Gottfried Wilhelm Leibniz Universität Hannover
Institut für Photogrammetrie und GeoInformation

Masterarbeit

Jointly Estimate Geometry and Semantics from Binocular Stereo Images

Yu Cao, B. Sc.

Matrikelnr. 10007863

Betreuer: Dr.-Ing. Max Mehltretter
Erstprüfer: Prof. Dr.-Ing. habil. Christian Heipke

Hannover, March 2022

Statement

I declare that this thesis is the result of independent research conducted by me under the guidance of my supervisor. It does not contain the results of any other scientific research that has been published or written by any other individuals or groups, except for those already cited in the thesis. Furthermore, I state that this work in the same or a similar form has not been submitted to an examination authority.

Signature

Place, Date

Abstract

Multitasking frameworks are attracting more and more attention from academia and industry, as the field of computer vision evolves in various directions. The need for perceiving the environment in real-world application scenarios is multifaceted. One of the most important goals of multitasking frameworks is to improve the performance of models by combining multiple tasks. Binocular stereo matching (also known as binocular disparity estimation) has been intensively studied as an important method for reconstructing depth information for decades. Although there have been many multitasking frameworks for target recognition and classification, there are relatively few studies on multi-task learning for reconstructing depth information, especially stereo matching. It would be a meaningful step forward if panoptic segmentation and disparity estimation could be combined in one method and made to work together.

In the present work, we propose a unified, lightweight and well-structured method that can simultaneously perform disparity estimation and panoptic segmentation. We experimentally demonstrate the feasibility of the approach and structural compatibility with multiple tasks. In addition, we designed two feature fusion modules, AFM and WFM, and further investigated the use of panoptic information to improve the disparity estimation by feature fusion. Moreover, through experiments on training strategies, datasets, and structural compatibility, we have obtained a lot of first-hand information on the above topic, which could guide the direction of the subsequent research.

Keywords: disparity estimation, dense stereo matching, multi-task training, panoptic segmentation, computer vision, deep learning

Contents

1. Introduction

Reconstructing depth information (3D data) from images (2D data) is a classical task in the field of photogrammetry as well as in computer vision. Dense and accurate depth information is crucial for solving high-level vision tasks. It can be used in the fields of robot navigation, augmented reality, autonomous driving, etc.

Binocular stereo matching (also known as binocular disparity estimation) has been intensively studied as an important method for reconstructing depth information for decades [1]. The principle of binocular stereo matching is similar to the function of human eyes. The key is to find the corresponding matching points in the left and right images taken by the two cameras, to get the disparity, and then use triangulation to calculate the depth information of the image. In the traditional approach [2, 3, 4, 5, 6], matching points in binocular stereo matching are found by matching manually extracted features in stereoscopic image pairs. Despite extensive research, these traditional methods still suffer from low-textured, poor illumination, or non-Lambertian surfaces as well as occlusions. The development of machine learning, especially deep learning (e.g. convolutional neural network) in computer vision has gained great progress in solving various 2D and 3D vision problems. Therefore, stereo matching based on deep learning has received a lot of attention and has gradually become a new direction in the research of reconstructing depth information.

Multitasking frameworks are attracting more and more attention from academia and industry, as the field of computer vision evolves in various directions. The need for perceiving the environment in real-world application scenarios is multifaceted. One of the most important goals of multitasking frameworks is to improve the performance of models by combining multiple tasks. The key to multi-task learning lies in finding and exploiting the relationships between tasks. If the relationships between tasks are established and are used correctly, the different tasks can support each other. If this is not the case, there will be increased noise to the tasks and the performance of models will decrease instead of increasing.

Panoptic segmentation can be seen as an example of multi-task learning. Panoptic segmentation has become a new research direction in recent years and an important topic in computer vision. In computer vision, the task of semantic segmentation is to predict the semantic class of each pixel. And the task of instance segmentation is to predict the pixel region contained in each instance object. Panoptic segmentation can be described as a combination of semantic segmentation and instance segmentation. Panoptic segmentation requires that each pixel in an image must be assigned a semantic

label and an instance ID, where the semantic label refers to the class of the object and the instance ID corresponds to a different number of similar objects.

