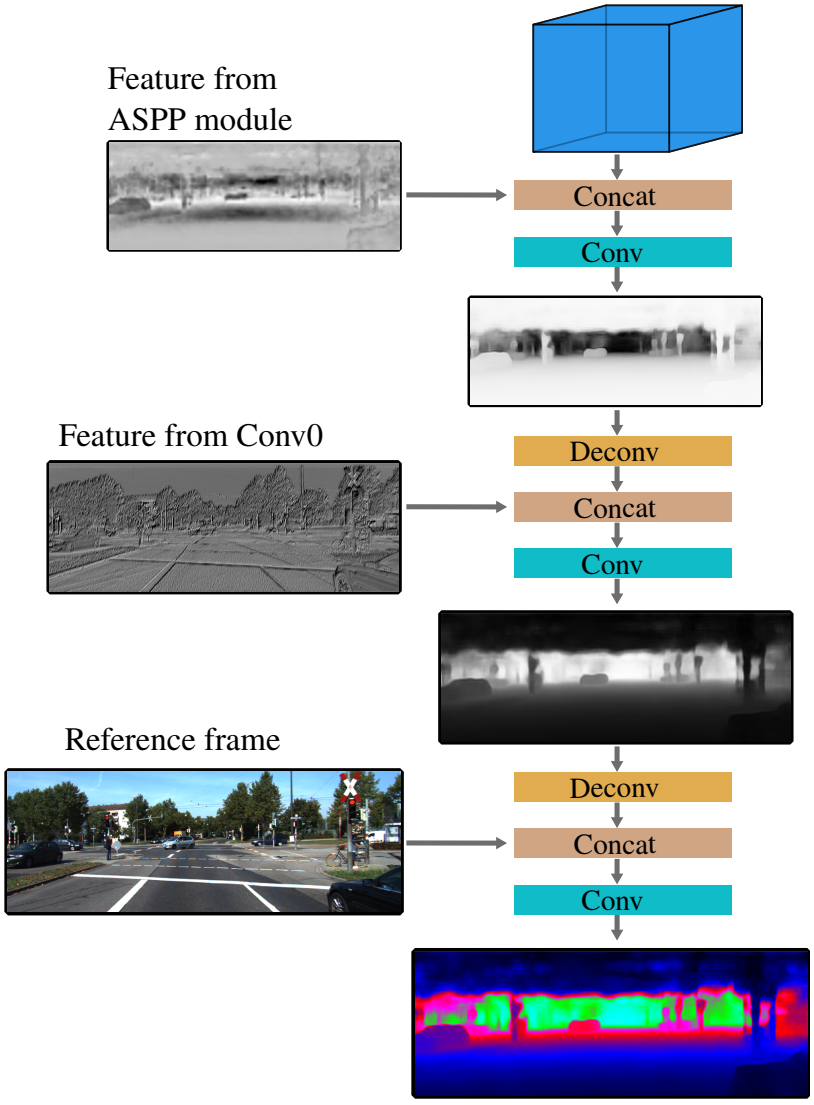

Figure 5.6 Architecture of the refinement subnetwork.

ing [63, 19], SC-GAN is trained with a per-pixel loss term $\mathcal{L}_{L1}$ and an adversarial loss term $\min_G \max_D \mathcal{L}_{GAN}$:

$$\mathcal{L} = \mathcal{L}_{L1} + \lambda \min_G \max_D \mathcal{L}_{GAN}, \tag{5.7}$$

where λ is the balancing factor. Since the ground-truth depth map is usually sparse, we define φ as the mask operation converting the estimated depth maps to the corresponding sparse ones.

The per-pixel loss term $\mathcal{L}_{L1}$ is defined as:

$$\mathcal{L}_{L1} = \mathbb{E}_{M_d, G(R)}[\|M_d - \varphi(G(R))\|_1]. \tag{5.8}$$

The adversarial loss is expressed as:

$$\begin{aligned} \min_G \max_D \mathcal{L}_{GAN}(G, D) = \mathbb{E}_{R, M_d}[\log D(R, M_d)] + \\ \mathbb{E}_{R, G(R)}[\log(1 - D(R, \varphi(G(R))))], \end{aligned} \tag{5.9}$$

where G tries to minimize this loss against an adversarial D that tries to maximize it. The purpose of using the adversarial loss is to classify the overlapping pairs of the reference frame and the depth patches as being real or fake.

5.3 Experiment

5.3.1 Datasets and evaluation metrics

The following two datasets are used for performance evaluation and comparisons of the proposed SC-GAN with many existing state-of-the-art networks on depth estimation from monocular videos.

KITTI [37]: The KITTI dataset is the most commonly used benchmark in prior works for evaluating the depth, disparity and ego-motion accuracy [35, 162], which includes a full suite of data sources such as stereo videos and sparse depth maps from LIDAR. For our experiments, we only use the monocular video streams and the corresponding sparse depth maps for training and the reference frames in the test split are the same as the KITTI Raw Eigen test split [32, 162].

Cityscapes [24]: The Cityscapes dataset consists of a large set of stereo video sequences recorded in streets from 50 different cities with ground-truth disparities. Due to the focus on monocular videos, we choose the image sequences, each of which is 30-frame snippet (17Hz) around a left 8-bit image from the train, validation, and test sets (150,000 images). For each sequence, we take its left 8-bit image as the

reference frame, together with its adjacent two frames, as input to the network. The ground-truth depth map for each reference frame is inferred from its disparities.

Our evaluations are based on several metrics from prior work [32] – absolute relative difference (Abs Rel):

$$\text{AbsRel} = \frac{1}{|T|}\sum_{y\in T}\frac{|y - y*|}{y^*} \tag{5.10}$$

where y is the predicted depth map, and y^* is the ground truth. squared relative difference (Sq Rel):

$$\text{SqRel} = \frac{1}{|T|}\sum_{y\in T}\frac{(y - y*)^2}{y^*} \tag{5.11}$$

root-mean-square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{|T|}\sum_{y\in T}(y - y^*)^2} \tag{5.12}$$

log RMSE (RMSE log)

$$\text{RMSE log} = \sqrt{\frac{1}{|T|}\sum_{y\in T}(\log y - \log y^*)^2}. \tag{5.13}$$

For these error metrics, the lower the better.

We also use the accuracy metrics: the accuracy within different thresholds $\delta = \{1.25, 1.25^2, 1.25^3\}$.

$$\text{accuracy} = \text{the ratio of pixels } i \text{ satisfy } \left(\max(\frac{y}{y^*}, \frac{y^*}{y}) < \delta\right) \tag{5.14}$$

For the accuracy metrics, the higher the better.

5.3.2 Model specification

The proposed SC-GAN was implemented based on PyTorch, and all trainings were done on two Nvidia 1080 GPUs with the minibatch SGD and the Adam solver (the momentum parameters $\beta_1 = 0.5, \beta_2 = 0.999$). Following the standard approach from [63], we performed one gradient descent step on discriminator, then one step on

generator. We trained our model from scratch using the training dataset with the batch size of 1, and kept the same learning rate ($lr = 0.0002$) for both generator and discriminator.

We performed color normalization on the entire dataset for data preprocessing, and during the training process, all images were randomly cropped to the size of 256×512 and augmented with random (flip and color) transformations as done in [32]. We set the number of channel $\tilde{C} = 256$ for feature extraction, then each of the resulting feature maps was a tensor of size $64 \times 128 \times 256$. The maximum displacement k was set to be 24 in the spatial correspondence module and the maximum depth to be 80 for both datasets. For all experiments, the balance factor $\lambda = 0.1$ was used in Eq. (5.7). We trained SC-GAN on the KITTI dataset with 10 epochs while on the Cityscapes dataset with 12 epochs.

5.3.3 Comparisons with existing networks

Firstly, we evaluate and compare the performance of SC-GAN with eleven existing state-of-the-art networks [32, 86, 33, 49, 41, 76, 48, 162, 148, 143, 122] for depth estimation on the KITTI dataset. All these networks are trained on the KITTI dataset and Table 5.1 reports the evaluation results. It is easy to see that SC-GAN achieves the best performance (with significant better results) under all error and accuracy metrics. Figure 5.7 presents three examples of the depth maps estimated by SC-GAN and DORN [33].

We also evaluate the *generalization ability* of SC-GAN. In this case, SC-GAN is trained only on the Cityscapes dataset and then tested on the KITTI dataset. Table 5.2 reports the corresponding performance evaluation results, from which we see that SC-GAN again significantly outperforms the three comparison methods [32, 41, 13]. The results of this generalization test for other comparison methods are not available in literature. We also note that when testing on the KITTI dataset, SC-GAN trained

Table 5.1 Performance comparison of SC-GAN and some existing state-of-the-art networks on KITTI. Note that the $\star$ marks the method is in the semi-supervised or unsupervised manner.

Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al.* [32]	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al.* [86]	0.202	1.614	6.523	0.275	0.678	0.895	0.965
DORN [33]	0.072	0.307	2.727	0.120	0.932	0.984	0.994
DfUSMC [49]	0.346	5.984	8.879	0.454	0.617	0.796	0.874
Godard *et al.* [41]	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Kuznietsov *et al.* [76]	0.113	0.741	4.621	0.189	0.862	0.960	0.986
Guo *et al.* [48]	0.097	0.653	4.170	0.170	0.889	0.967	0.986
Zhou *et al.* [162]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Yin *et al.* [148]	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Yang *et al.* [143]	0.097	0.734	4.442	0.187	0.888	0.958	0.980
Teed *et al.* [122]	0.091	0.582	3.644	0.154	0.923	0.970	0.987
SC-GAN	0.063	0.178	2.129	0.097	0.961	0.993	0.998

on Cityspaces can even get comparable performance to several supervised and semi-supervised models [41, 148, 76] that are trained on KITTI (see Table 5.1 and Table 5.2). Figure 5.8 and 5.9 visually illustrates the estimated depth maps for three examples produced by SC-GAN that are trained on different datasets. All these results clearly demonstrate the excellent generalization ability of the proposed SC-GAN.

Table 5.2 Performance comparison of SC-GAN and some state-of-the-art networks on KITTI, when trained on Cityscapes and then tested on KITTI.

Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al.* [32]	0.423	4.373	8.487	0.356	0.655	0.871	0.951
Godard *et al.* [41]	0.233	3.533	7.412	0.292	0.700	0.892	0.953
Caser *et al.* [13]	0.153	1.109	5.557	0.227	0.796	0.934	0.975
SC-GAN	0.149	0.921	4.812	0.192	0.818	0.954	0.987

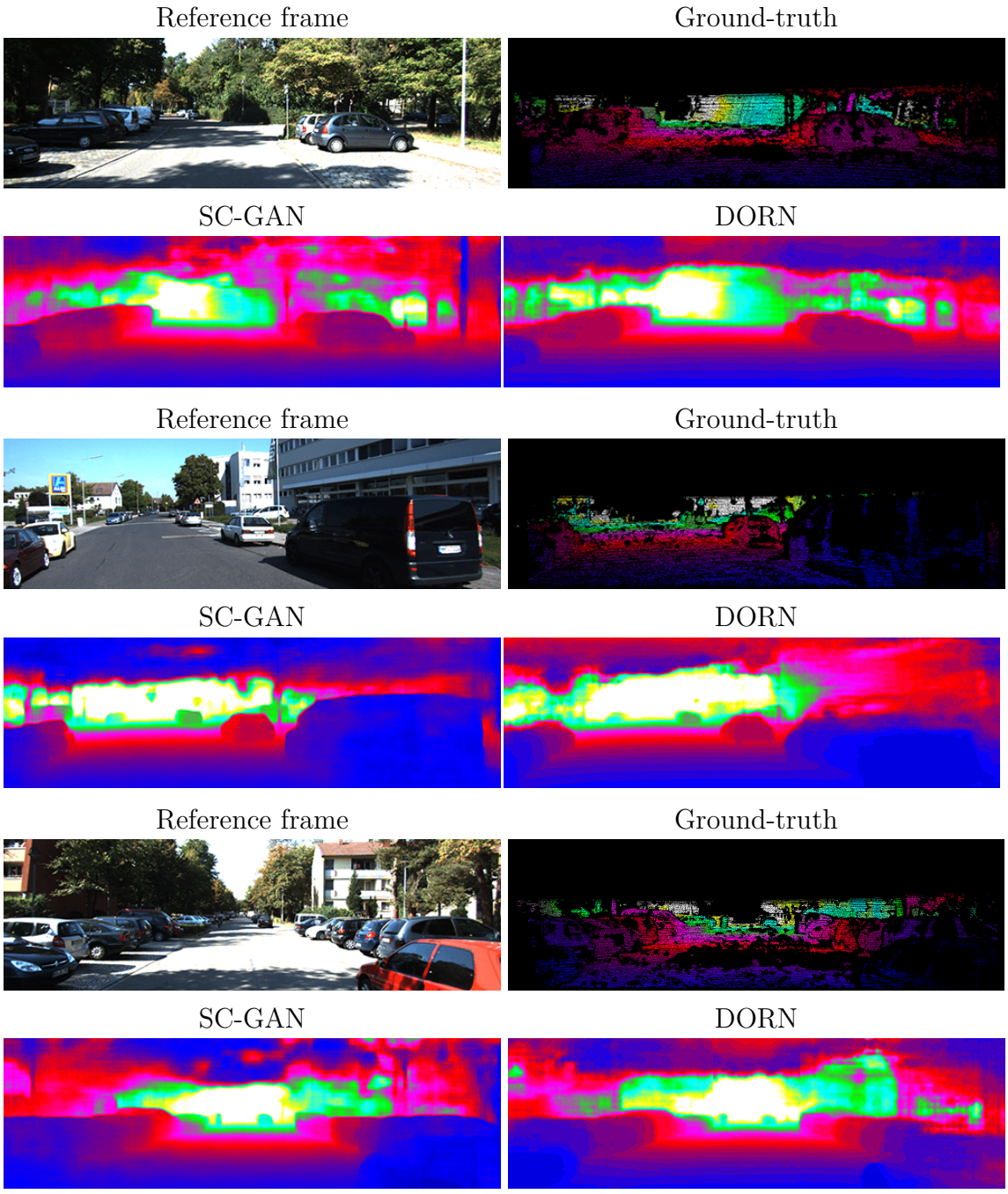

Figure 5.7 Visualization of three testing results from KITTI. From left to right: the reference frame, the ground-truth depth map, the predicted depth map by SC-GAN and the predicted depth map by DORN.

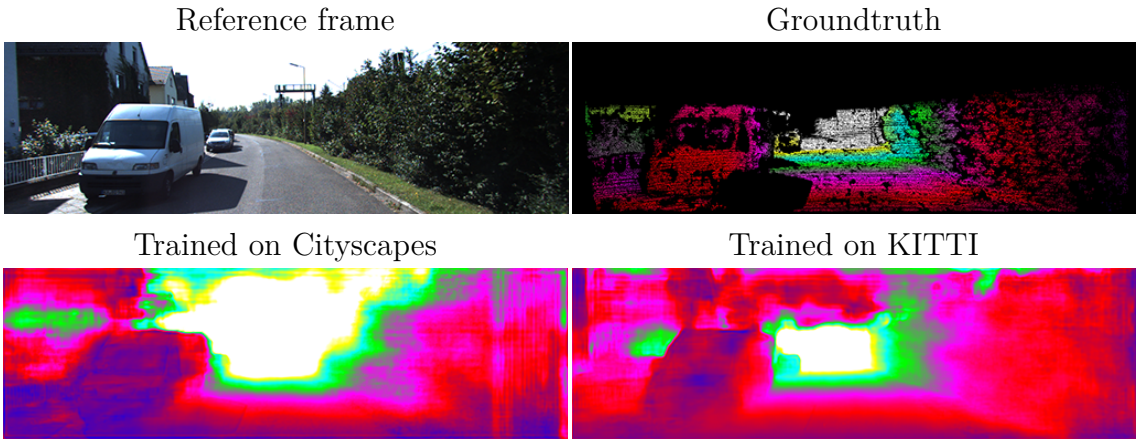

Figure 5.8 One sample result from KITTI in testing the generalization ability of SC-GAN. It shows the reference frame, the ground-truth depth map, the depth map predicted by SC-GAN trained on Cityscapes, the depth map predicted by SC-GAN trained on KITTI.

5.3.4 Ablation studies

We first carry out a study to select the down-sampling method and parameters for feature correspondence in the proposed SC-GAN. Without down-sampling, we need search all $C = 49 \times 49 = 2,401$ points in the patch. The uniform sampling with different rates ($r = 4, 3, 2$) and the approximated uniform-type and Chebyshev-type Smolyak sparse grids $\tilde{\mathcal{S}}_l$ at different levels ($l = 5, 6, 7, 8$) are tested and compared on the KITTI dataset.

The results in Table 5.3 show that: 1) compared to uniform sampling, the Smolyak sparse grids tend to be more efficient and effective for spatial correlation when C becomes larger since uniform sampling may contain a lot of redundant information; 2) by seeking a compromise between the time efficiency (and memory cost) and the depth estimation accuracy, we choose the Chebyshev-type Smolyak sparse grid at level 7, which clearly beats the other two down-sampling methods with similar numbers of sampled points, i.e., the uniform sampling with $r = 2$ and the uniform-type Smolyak

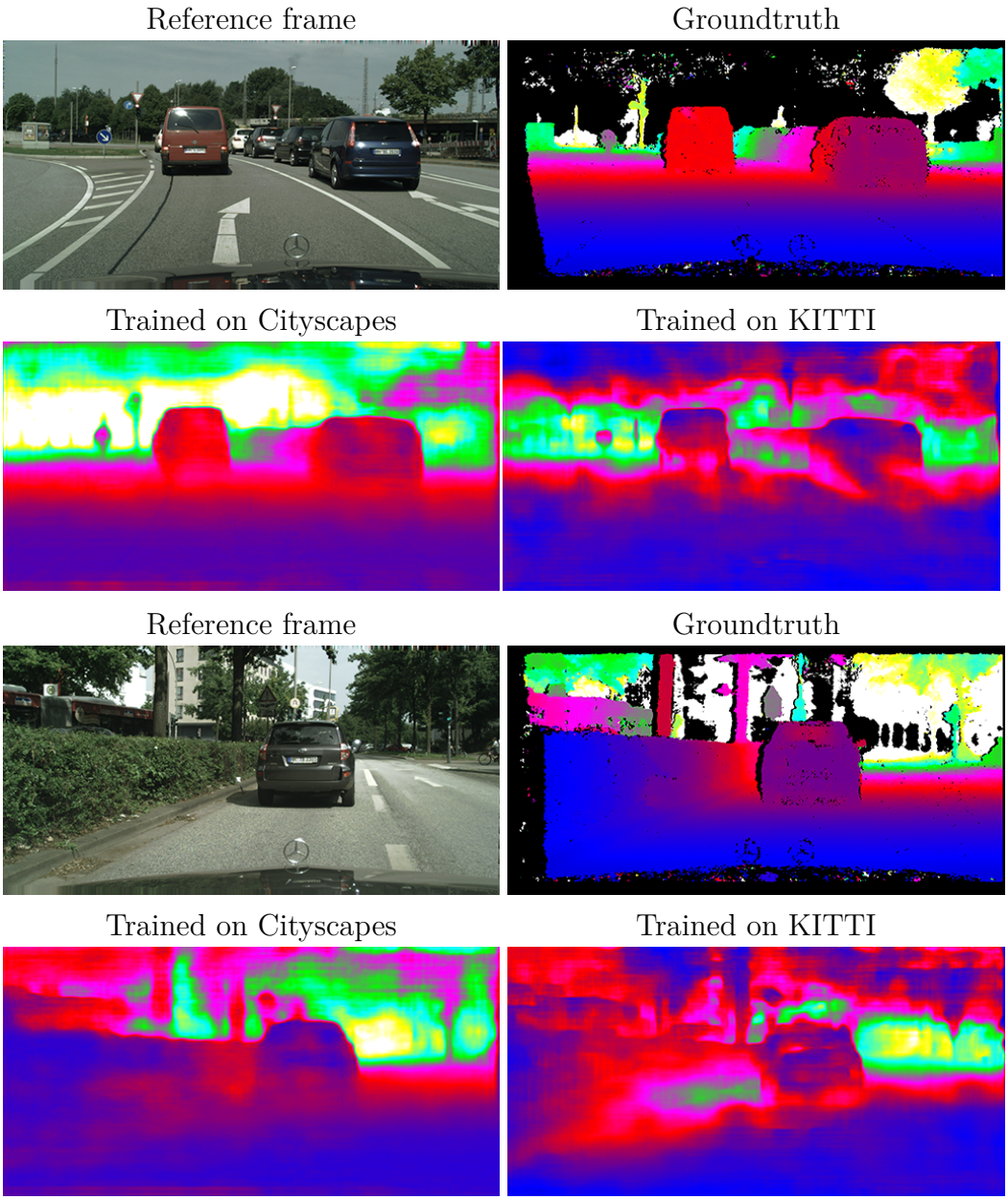

Figure 5.9 Two sample results from Cityscapes in testing the generalization ability of SC-GAN. It shows the reference frame, the ground-truth depth map, the depth map predicted by SC-GAN trained on Cityscapes, the depth map predicted by SC-GAN trained on KITTI.

sparse grid at level 7.