In recent years, there have been many studies on panoptic segmentation [7, 8, 9, 10, 11, 12], which have achieved exciting results. They also demonstrate the feasibility of multi-task learning. However, at the same time, there are relatively few studies on multi-task learning for reconstructing depth information, especially stereo matching.

As properties of objects in the 3D world, although the physical properties expressed are very different, there is a strong connection between semantics and disparity. This connection includes, but is not limited to, the consistency between semantics and disparity, i.e., abrupt changes in semantics are often accompanied by abrupt changes in disparity; smooth disparity is often accompanied by uniform semantics, e.g., abrupt changes in disparity and semantics between the edge of a car and a distant building in the background, and smooth distribution of disparity and semantics in regions such as buildings and the sky. This connection makes it possible for semantic segmentation and disparity estimation to positively influence each other. In addition, a similar connection can occur for different instances of the same semantics, e.g., different vehicles on a street that are far away from each other. Instance-level information allows the above associations to be established not only at the semantic level, but also at the instance level in more detail. Therefore, the combination of panoptic segmentation with semantic and instance information and disparity estimation is worth investigating in terms of information and attribute properties.

Similar to panoptic segmentation, dense disparity estimation through binocular stereo matching also performs pixel-level prediction of images (each pixel corresponds to a disparity value). With respect to network structure and output results, it is possible to combine panoptic segmentation and disparity estimation. At the same time, according to recent research on disparity estimation, it should be noted that the calculated consumption of disparity estimation networks, especially the models using 3D convolutions represented by PSM [13], is still very large. It would be a valuable advancement if the panoptic segmentation branch could be used to introduce additional information to replace the computationally intensive part of the disparity branch in order to achieve the same or higher accuracy of prediction with fewer or the same total amount of computation. Research into multi-task learning based on panoptic segmentation and disparity estimation is therefore of great interest.

1.1. Problem Statement

Weak textures and variations in light intensity have always posed a great challenge to binocular stereo matching algorithms. Pure disparity computation is difficult to break through these difficulties due to the limitation of the dimensionality of the information it utilizes. As properties of things in the 3D world, there is a strong connection between

semantics and disparity, although the physical properties expressed are very different. This connection includes, but is not limited to, the consistency between semantics and disparity, i.e., abrupt changes in semantics are often accompanied by abrupt changes in disparity; smooth disparity is often accompanied by uniform semantics. Moreover, the instance-level information allows the above connections to be made not only at the semantic level, but also at the instance level in more detail. Therefore, combining panoptic segmentation and disparity estimation to improve the prediction accuracy could be a feasible solution.

The objective of this thesis is to develop a methodology that combines dense stereo matching and panoptic segmentation. More precisely, the tasks of dense stereo matching and panoptic segmentation are to be fused into a consistent approach, operating on stereo image pairs which are used as input data. The expected outcome of the method to be developed is a disparity estimate, a semantic label and an instance label for every pixel of a reference image. This work focuses on improving the disparity estimation results by using panoptic branch while ensuring the accuracy of panoptic segmentation. The main questions we are tackling can be listed as follows:

1. How to design the network so that end-to-end panoptic segmentation and stereo matching can be performed simultaneously?
2. How to fuse the features of the panoptic branch and the disparity branch?

1.2. Contributions

In the present work, we propose a method that can simultaneously perform disparity estimation and panoptic segmentation based on the task requirement of jointly estimate geometry and semantic. The method is proposed by taking the advantages of Panoptic-DeepLab, PSM and AnyNet work. The design of each part of the method is not only theoretically sound but also unified, lightweight and well-structured in its entirety. Experimentally, we also demonstrate the feasibility of the approach in terms of task requirements and structural compatibility with multiple tasks. This makes the method a strong reference and a starting point for subsequent research. On this basis, we designed two feature fusion modules, AFM and WFM, and further investigated the use of panoptic information to improve the disparity estimation by feature fusion. In addition, through experiments on training strategies, datasets, and structural compatibility, we have obtained a lot of first-hand information on the above topic, which will guide the direction of the subsequent research.

1.3. Thesis Outline

The basics of this method, such as deep learning, dense binocular stereo matching and panoptic segmentation are introduced in Chapter 2, where the smooth L1 loss, residual structure, AnyNet and Panoptic-DeepLab are highlighted. Recent studies related to it are reviewed in Chapter 3. The method is described in Chapter 4. The experimental setup and evaluation metrics can be found in Chapter 5. In Chapter 6, the experimental results are presented and analyzed. A conclusion of this work and some suggestions for further work are given in Chapter 7.

2. Fundamentals

The concepts presented in this thesis correspond to the field of joint binocular stereo matching and multi-task learning with panoptic segmentation and are based on techniques from the deep learning domain. In this chapter, the basic concepts of deep learning are first reviewed in Section 2.1. Among them, the idea of CNNs, smooth L1 loss and ResNet are introduced in more detail as application areas related to the present work. In Section 2.2, the basic definition of binocular stereo matching and related methods are briefly introduced. Finally, panoptic segmentation is briefly introduced in Section 2.3.

2.1. Deep Learning

Deep learning (DL) is a research direction in the field of machine learning (ML). It was introduced into Machine Learning to bring it closer to its original goal - artificial intelligence (AI). "Learning" involves learning the intrinsic laws and hierarchical features of data. Deep learning is called "deep" in contrast to other "shallow learning" methods in machine learning such as support vector machine (SVM), boosting, and so on. These "shallow learning" methods rely on manually designed or extracted features. These models are single layer features without hierarchy. Deep learning, on the other hand, can automatically find patterns and features in the data by performing multi-level nonlinear transformations on the original data. At the same time, because it is "data-driven", it is able to extract more effective patterns and features from complex and massive data than manual design. After years of research, deep learning has achieved many results in image classification, data mining, machine translation, natural language processing, recommendation and personalization techniques, and other related fields.

Convolutional neural networks (CNNs) have proven to be a very effective deep learning method in image recognition and classification. Generally, CNNs operate in a similar way to other methods that are also from artificial neural networks (ANNs), such as recurrent neural networks (RNNs) and generative adversarial networks (GANs). CNNs also perform inference (prediction) by forward propagation. The parameters of the model are updated by data-driven back propagation. One of the most important advantages of CNNs is weight sharing. A convolutional layer can have several different convolutional kernels. Each convolutional kernel corresponds to a feature map after filtering. And each pixel in the same feature map comes from the exact same convolutional kernel, which is the weight sharing. It can reduce the training parameters of a network and

make the neural network structure simpler and more adaptive.

Smooth L1

Smooth L1 is the loss function used in many deep learning based stereo matching algorithms. There are many loss functions in deep learning, such as L1 loss and L2 loss. Smooth L1 can be considered as a smoothed version of L1 loss and can be described by Equation 2.1.

$$SmoothL_1(x) = \begin{cases} 0.5x^2 & if|x|<1 \\ |x| - 0.5 & otherwise \end{cases} \tag{2.1}$$

where x=$f(x_i)$ - y_i is the difference between the true value and the predicted value.

Smooth L1 is actually a segmentation function, which is actually L2 loss between [-1,1]. This solves the problem that the derivative of L1 loss is not unique at point 0. Outside the interval [-1,1], it is actually the L1 loss, which solves the problem of outlier sensitivity and gradient explosion. In general, Smooth L1 can converge faster compared to L1 loss. Compared with the L2 loss function, Smooth L1 is not sensitive to outliers.

ResNet

ResNet (deep residual network) [14] is currently one of the most popular network structures to be used as encoders. The main contribution of ResNets is to solve the "degradation" problem of deep neural networks. The authors found that the degradation occurs as the number of layers in a network increase. Degradation refers to the situation where the performance of a network degrades rapidly when more layers are added to it. One reason for this phenomenon is the irreversible loss of information caused by nonlinear activation functions, such as Relu. The ability of identity mapping is lost in this case. This is not overfitting, because in overfitting the training loss always decreases. In the case of degradation, the loss of the training set increases instead. Therefore, the authors start from the perspective of model structure to find a better model structure. They have proposed that residual learning in ResNet solve the degradation problem. The structure that implements this concept of residual learning is the residual block (Figure 2.1).