In all the remaining experiments, the Chebyshev-type $\tilde{\mathcal{S}}_7$ (441 points, about 18.37% of the original 2,401 points) is used as the down-sampling method in SC-

Table 5.3 Ablation study for selecting the down-sampling method for feature correspondence in SC-GAN on KITTI. The rightmost column is the computation time (measured in seconds) for processing of one group of triple frames in training.

Method	Level	C	Abs Rel	Sq Rel	RMSE	Time
Uniform sampling	$r = 4$	169	0.071	0.227	2.444	0.24
	$r = 3$	289	0.068	0.207	2.263	0.26
	$r = 2$	625	0.064	0.181	2.116	0.39
Smolyak sparse grids-Uniform	$l = 5$	145	0.073	0.228	2.316	0.23
	$l = 6$	289	0.069	0.209	2.312	0.26
	$l = 7$	481	0.066	0.201	2.270	0.32
	$l = 8$	737	0.063	0.181	2.084	0.41
Smolyak sparse grids-Chebyshev	$l = 5$	129	0.074	0.220	2.319	0.23
	$l = 6$	261	0.069	0.212	2.464	0.26
	$l = 7$	441	0.063	0.178	2.129	0.32
	$l = 8$	669	0.062	0.174	2.082	0.39

GAN. In this case, it took about 50 hours of training for SC-GAN on the KITTI dataset and 40 hours on the Cityscapes dataset.

Next, we conduct ablation study to justify different modules of SC-GAN on the KITTI dataset, including 1) the spatial correspondence module; 2) the direction-based attention mechanism; 3) the refinement subnetwork; and 4) the adversarial loss. The quantitative results are reported in Table 5.4. A sample result of depth maps are shown in Figure 5.10 (a) the reference frame; (b) the depth map estimated without the spatial correspondence module; (c) the depth map estimated without the direction-based attention module; (d) the depth map estimated without the refinement subnetwork; (e) the depth map estimated without the adversarial loss; (f) the depth map estimated by the full version of SC-GAN. We can see that all these four modules can help improve the depth estimation.

Table 5.4 Comparison of a number of different model variants of SC-GAN on KITTI: (a) the reference frame; (b) the depth map estimated without the spatial correspondence module; (c) the depth map estimated without the direction-based attention module; (d) the depth map estimated without the refinement subnetwork; (e) the depth map estimated without the adversarial loss; (f) the depth map estimated by the full version of SC-GAN.

Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
w/o Spatial correspondence	0.079	0.222	2.301	0.111	0.949	0.981	0.994
w/o DBA mechanism	0.065	0.174	2.182	0.094	0.956	0.991	0.997
w/o Refinement	0.103	0.409	3.354	0.150	0.895	0.979	0.991
w/o Adversarial loss	0.069	0.234	2.653	0.120	0.934	0.990	0.995
Full version of SC-GAN	0.063	0.178	2.129	0.097	0.961	0.993	0.998

5.4 Chapter Summary

In this work, we developed a novel end-to-end SC-GAN network for depth estimation from monocular videos. SC-GAN consists of a generator and a discriminator. In the generator, a spatial correspondence module is designed to match the features between the reference frame and its adjacent frames. We proposed to use the approximate Smolyak sparse grids for patch down-sampling that can significantly speed up the feature correspondence. We further developed a direction-based attention mechanism to learn the importance of features in different directions, and included a refinement subnetwork to refine the initially estimated depth maps. Extensive experiments on the KITTI and Cityscapes datasets demonstrate that the proposed SC-GAN significantly promotes the state-of-the-art performance of the depth estimation from monocular videos.

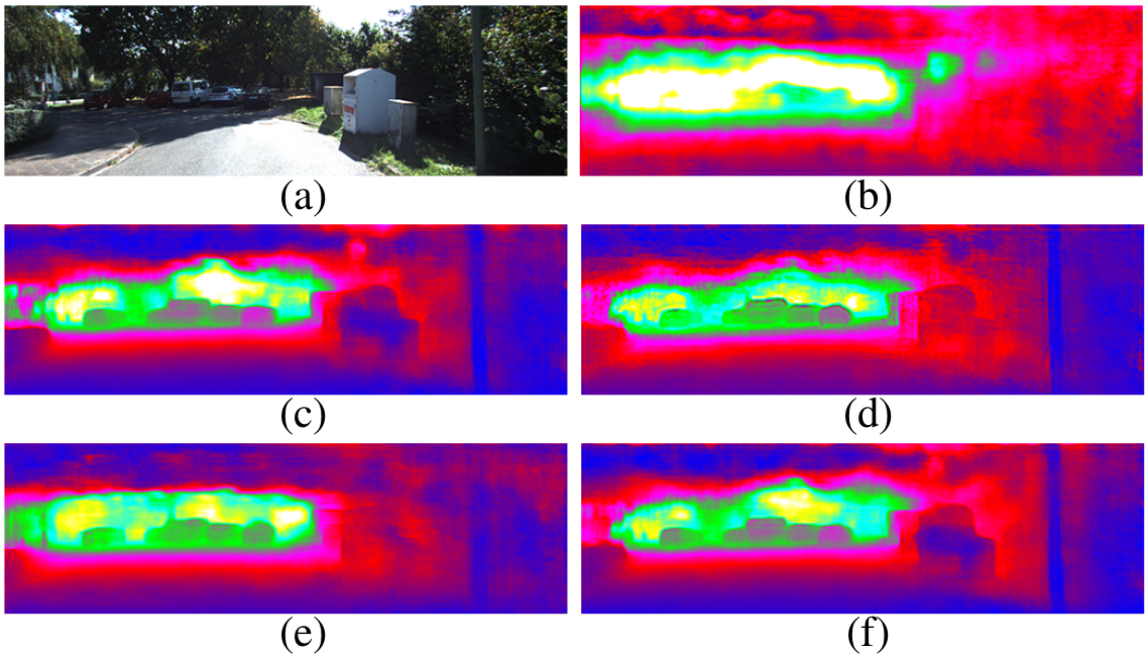

Figure 5.10 Depth maps resulting from different model variants of SC-GAN

Chapter 6

Learning Depth from Single Image using Depth-Aware Convolution and Stereo Knowledge

6.1 Motivation

Learning depth from a single image is an intriguing computer vision problem and has many important applications. Compared with depth estimation from a stereo pair of images [72, 14, 137] or video sequences [42], inferring accurate depth from single image is much more challenging without the help of multiple view information.

In early years, many approaches make use of Markov Random Fields (MRF), semantic classifiers and superpixels to tackle the single-view depth estimation task. Later, Eigen *et al.* [32] first proposed the use of a multi-scale convolutional architecture to learn depth from single image based on deep learning techniques. Following this innovative work, many more approaches based on convolutional neural networks (CNNs) [78, 12, 33, 34, 139, 145] have been proposed for monocular depth estimation. All of these methods treat the features in different depth equally using traditional convolution operations. However, these convolution operations may mix the features from different objects, which might cause inaccurate prediction of depth and abrupt depth change near the border of two adjacent objects in the image. Inspired by the work of [50] which proposes a segmentation-aware CNN by adapting its filters at each pixel based on segmentation cues, we design a novel depth-aware convolution operation for single-view depth estimation based on depth cues.

Moreover, as an ill-posed problem, single-image depth estimation still shows a very large performance gap from stereo matching. This is no strange because the former one lacks the crucial multi-view geometric information, even if the use of deep learning techniques can help infer geometric information with data-driven approaches.

In this work, we also propose to make use of the feature extracted from the stereo pair to rectify the ill-posed features extracted from a single image by using the knowledge-distillation technique [52], which was initially proposed for model compression. As shown in Figure 6.1, the main idea is to let student network learn from the teacher network, *i.e.*, the results of student network mimic the final results of the

teacher model.

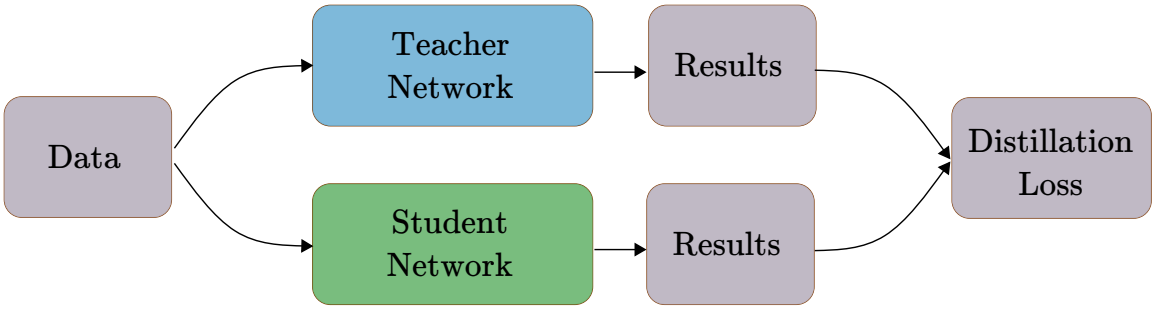

Figure 6.1 An illustration of knowledge distillation.

Knowledge distillation [52] technique is initially proposed for model compression, *i.e.*, transferring knowledge from a cumbersome model to a light-weight model. Later this approach is also taken for knowledge transfer across different domains [56]. Knowledge used for distillation and transfer can be softened labels [52, 98] or intermediate features [109, 74]. Until now, knowledge distillation has been widely applied in computer vision applications, such as object detection [79], pedestrian re-identification [20], and semantic segmentation [138, 88].

Previous works [48, 125, 99] have leveraged the distillation to help depth estimation, e.g., Guo *et al.* [48] use pre-trained stereo matching network as a proxy to provide a supervision for the monocular depth estimation. Similarly, Tosi *et al.* [125] use the traditional Semi-Global Matching (SGM) approach to calculate accurate proxy labels for the same purpose. Pilzer *et al.* [99] propose to use the principle of distillation to transfer knowledge from their whole network to the student network which is a part of the teacher network. Different from these existing approaches, the proposed method enforces not only the output similarity, but also the intermediate-feature similarity across the pre-trained stereo network and the student network, with an expectation of further reducing the performance gap between the single-image and the stereo-image depth estimations.