The basic unit of the ResNet is the residual block. Assume that the input is x and the ground truth is H(x). In short, the residual block makes the network's prediction F(x) no longer converge to H(x) directly, but fit the difference between H(x) and x, i.e., F(x):H(x)-x. It is easier to let F(x) learn to be 0 than to let F(x) learn to be a constant mapping to ground truth [14]. Thus, the constant mapping can be obtained by simply F(x)→0.

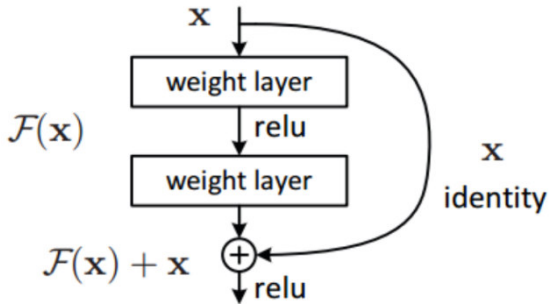


Figure 2.1.: Residual block in ResNet [14]. The input x is connected to the last part of the block as it passes through the layer. This residual connection allows the result of layer processing F(x) and x to be added together to become the output of the block.

2.2. Binocular Stereo Matching

This work has a strong relevance to binocular stereo matching. The rest of this chapter reviews the basic concepts and principles of binocular stereo matching (hereinafter referred to as stereo matching).

The principle of stereo matching is similar to that of human eyes. The human eyes have left and right positions, so the two eyes do not see exactly the same image for the same scene at the same moment. More specifically, for the same object in the scene, the absolute positions in the image perceived by the left and right eyes are different (as shown in Figure 2.2). The closer the object is to the observer, the greater the difference in the absolute positions of the object seen by the two eyes in the horizontal direction. This difference is called disparity. The brain can process disparity to determine the distance between an object and us (i.e., depth information). In a similar way, two calibrated cameras can obtain disparity. It can be expressed as d = x_L - x_R, where x_L and x_R are the positions of the same thing in the scene in the two images. The depth information can be reconstructed from the image by establishing a mathematical relationship between disparity and depth. This relationship is shown in Figure 2.3 and Equation 2.2, where b is the distance between the left and right cameras, and f is the focal length of the camera.

$$Z = \frac{b \times f}{X_R - X_L} = \frac{b \times f}{d} \tag{2.2}$$

The basic process of binocular stereo matching is shown in the Figure 2.4. As mentioned before, one of the important steps in stereo matching is to find the correspondence between the contents of the two images. As shown in Figure 2.5 (b), the point P in the scene is projected on the left and right images as p' and p". A typical method for finding content correspondence in stereo pairs is point-by-point matching [1], where the pixels of

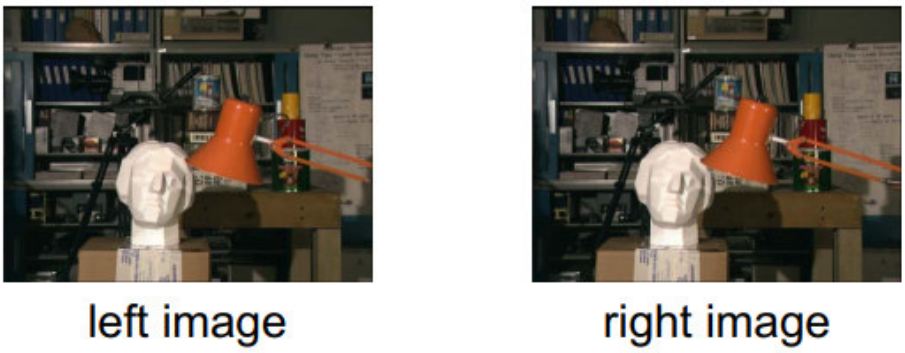


Figure 2.2.: An example of a stereo image pair [2]. It can be seen that the absolute positions of the same objects in the scene are different in the horizontal positions of the left and right images. This difference in position becomes more pronounced with increased disparity to the observation point (e.g., the statue). This difference in position is called disparity.