The framework of our depth-aware convolutional neural network (DACNN) consists of a pre-trained stereo network as the teacher and a monocular depth estimation network as a student. Overall the main contributions of our work in this work include: Firstly, we design a novel depth-aware convolution operation in DACNN to learn the depth with the help of depth cues. Secondly, we also introduce a pre-trained stereo network into DACNN to provide additional supervision on both intermediate features and output of the student through knowledge distillation. Thirdly, our method achieves the state-of-the-art performance for single-image depth estimation on the KITTI online benchmark [37] and the KITTI Eigen split [32].

6.2 Method

The framework of our method DACNN has been presented in Figure 6.2, which consists of a pre-trained stereo network (teacher) and a monocular network (student). In this section, we elaborate on the proposed depth-aware convolution operation and the specially designed distillation loss function.

6.2.1 Feature extraction

Both the teacher and student networks use ResNest-50 [157] to extract features from the input, followed by Atrous Spatial Pyramid Pooling (ASPP) [17] with dilation rates $(1, 6, 12, 18)$ to further extract features from multiple receptive fields. Specifically, the stereo network (teacher) takes the stereo image pair as the input and the left and right images share weights during the pre-training. The output of the student network is denoted as f_s and that of the teacher network as f_l and f_r.

6.2.2 Depth guidance generation

The depth guidance, also referred to as depth cues, is an intermediate depth feature generated from both the teacher and student networks. For the teacher network, the

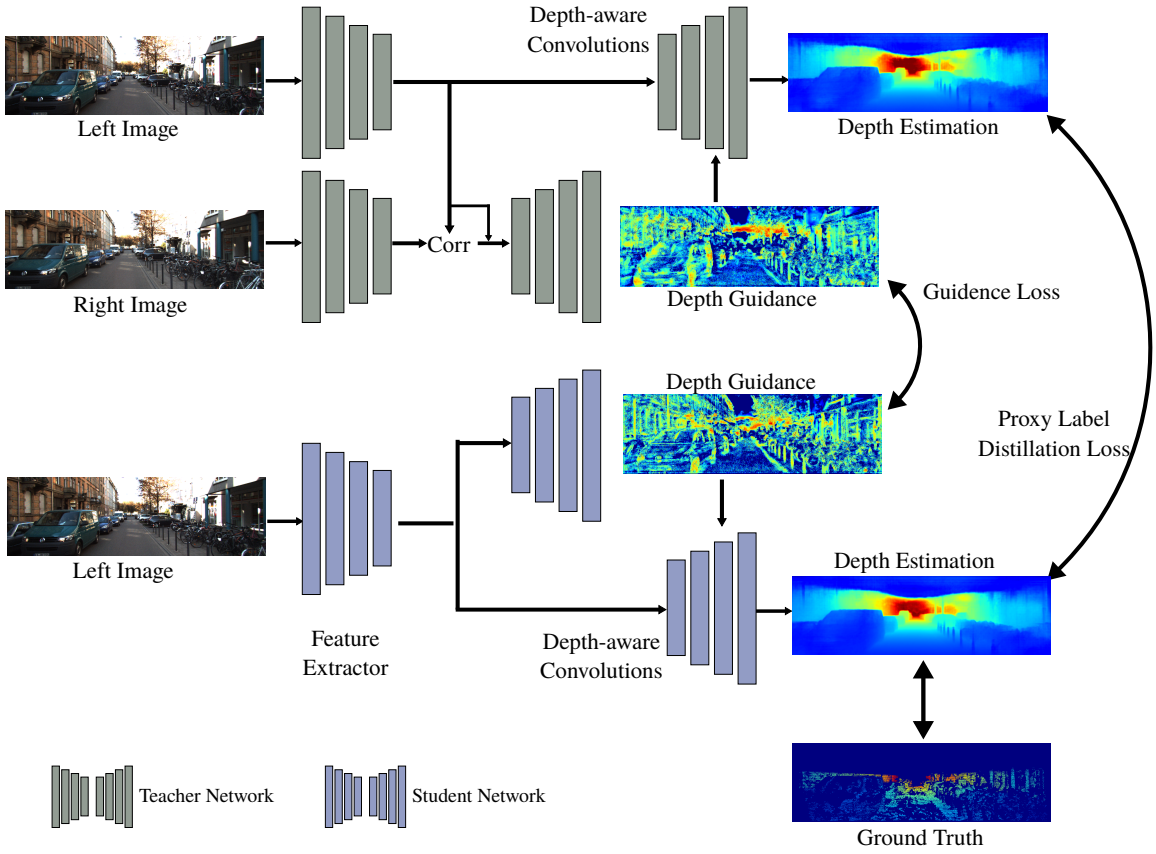

Figure 6.2 Framework of the proposed DACNN. The pre-trained stereo network (teacher) shown in the top takes the stereo image pair as the input while the monocular network (student) shown at the bottom takes the single image as the input. We constrain both the output similarity and the intermediate-feature similarity across the teacher and the student.

extracted features (f_l and f_r) of the stereo pair are passed into a correlation block, consisting of a correlation layer [30], a 3×3 convolutional layer and a batch normalization layer, to calculate the matching volume. In parallel, the extracted feature from left image (f_l) is also passed into the convolutional layer and the batch normalization layer and the result is concatenated with the matching volume to form the output O_C. Here, we consider a maximum displacement of 24 pixels when calculating the matching volume, which corresponds to 192 pixels in the input image.

For the student network, we employ the Asymmetric Pyramid Non-Local (AP-

NonLocal) Block [166] on the extracted feature of the input single image f_s to obtain the output O_A. Specifically, it first calculates the query Q with an MLP layer:

$$Q = T(f_s)W_q, \tag{6.1}$$

where T is flatten operation which reshapes the feature to $C \times N$, C is the channel number, $N = H \times W$ is the multiplication of spatial width and height, and W_q is the weight of an MLP layer. Then the Key and Value are obtained using different ways. For Key,

$$K = C(T(P_1(f_s)), T(P_2(f_s)), T(P_3(f_s)), T(P_4(f_s)))W_k, \tag{6.2}$$

where C is the concatenation operation, P_1, P_2, P_3, and P_4 are pooling operation which has different target resolution, and W_k is the weight of an MLP layer. Similarly, we can obtain the Value V by

$$V = C(T(P_1(f_s)), T(P_2(f_s)), T(P_3(f_s)), T(P_4(f_s)))W_v, \tag{6.3}$$

where W_v is the weight of an MLP layer. Finally, the output O_A can be formulated with self-attention operation as

$$O_A = softmax(QK^T)V. \tag{6.4}$$

Then, O_C and O_A are sent to the depth guidance generation branch separately to obtain the depth guidance of the teacher and student networks. The structure of the branch is shown in Figure 6.3, which contains several 3×3 convolution layers and upsampling layers. Finally, we get the depth guidance in three scales for both the teacher and student networks, that are denoted by g_1^t, g_2^t, g_3^t, and g_1^s, g_2^s and g_3^s, respectively.

6.2.3 Depth-aware convolution

The depth-aware convolution calculates the value for each of the positions based on the depth guidance that was obtained in last step of the previous subsection, and such convolution is employed by both the teacher and student networks.

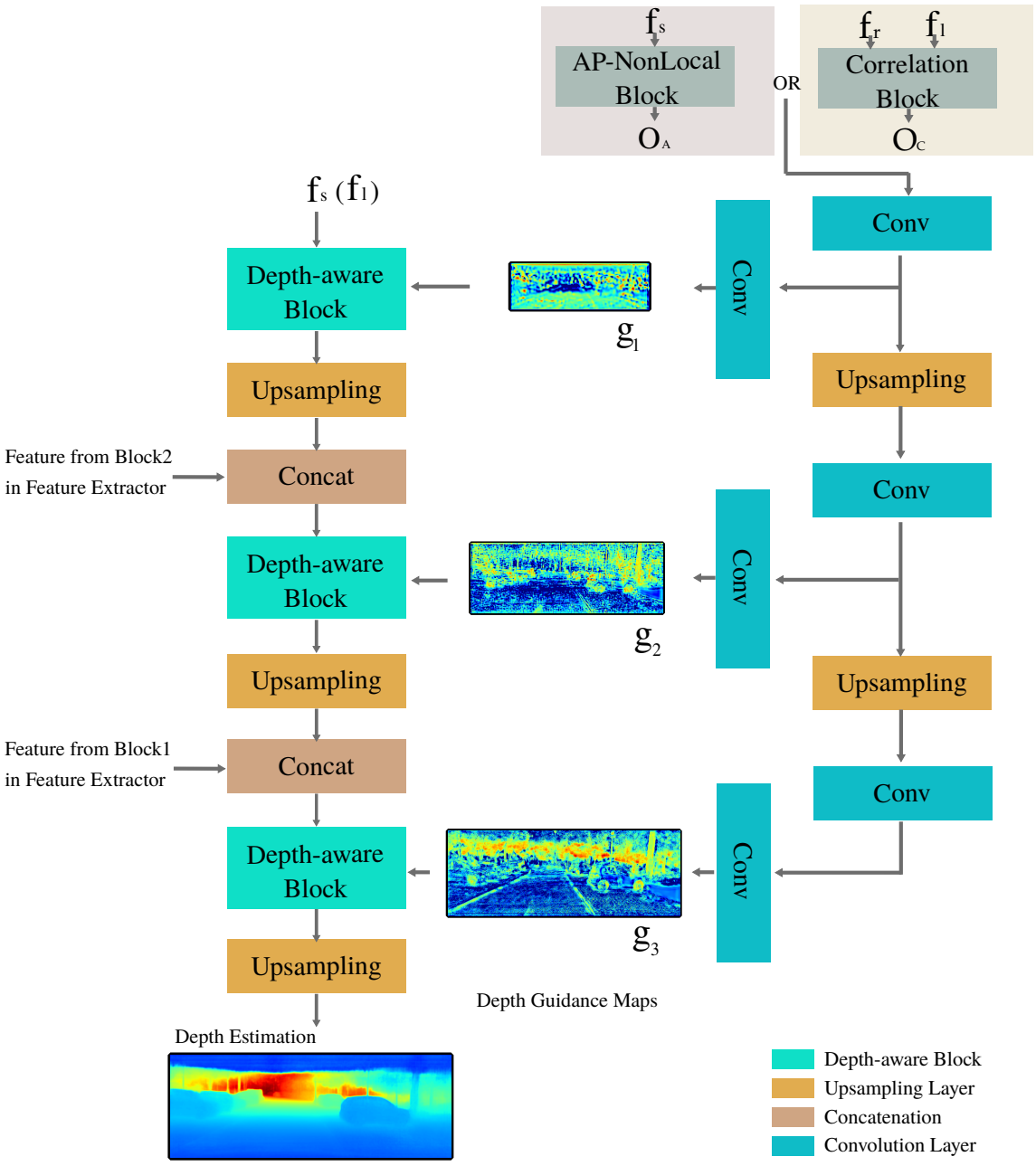

Figure 6.3 The architecture of the depth guidance estimation branch and the depth map estimation branch in the proposed DACNN. Note that the teacher network and the student network take different input and use different blocks at the beginning of the depth guidance estimation branch. After that, the architectures of the teacher network and the student network are the same.