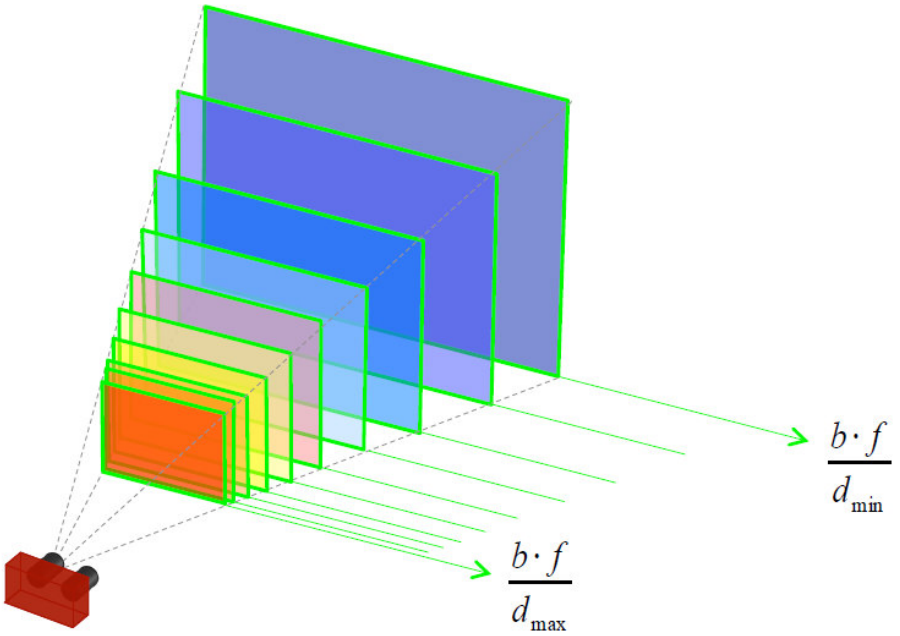


Figure 2.3.: Visualization of the relationship between disparity and depth [15]. Depth is inversely proportional to disparity, and the maximum disparity corresponds to the minimum depth, i.e., the closest distance to the observation point.

the two images are compared point by point. Since there is no direct constraint between the coordinates of the two cameras (o' and o") and the imaging plane (L and R), all pixels in the right image must be considered in order to find the correspondence of p'