Let $\mathcal{P}_d = \{-d, 0, d\} \times \{-d, 0, d\}$ represent the receptive field with dilation d, then a standard 3×3 convolutional operation acting on the position $\mathbf{p}$, which takes the single feature $\mathbf{f} \in \mathbb{R}^{C \times H \times W}$ as an input and outputs another feature $\mathbf{f}' \in \mathbb{R}^{C' \times H \times W}$, can be defined by

$$f'_j(\mathbf{p}) = \sum_{i=1}^{C} \sum_{\mathbf{o} \in \mathcal{P}_d} w_{i,j}(\mathbf{o}) f_i(\mathbf{p} + \mathbf{o}), \quad j = 1, \cdots, C', \tag{6.5}$$

where $(w_{i,j}(\mathbf{o})) \in \mathbb{R}^{3 \times 3 \times C \times C'}$ is the weight tensor of a filter, and C and C' are the numbers of channels of the input feature $\mathbf{f}$ and the output feature $\mathbf{f}'$, respectively (these two features have the same height H and width W). Following the previous works [119, 133], we propose the depth-aware convolution across different dilation d's based on depth guidance g as follows:

$$f'_j(\mathbf{p}) = \sum_{i=1}^{C} \sum_{\mathbf{o} \in \mathcal{P}_d} w_{i,j}(\mathbf{o}) K(g(\mathbf{p}), g(\mathbf{p} + \mathbf{o})) f_i(\mathbf{p} + \mathbf{o}), \quad j = 1, \cdots, C', \tag{6.6}$$

where K is a Gaussian operation which makes the convolution to be depth adaptive.

For each scale of depth guidance, we use multiple dilations d ($d \in \{1, 6, 12, 18\}$) during the convolution and obtain multi-scale depth features $f'(d)$. These features are concatenated and then fed into a 1×1 standard convolution layer. All of the above steps compose a depth-aware block. Following this way, we construct multiple depth-aware blocks and upsampling layers to upscale and refine the depth map as shown in Figure 6.3. Each upsampling layer doubles the resolution of the results and is followed by a 3×3 standard convolution layer and an ReLU layer. We also concatenate the features from the first two blocks of the feature extractor with the results of upsampling layer to combine the high-level and low-level information. Finally, we obtain the depth map whose resolution is the same as the original image's resolution.

6.2.4 Loss function

To pre-train the teacher network, the loss function is defined by the per-pixel loss $\mathcal{L}_{pixel}$ to measure the distance between the ground truth $d^*_{i,j}$ and the final estimated

depth map $d_{i,j}$, i.e.,

$$\mathcal{L}^T = \mathcal{L}_{pixel} = \frac{1}{N}\sum_{(i,j)} |d_{ij} - d^*_{ij}|. \tag{6.7}$$

For the proposed student network, we also adopt the per-pixel loss $\mathcal{L}_{pixel}$. In addition, we use the smooth loss to encourage the estimated depth map to be locally smooth, which is defined as:

$$\mathcal{L}_{smooth} = \frac{1}{N}\sum_{(i,j)} |\varphi_x d_{ij}| e^{-|\varphi_x I_{ij}|} + \varphi_y d_{ij}| e^{-|\varphi_y I_{ij}|}, \tag{6.8}$$

where I is the input image, and the function φ_x and φ_y calculate the intensity gradients between the neighboring pixels along the x and y directions.

Two more loss functions are proposed for distillation, namely the proxy label transfer loss $\mathcal{L}_{proxy}$ and the guidance transfer loss $\mathcal{L}_{guidance}$, respectively. The former one aims at constraining the output of the student network which uses the estimated result ($\hat{d}_{i,j}$) from the teacher network as a proxy ground truth to coach the student which is defined by

$$\mathcal{L}_{proxy} = \frac{1}{N}\sum_{(i,j)} |d_{ij} - \hat{d}_{ij}|. \tag{6.9}$$

The latter one is to constrain the similarity between the depth guidance from the teacher and the student network. To achieve this, a softmax operation is firstly applied to convert the multi-scale guidance maps into distributions, i.e.,

$$p^t_k = \text{softmax}(g^t_k),\ k = 1, 2, 3, \tag{6.10}$$

$$p^s_k = \text{softmax}(g^s_k),\ k = 1, 2, 3, \tag{6.11}$$

and then the Kullback-Leibler (KL) divergence is adopted to measure the dissimilarity of the distributions. Specifically, it is defined by

$$\mathcal{L}_{guidance} = \sum_{k=1}^{3} \text{KL}(p^t_k \| p^s_k). \tag{6.12}$$

The loss function for the student network is finally defined as:

$$\mathcal{L}^S = \mathcal{L}_{pixel} + \alpha\mathcal{L}_{smooth} + \beta\mathcal{L}_{proxy} + \gamma\mathcal{L}_{guidance}, \tag{6.13}$$

where α, β and γ are weighting factors which are set to 0.01, 0.01 and 1000, respectively.

6.3 Experiment

6.3.1 Datasets and evaluation metrics

We use the following popular datasets in experiments for performance evaluation and comparison of the proposed method with many existing state-of-the-art approaches on the monocular depth estimation task.

KITTI online benchmark [37]: The KITTI dataset contains over 93K outdoor images and depth maps with the resolution of $1,240 \times 374$. All the images are captured on driving cars by stereo cameras and a Lidar sensor. We use the images from "city", "residential", "road" and "campus" categories to train our model and test on the official test set including 500 images. The scale invariant logarithmic error (SILog), the relative squared error (sqErrorRel), the relative absolute error (absErrorRel) and the root mean squared error of the inverse depth (iRMSE) are used to evaluate the performance on this dataset.

KITTI Eigen split [32]: Eigen *et al.*provide a subset of training and testing split from the KIITI dataset for monocular depth estimation, which is commonly used in recent works. The training set include 23,488 images from 32 scenarios, and the testing set includes 697 images from 29 scenarios. Following [32], we use the absolute relative difference (Abs Rel), the squared relative difference (Sq Rel), the root-mean-square error (RMSE) and the log RMSE (RMSE log) as the error metrics and the accuracy with threshold $\delta = \{1.25, 1.25^2, 1.25^3\}$ as the accuracy metrics. For all error metrics, the lower the better, while for the accuracy metrics, the higher the better.

6.3.2 Experimental settings

The proposed method, DACNN, is implemented using PyTorch, and we pre-train the teacher network and perform the knowledge distillation on the student network using two Nvidia 2080Ti GPUs with the Adam solver (the momentum parameters $\beta_1 = 0.9, \beta_2 = 0.999$). The models are trained from scratch with a batch size of 6. Following [142], we employ the poly learning rate policy from the base learning rate 10^{-4} with the power $p = 0.9$. We pre-train the teacher network for 10 epochs and train the student network (distillation) for 15 epochs.

We also perform color normalization on these two datasets for data preprocessing, and during training, all images were randomly cropped to the size of 256×512. To avoid the over-fitting problem, we use the data augmentation strategy in [32]. Specifically, the images are augmented with the random contrast, brightness, and color adjustment in a range of $[0.9, 1.1]$ with 50% of chance.

During the test phase, we split each of the testing image to overlapping windows with the same cropping size as in the training processing, and then obtain the estimated depth values in overlapped regions by averaging the estimations.

6.3.3 Results on the KITTI datasets

The quantitative results evaluated on the KITTI Eigen split are reported in Table 6.1, which shows that our DACNN achieves the best or close to the best performance in each of the error or accuracy metrics among all compared state-of-the-art networks. To exhibit the visual improvements, we also show some depth estimation results from the test set of Eigen split in Figure 6.4, from which it is easy to see that the estimated depth maps by our method are much smoother and possess clearer boundary between objects than that by DORN.

The quantitative results evaluated from the KITTI online leaderboard are reported in Table 6.2. For the quantitative comparison, we again present some visualization

Table 6.1 Performance comparison of DACNN and some existing state-of-the-art networks on the KITTI Eigen split.

Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Make3D [112]	0.280	3.012	8.734	0.361	0.601	0.820	0.926
Eigen *et al.* [32]	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al.* [85]	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Godard *et al.* [41]	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Kuznietsov *et al.* [77]	0.113	0.741	4.621	0.189	0.862	0.960	0.986
Gan *et al.* [34]	0.098	0.666	3.933	0.173	0.890	0.964	0.985
Yin *et al.* [145]	**0.072**	-	3.256	0.117	0.938	**0.993**	**0.998**
DORN [33]	**0.072**	0.307	**2.727**	0.120	0.932	0.984	0.994
Our method (DACNN)	0.073	**0.304**	2.801	**0.116**	**0.939**	0.990	0.997

results results from our method and DORN [33] in Figure 6.5.

Table 6.2 Performance comparison of DACNN and some existing state-of-the-art networks on the KITTI online benchmark.

Method	SILog	sqErrorRel	absErrorRel	iRMSE
DABC *et al.* [80]	14.49	4.08	12.72	15.53
Guo *et al.* [48]	13.41	2.86	10.60	15.06
Zhang *et al.* [159]	13.08	2.72	10.27	13.95
Yin *et al.* [145]	12.65	2.46	10.15	13.02
DORN [33]	11.77	2.23	8.78	12.98
Our method (DACNN)	12.95	2.60	10.35	13.95

6.3.4 Ablation study

In this study, in order to demonstrate effectiveness of the proposed depth-aware blocks in DACNN and the knowledge distillation from the teacher network, we conduct ablation studies to compare some model variants for DACNN on the Eigen split of KITTI dataset. The results are reported in Table 6.3, from which we can see that the

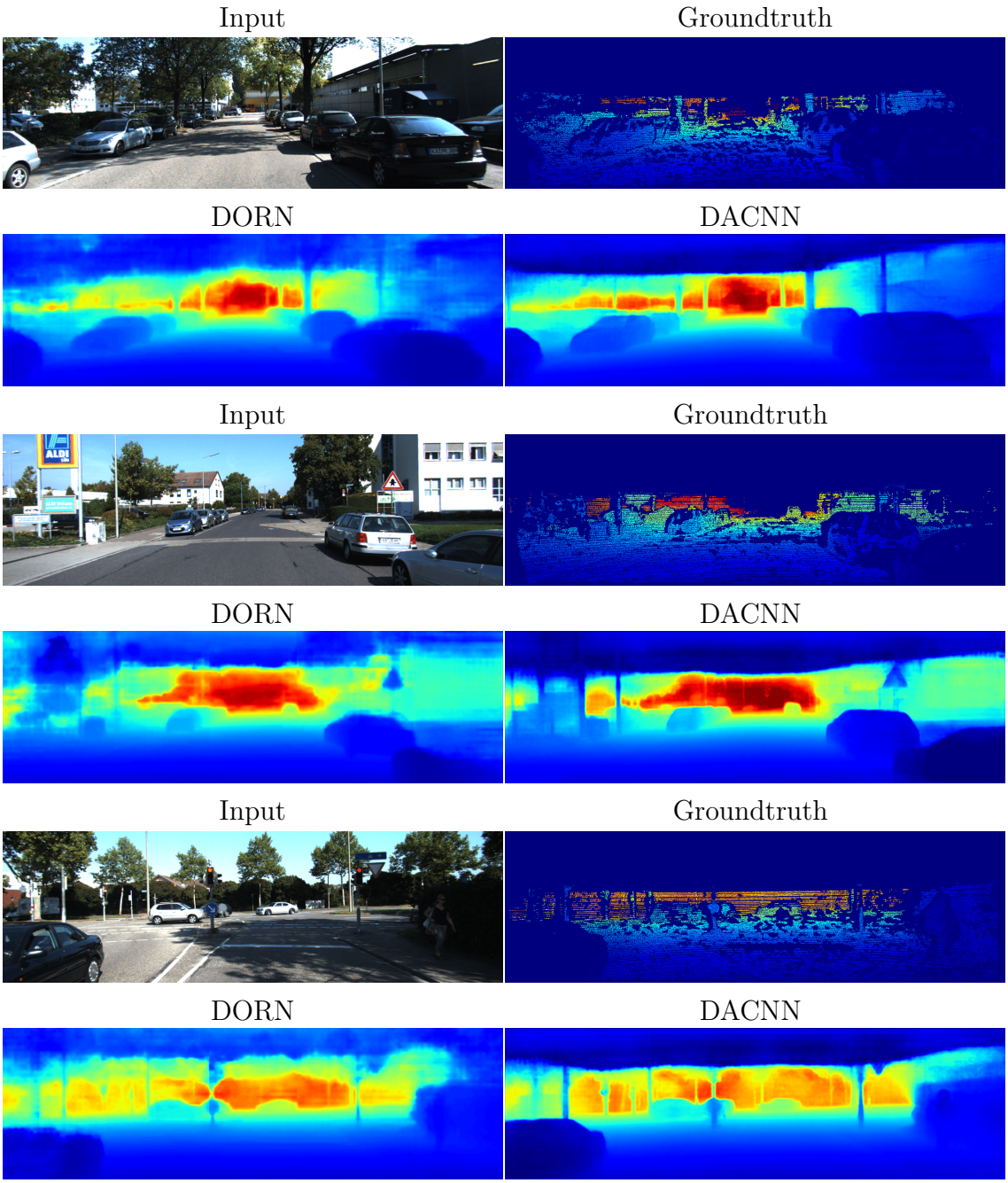

Figure 6.4 Some depth estimation results from the test set of the KITTI Eigen split.

depth-aware blocks (DAB) are useful for improving the monocular depth estimation, and the knowledge distillation (KD) from the teacher network can further improve

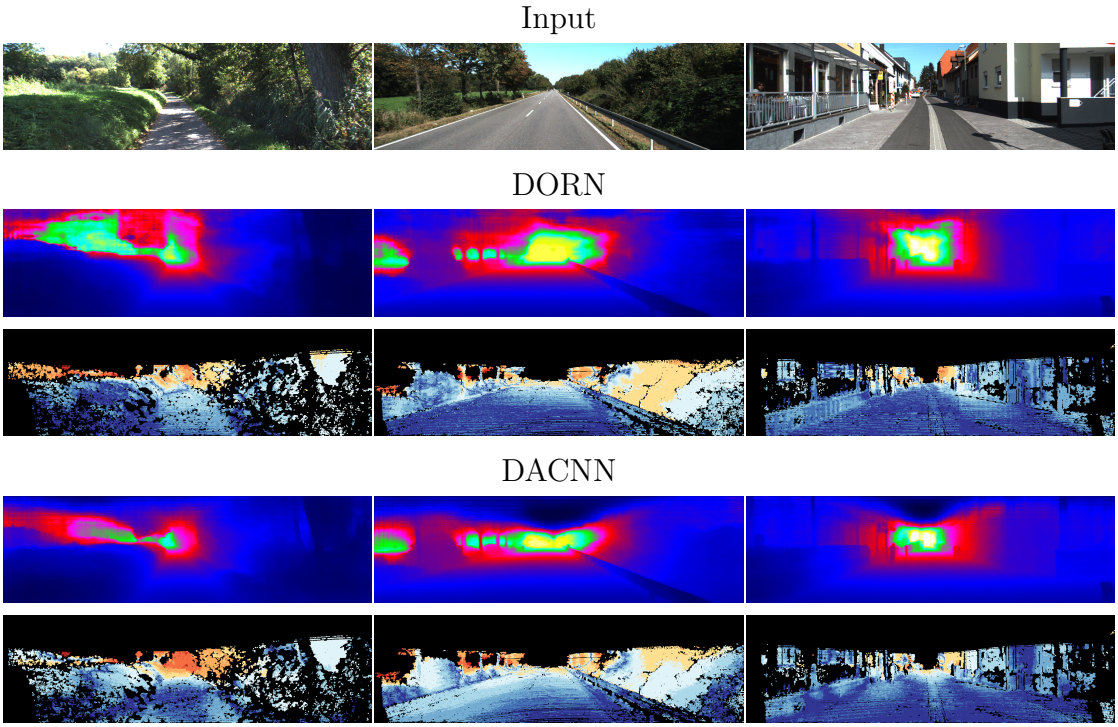

Figure 6.5 The depth estimation results from the KITTI online leaderboard. For each input image, we show the estimated depth maps and the corresponding error maps from DORN [33] and our DACNN, respectively.

the overall performance.

Table 6.3 Performance comparison of some model variants of DACNN on the KITTI Eigen split.

Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
w/o DAB	0.086	0.563	3.675	0.162	0.916	0.972	0.991
w/o KD	0.079	0.462	3.174	0.126	0.925	0.981	0.993
DACNN	0.073	0.304	2.801	0.116	0.939	0.990	0.997

6.3.5 Visualization of depth guidance

To better illustrate the impact of the depth-aware blocks, we visualize the depth guidance map of several sample images in the KITTI Eigen split dataset in Figure

6.6. From these results, we clearly observe that the depth guidance map can provide high level information, especially the boundary cues for depth-aware blocks.

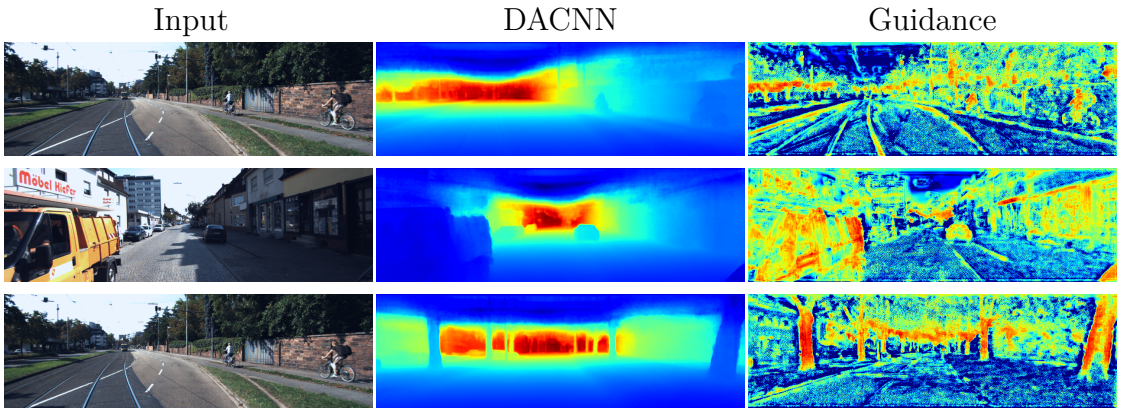

Figure 6.6 The visualization of depth guidance. For each row, the first column is the input image, the second is the estimated depth map from DACNN, and the last is the guidance map g_3^s.

6.4 Chapter Summary

In this work, we develope a new depth-aware convolution operation to learn depth from single image by leveraging depth cues. In addition, we incorporate a pre-trained stereo network as a teacher to provide additional supervision for the features and output generated by the student network which is a monocular depth estimation network. Experimental results on the KITTI Eigen split and online benchmark demonstrate that the proposed method can significantly improve the accuracy of monocular depth estimation.

Chapter 7

Conclusion

7.1 Conclusion

To summarize, this dissertation focus on depth estimation from the RGB images, including stereo matching, depth estimation from monocular video, and single image depth estimation. To explore the rich and comprehensive spatial correspondence across images, three novel methods were proposed in this dissertation based on three representative inputs.

In the first work, a new deep neural network for stereo matching was developed, where the pyramid cost volumes are constructed with the semantic and spatial correspondence on multiple levels. We also introduce a multi-cost aggregation module proposed to fuse the extracted multilevel correspondence. Comprehensive experiments on Scene Flow, KITTI 2015, KITTI 2012, and Cityscapes datasets demonstrated that the proposed approach with semantic and multi-level spatial correspondence can improve the accuracy of stereo matching over many existing state-of-the-art neural networks.

In the second work, we developed a novel end-to-end network for depth estimation from monocular videos. We carefully designed a spatial correspondence module to match the features between the reference frame and its adjacent frames. To further speed up the feature correspondence, We use the Smolyak sparse grids for patch down-sampling. A direction-based attention mechanism is proposed to learn the importance of correspondence in different directions, and a refinement subnetwork is developed to refine the initially estimated depth maps. Extensive experiments on the KITTI and Cityscapes datasets justify that the proposed SC-GAN promotes the state-of-the-art performance of the depth estimation from monocular video.

In the last work, we incorporate a pre-trained stereo network as a teacher to provide additional supervision for the features and output generated by the student network which is a monocular depth estimation network. By using the stereo network as a teacher, the student network can learn depth guidance from spatial correspon-

dence. To further leverage the depth guidance map, we developed a new depth-aware convolution operation to learn depth from a single image. Experimental results on the KITTI Eigen split and online benchmark demonstrate that the proposed method can significantly improve the accuracy of monocular depth estimation.

7.2 Future works

Based on the above studies on depth estimation from images, we can highlight the challenges for future research works.

Firstly, the resolution of the estimated depth map is usually low. Depth estimation from images is a fundamental task that can be used for many applications, such as autonomous driving and augmented reality, and these applications require taking high-accuracy and high-resolution depth maps. However, the current model cannot generate high-resolution depth maps accurately. There are two solutions to these issues. The first one is using the image super-resolution method to upscale the input image, while it will increase the computation cost for depth estimation. The second one directly upsamples the depth map using the generative adversarial network.

Secondly, most depth estimation datasets focus on only one specific dataset. The generalization ability and robustness of the deep learning approaches always depend on the quality of datasets. However, the captured scenes of the existing datasets are limited. To enhance the generalization ability, it is interesting to work on the depth estimation domain adaption, *i.e.*the model trained on one domain is used to estimate the depth of another domain. Besides, a new dataset including diverse and complicated scenes will benefit this task.

Thirdly, the analysis of dynamic objects is missing. The moving objects will change the spatial correspondence between the frames. A possible solution to eliminate the issues caused by dynamic objects can be warping the dynamic objects to the same 3d position based on the scene flow or a generative adversarial network.

Lastly, the current work of monocular depth estimation approaches can only obtain the relative depth instead of the absolute depth. In the future, more physics features (*e.g.*, vanishing points) and prior knowledge of real-world objects can be involved in the depth estimation of the images. Moreover, we can consider using the off-the-shelf depth value, *e.g.*, sparse LiDAR data to correct the depth obtained by a neural network.

Bibliography

[1] Filippo Aleotti et al. "Generative adversarial networks for unsupervised monocular depth prediction". In: *European Conference on Computer Vision (ECCV)*. 2018.

[2] Amir Atapour-Abarghouei and Toby P Breckon. "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2800–2810.

[3] Eu-Tteum Baek and Yo-Sung Ho. "Depth Estimation and View Synthesis using Vanishing Point from Single View Image". In: 2014.

[4] Volker Barthelmann, Erich Novak, and Klaus Ritter. "High dimensional polynomial interpolation on sparse grids". In: *Advances in Computational Mathematics* 12.4 (2000), pp. 273–288.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer. 2006, pp. 404–417.

[6] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. "Adabins: Depth estimation using adaptive bins". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 4009–4018.

[7] Joydeep Biswas and Manuela Veloso. "Depth camera based localization and navigation for indoor mobile robots". In: *RGB-D Workshop at RSS*. Vol. 2011. 2011, p. 21.

[8] Michael Bleyer et al. "Object stereo–Joint stereo matching and object segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[9] Konstantinos Bousmalis et al. "Unsupervised Pixel-Level Domain Adaptation With Generative Adversarial Networks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[10] Matthew Brown, Gang Hua, and Simon Winder. “Discriminative learning of local image descriptors”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.1 (2011), pp. 43–57.

[11] Hans-Joachim Bungartz and Michael Griebel. “Sparse grids”. In: *Acta numerica* 13 (2004), pp. 147–269.

[12] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. “Estimating depth from monocular images as classification using deep fully convolutional residual networks”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.11 (2017), pp. 3174–3182.

[13] Vincent Casser et al. “Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. Vol. 33. 01. 2019, pp. 8001–8008.

[14] Jia-Ren Chang and Yong-Sheng Chen. “Pyramid Stereo Matching Network”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[15] Chenyi Chen et al. “DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015.

[16] Liang-Chieh Chen et al. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848.

[17] Liang-Chieh Chen et al. “Rethinking atrous convolution for semantic image segmentation”. In: *arXiv preprint arXiv:1706.05587* (2017).

[18] Qiuyu Chen et al. “Virtual Blood Vessels in Complex Background Using Stereo X-Ray Images.” In: *IEEE International Conference on Computer Vision Workshops*. 2017, pp. 99–106.

[19] Richard Chen et al. “Rethinking monocular depth estimation with adversarial training”. In: *arXiv preprint arXiv:1808.07528* (2018).

[20] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. “Darkrank: Accelerating deep metric learning via cross sample similarities transfer”. In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2018.

[21] Jingchun Cheng et al. “SegFlow: Joint Learning for Video Object Segmentation and Optical Flow”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[22] Xinjing Cheng, Peng Wang, and Ruigang Yang. “Learning depth with convolutional spatial propagation network”. In: *arXiv preprint arXiv:1810.02695* (2018).

[23] Xuelian Cheng et al. “Hierarchical neural architecture search for deep stereo matching”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 22158–22169.

[24] Marius Cordts et al. “The cityscapes dataset for semantic urban scene understanding”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3213–3223.

[25] Arun CS Kumar, Suchendra M Bhandarkar, and Mukta Prasad. “Monocular depth prediction using generative adversarial networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 300–308.

[26] Angela Dai et al. “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[27] Jifeng Dai et al. “Deformable convolutional networks”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 764–773.

[28] Weijian Deng et al. “Image-Image Domain Adaptation With Preserved Self-Similarity and Domain-Dissimilarity for Person Re-Identification”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[29] Xuanyi Dong et al. “Style Aggregated Network for Facial Landmark Detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[30] Alexey Dosovitskiy et al. “Flownet: Learning optical flow with convolutional networks”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015, pp. 2758–2766.

[31] David Eigen and Rob Fergus. “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015, pp. 2650–2658.

[32] 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*. 2014, pp. 2366–2374.

[33] Huan Fu et al. “Deep Ordinal Regression Network for Monocular Depth Estimation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2002–2011.

[34] Yukang Gan et al. “Monocular depth estimation with affinity, vertical pooling, and label enhancement”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 224–239.

[35] Ravi Garg et al. “Unsupervised cnn for single view depth estimation: Geometry to the rescue”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 740–756.

[36] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for autonomous driving? the KITTI vision benchmark suite”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2012, pp. 3354–3361.

[37] Andreas Geiger et al. “Vision meets robotics: The KITTI dataset”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[38] Thomas Gerstner. “Multiresolution visualization and compression of global topographic data”. In: *GeoInformatica* 7.1 (2003), pp. 7–32.

[39] Spyros Gidaris and Nikos Komodakis. “Detect, Replace, Refine: Deep Structured Prediction for Pixel Wise Labeling”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[40] Ross Girshick. “Fast R-CNN”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015, pp. 1440–1448.

[41] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow. “Unsupervised Monocular Depth Estimation With Left-Right Consistency”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[42] Clément Godard et al. “Digging into Self-Supervised Monocular Depth Prediction”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[43] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems*. 2014, pp. 2672–2680.

[44] Jianping Gou et al. “Knowledge distillation: A survey”. In: *International Journal of Computer Vision* 129.6 (2021), pp. 1789–1819.

[45] Xiaodong Gu et al. “Cascade cost volume for high-resolution multi-view stereo and stereo matching”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2495–2504.

[46] Fatma Guney and Andreas Geiger. “Displets: Resolving Stereo Ambiguities Using Object Knowledge”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[47] Xiaoyang Guo et al. “Group-wise correlation stereo network”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3273–3282.

[48] Xiaoyang Guo et al. “Learning monocular depth by distilling cross-domain stereo networks”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 484–500.

[49] Hyowon Ha et al. “High-quality depth from uncalibrated small motion clip”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5413–5421.

[50] Adam W Harley, Konstantinos G Derpanis, and Iasonas Kokkinos. “Segmentation-aware convolutional networks using local attention masks”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017, pp. 5038–5047.

[51] Kaiming He et al. “Deep residual learning for image recognition”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[52] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* (2015).

[53] Heiko Hirschmuller. “Accurate and efficient stereo processing by semi-global matching and mutual information”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. IEEE. 2005, pp. 807–814.

[54] Heiko Hirschmuller. “Stereo processing by semiglobal matching and mutual information”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 328–341.

[55] Heiko Hirschmuller and Daniel Scharstein. “Evaluation of cost functions for stereo matching”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2007, pp. 1–8.

[56] Judy Hoffman et al. “CyCADA: Cycle-Consistent Adversarial Domain Adaptation”. In: *International Conference on Machine Learning*. 2018, pp. 1989–1998.

[57] Derek Hoiem, Alexei A Efros, and Martial Hebert. “Automatic photo pop-up”. In: *ACM transactions on graphics (TOG)*. Vol. 24. 3. ACM. 2005, pp. 577–584.

[58] Berthold KP Horn and Brian G Schunck. “Determining optical flow”. In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203.

[59] Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-excitation networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7132–7141.

[60] Xiaowei Hu et al. “Direction-aware spatial context features for shadow detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[61] Po-Han Huang et al. “DeepMVS: Learning Multi-View Stereopsis”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[62] Sunghoon Im et al. “DPSNet: End-to-end Deep Plane Sweep Stereo”. In: *International Conference on Learning Representations*. 2019.

[63] Phillip Isola et al. “Image-to-image translation with conditional adversarial networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 5967–5976.

[64] Mengqi Ji et al. “SurfaceNet: An End-To-End 3D Neural Network for Multiview Stereopsis”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[65] Xu Jia et al. “Dynamic filter networks”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 667–675.

[66] Jianbo Jiao et al. “Look Deeper into Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[67] Jianbo Jiao et al. “Look Deeper into Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 53–69.

[68] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. “Perceptual losses for real-time style transfer and super-resolution”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 694–711.

[69] Kevin Karsch, Ce Liu, and Sing Bing Kang. “Depth extraction from video using non-parametric sampling”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2012, pp. 775–788.

[70] Alex Kendall, Yarin Gal, and Roberto Cipolla. “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7482–7491.

[71] Alex Kendall et al. “End-To-End Learning of Geometry and Context for Deep Stereo Regression”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[72] Alex Kendall et al. “End-to-end learning of geometry and context for deep stereo regression”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. 2017, pp. 66–75.

[73] Sameh Khamis et al. “StereoNet: Guided Hierarchical Refinement for Real-Time Edge-Aware Depth Prediction”. In: *arXiv preprint arXiv:1807.08865* (2018).

[74] Jangho Kim, SeongUk Park, and Nojun Kwak. “Paraphrasing complex network: Network compression via factor transfer”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 2760–2769.

[75] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[76] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. “Semi-Supervised Deep Learning for Monocular Depth Map Prediction”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[77] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. “Semi-supervised deep learning for monocular depth map prediction”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6647–6655.

[78] Bo Li et al. “Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1119–1127.

[79] Quanquan Li, Shengying Jin, and Junjie Yan. “Mimicking very efficient network for object detection”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6356–6364.

[80] Ruibo Li et al. “Deep attention-based classification network for robust depth prediction”. In: *Asian Conference on Computer Vision (ACCV)*. Springer. 2018, pp. 663–678.

[81] Zhengfa Liang et al. “Learning Deep Correspondence through Prior and Posterior Feature Constancy”. In: *arXiv preprint arXiv:1712.01039* (2017).

[82] Di Lin et al. “Multi-Scale Context Intertwining for Semantic Segmentation”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 603–619.

[83] Guosheng Lin et al. “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1925–1934.

[84] Fayao Liu, Chunhua Shen, and Guosheng Lin. “Deep convolutional neural fields for depth estimation from a single image”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5162–5170.

[85] Fayao Liu et al. “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.

[86] Fayao Liu et al. “Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.10 (2016), pp. 2024–2039.

[87] Guilin Liu et al. “Image inpainting for irregular holes using partial convolutions”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 85–100.

[88] Yifan Liu et al. “Structured knowledge distillation for semantic segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2604–2613.

[89] Wenjie Luo, Alexander G. Schwing, and Raquel Urtasun. “Efficient Deep Learning for Stereo Matching”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[90] Reza Mahjourian, Martin Wicke, and Anelia Angelova. “Unsupervised learning of depth and ego-motion from monocular video using 3d geometric con-

straints". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 5667–5675.

[91] Nikolaus Mayer et al. "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[92] Moritz Menze and Andreas Geiger. "Object Scene Flow for Autonomous Vehicles". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[93] Alican Mertan, Damien Jade Duff, and Gozde Unal. "Single image depth estimation: An overview". In: *Digital Signal Processing* (2022), p. 103441.

[94] S Mahdi H Miangoleh et al. "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 9685–9694.

[95] Yue Ming et al. "Deep learning for monocular depth estimation: A review". In: *Neurocomputing* 438 (2021), pp. 14–33.

[96] Erich Novak and Klaus Ritter. "Global optimization using hyperbolic cross points". In: *State of the Art in global Optimization*. Springer, 1996, pp. 19–33.

[97] Jiahao Pang et al. "Cascade Residual Learning: A Two-Stage Convolutional Neural Network for Stereo Matching." In: *IEEE International Conference on Computer Vision Workshops*. Vol. 7. 2017.

[98] Nikolaos Passalis and Anastasios Tefas. "Learning deep representations with probabilistic knowledge transfer". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 268–284.

[99] Andrea Pilzer et al. "Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9768–9777.

[100] Andrea Pilzer et al. "Unsupervised Adversarial Depth Estimation using Cycled Generative Networks". In: *International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 587–595.

[101] Can Pu et al. "Sdf-GAN: Semi-supervised Depth Fusion with Multi-scale Adversarial Networks". In: *arXiv preprint arXiv:1803.06657* (2018).

[102] AN Rajagopalan, Subhasis Chaudhuri, and Uma Mudenagudi. "Depth estimation and image restoration using defocused stereo pairs". In: *IEEE transactions on pattern analysis and machine intelligence* 26.11 (2004), pp. 1521–1525.

[103] Pierluigi Zama Ramirez et al. "Geometry meets semantics for semi-supervised monocular depth estimation". In: *Asian Conference on Computer Vision (ACCV)*. Springer. 2018, pp. 298–313.

[104] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. "Vision transformers for dense prediction". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 12179–12188.

[105] René Ranftl et al. "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer". In: *IEEE transactions on pattern analysis and machine intelligence* (2020).

[106] Anurag Ranjan and Michael J. Black. "Optical Flow Estimation Using a Spatial Pyramid Network". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[107] Jerome Revaud et al. "Deepmatching: Hierarchical deformable dense matching". In: *International Journal of Computer Vision* 120.3 (2016), pp. 300–323.

[108] Danilo Jimenez Rezende et al. "Unsupervised learning of 3d structure from images". In: *Advances in Neural Information Processing Systems*. 2016, pp. 4996–5004.

[109] Adriana Romero et al. "Fitnets: Hints for thin deep nets". In: *International Conference on Learning Representations*. 2015.

[110] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[111] Ashutosh Saxena, Min Sun, and Andrew Y Ng. "Make3D: Learning 3d scene structure from a single still image". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5 (2009), pp. 824–840.

[112] Ashutosh Saxena, Min Sun, and Andrew Y Ng. "Make3d: Learning 3d scene structure from a single still image". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5 (2008), pp. 824–840.

[113] Amit Shaked and Lior Wolf. "Improved Stereo Matching With Constant Highway Networks and Reflective Confidence Learning". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

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

[115] Nathan Silberman et al. "Indoor segmentation and support inference from rgbd images". In: *European Conference on Computer Vision (ECCV)*. Springer. 2012, pp. 746–760.

[116] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *International Conference on Learning Representations*. 2014.

[117] Sergei Abramovich Smolyak. "Quadrature and interpolation formulas for tensor products of certain classes of functions". In: *Doklady Akademii Nauk*. Vol. 148. Russian Academy of Sciences. 1963, pp. 1042–1045.

[118] Xiao Song et al. "EdgeStereo: A Context Integrated Residual Pyramid Network for Stereo Matching". In: *Asian Conference on Computer Vision (ACCV)*. 2018.

[119] Hang Su et al. "Pixel-Adaptive Convolutional Neural Networks". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[120] Deqing Sun et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[121] Christian Szegedy et al. "Going deeper with convolutions". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9.

[122] Zachary Teed and Jia Deng. "DeepV2D: Video to Depth with Differentiable Structure from Motion". In: *arXiv preprint arXiv:1812.04605* (2018).

[123] Zachary Teed and Jia Deng. "Raft: Recurrent all-pairs field transforms for optical flow". In: *European conference on computer vision*. Springer. 2020, pp. 402–419.

[124] Federico Tombari et al. "Classification and evaluation of cost aggregation methods for stereo correspondence". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2008, pp. 1–8.

[125] Fabio Tosi et al. “Learning monocular depth estimation infusing traditional stereo knowledge”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9799–9809.

[126] Sheng-Po Tseng and Shang-Hong Lai. “Accurate depth map estimation from video via MRF optimization”. In: *Visual Communications and Image Processing (VCIP)*. IEEE. 2011, pp. 1–4.

[127] Benjamin Ummenhofer et al. “DeMoN: Depth and Motion Network for Learning Monocular Stereo”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[128] Aaron Van den Oord et al. “Conditional image generation with pixelcnn decoders”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 4790–4798.

[129] Igor Vasiljevic et al. “Diode: A dense indoor and outdoor depth dataset”. In: *arXiv preprint arXiv:1908.00463* (2019).

[130] Chaoyang Wang et al. “Learning depth from monocular videos using direct methods”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2022–2030.

[131] Kaixuan Wang and Shaojie Shen. “MVDepthNet: Real-time Multiview Depth Estimation Neural Network”. In: *International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 248–257.

[132] Rui Wang et al. “Vplnet: Deep single view normal estimation with vanishing points and lines”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 689–698.

[133] Anne S. Wannenwetsch and Stefan Roth. “Probabilistic Pixel-Adaptive Refinement Networks”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[134] Jamie Watson et al. “Self-supervised monocular depth hints”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 2162–2171.

[135] Jamie Watson et al. “The temporal opportunist: Self-supervised multi-frame monocular depth”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 1164–1174.

[136] Cho-Ying Wu et al. “Toward Practical Monocular Indoor Depth Estimation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[137] Zhenyao Wu et al. “Semantic stereo matching with pyramid cost volumes”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 7484–7493.

[138] Jiafeng Xie et al. “Improving fast segmentation with teacher-student learning”. In: *arXiv preprint arXiv:1810.08476* (2018).

[139] Dan Xu et al. “Multi-scale continuous crfs as sequential deep networks for monocular depth estimation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5354–5362.

[140] Haofei Xu and Juyong Zhang. “Aanet: Adaptive aggregation network for efficient stereo matching”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 1959–1968.

[141] Guorun Yang et al. “SegStereo: Exploiting Semantic Information for Disparity Estimation”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[142] Maoke Yang et al. “Denseaspp for semantic segmentation in street scenes”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3684–3692.

[143] Nan Yang et al. “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 817–833.

[144] Yao Yao et al. “MVSNet: Depth Inference for Unstructured Multi-view Stereo”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[145] Wei Yin et al. “Enforcing geometric constraints of virtual normal for depth prediction”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 5684–5693.

[146] Wei Yin et al. “Learning to recover 3d scene shape from a single image”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 204–213.

[147] Zhichao Yin, Trevor Darrell, and Fisher Yu. “Hierarchical discrete distribution decomposition for match density estimation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6044–6053.

[148] Zhichao Yin and Jianping Shi. “GeoNet: Unsupervised learning of dense depth, optical flow and camera pose”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 1983–1992.

[149] Changqian Yu et al. “BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation”. In: *European Conference on Computer Vision (ECCV)*. 2018.

[150] Fisher Yu and Vladlen Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *International Conference on Learning Representations (ICLR)*. 2016.

[151] Jure Zbontar and Yann LeCun. “Stereo matching by training a convolutional neural network to compare image patches”. In: *Journal of Machine Learning Research* 17.1-32 (2016), p. 2.

[152] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. “Adaptive deconvolutional networks for mid and high level feature learning”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. 2011, pp. 2018–2025.

[153] Huangying Zhan et al. “Unsupervised Learning of Monocular Depth Estimation and Visual Odometry With Deep Feature Reconstruction”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[154] Chi Zhang et al. “MeshStereo: A Global Stereo Model With Mesh Alignment Regularization for View Interpolation”. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015.

[155] Feihu Zhang et al. “Domain-invariant stereo matching networks”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 420–439.

[156] Feihu Zhang et al. “Ga-net: Guided aggregation net for end-to-end stereo matching”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 185–194.

[157] Hang Zhang et al. “ResNeSt: Split-Attention Networks”. In: *arXiv preprint arXiv:2004.08955* (2020).

[158] Zhenyu Zhang et al. “Joint Task-Recursive Learning for Semantic Segmentation and Depth Estimation”. In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 235–251.

[159] Zhenyu Zhang et al. “Pattern-Affinitive Propagation across Depth, Surface Normal and Semantic Segmentation”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4106–4115.

[160] Hengshuang Zhao et al. “Pyramid Scene Parsing Network”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[161] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. "DeepTAM: Deep tracking and mapping". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 822–838.

[162] Tinghui Zhou et al. "Unsupervised learning of depth and ego-motion from video". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. 2017, p. 7.

[163] Jun-Yan Zhu et al. "Generative visual manipulation on the natural image manifold". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 597–613.

[164] Jun-Yan Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2017.

[165] Xizhou Zhu et al. "Deformable convnets v2: More deformable, better results". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9308–9316.

[166] Zhen Zhu et al. "Asymmetric non-local neural networks for semantic segmentation". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 593–602.