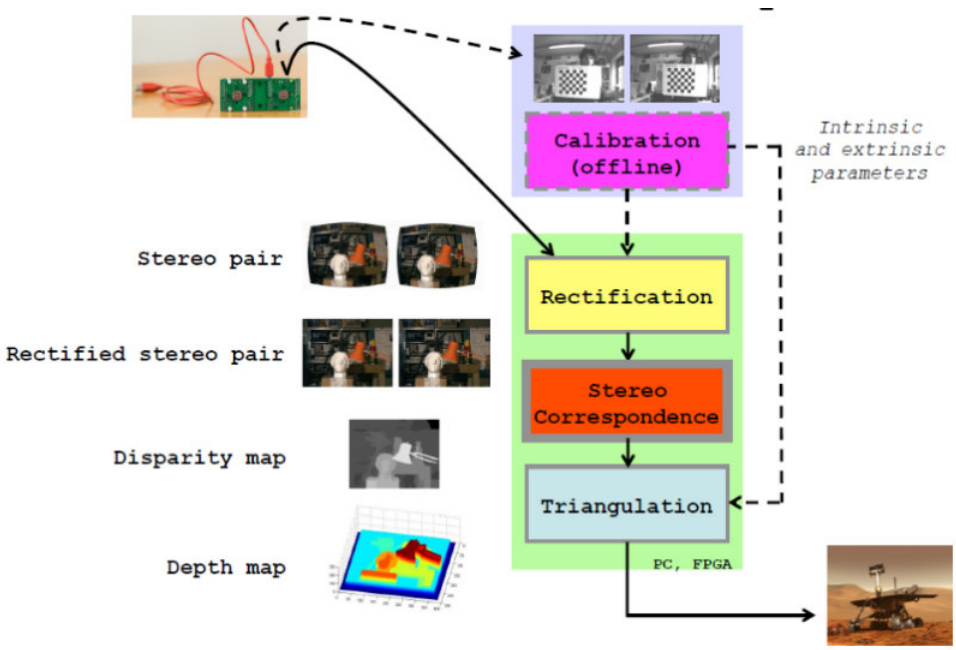
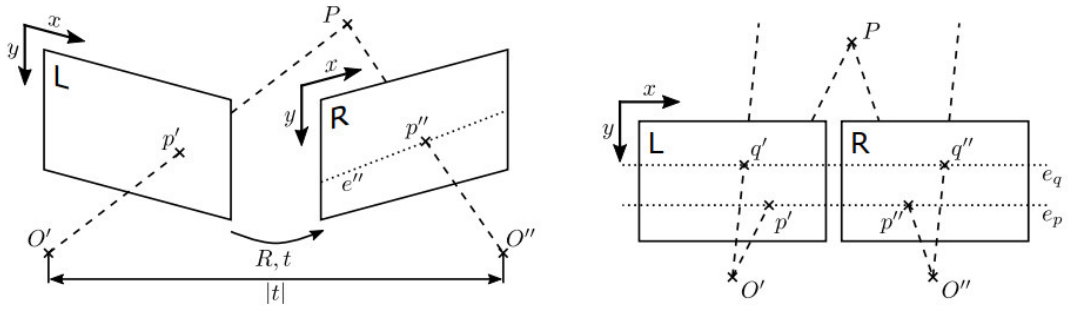


Figure 2.4.: Basic process of depth information reconstruction based on binocular stereo image pairs [16]. The calibration part is used to obtain the parameters of the camera. These parameters are used in the final conversion of the disparity map to depth map. The rectification part aligns the stereo image pairs and feeds them into the next part (corresponds to the red module in the figure, which is also the relevant part of this work).

in that image. This process is computationaly intensive. The two image planes can be aligned on the same plane by rectification. In this way, the search space can be reduced from the entire image to an epipolar line, thus simplifying the matching task from 2D to 1D [1]. Also, errors caused by ambiguities can be reduced by adding constraints to the search space.



(a) Epipolar geometry (b) Planar rectification

Figure 2.5.: Rectification of stereo image pairs [1].

Stereo matching is an important part of the reconstruction of depth information based on stereo image pairs, which is also a relevant part of this work. The purpose of this part is to find the correspondence between the contents of two images and to generate a disparity map. Traditional stereo matching methods can be classified into three categories: local matching, semi-global matching and global matching. Each of these methods can be divided into four steps: matching cost, cost aggregation, disparity calculation and disparity refinement, with emphasis on the first two steps. Since this work uses a deep learning-based approach, traditional stereo matching methods are not described in detail here. Similar to traditional methods, some deep learning-based stereo matching methods can be divided into the four steps mentioned above. The features of left and right images are first extracted in backbone (e.g., ResNet mentioned in Section 2.1). During the subsequent processing, the cost volume is generated according to the author's design. The cost volume is then subjected to a series of subsequent convolution operations. Finally, the features map is obtained.

AnyNet

AnyNet [17]is used as a basis for the approach developed in the context of this thesis. The structure of AnyNet is shown in Figure 2.6. In AnyNet, stereo image pairs are first extracted in the backbone with a U-shaped structure (Figure 2.7). The features are fed into three subsequent branches depending on the resolution of the feature map (1/16, 1/8, 1/4 of the original resolution). In these three branches, the feature map is processed by the disparity network (Figure 2.8) to obtain the disparity map of the three stages. This map is refined by the residual structure (see Section 2.1 for details) step by step. This is achieved by warping the input feature maps of this stage using the predicted disparity maps of the previous stage. In this case, the disparity map obtained by the disparity network no longer contains the "absolute value" of disparity, but the residuals.

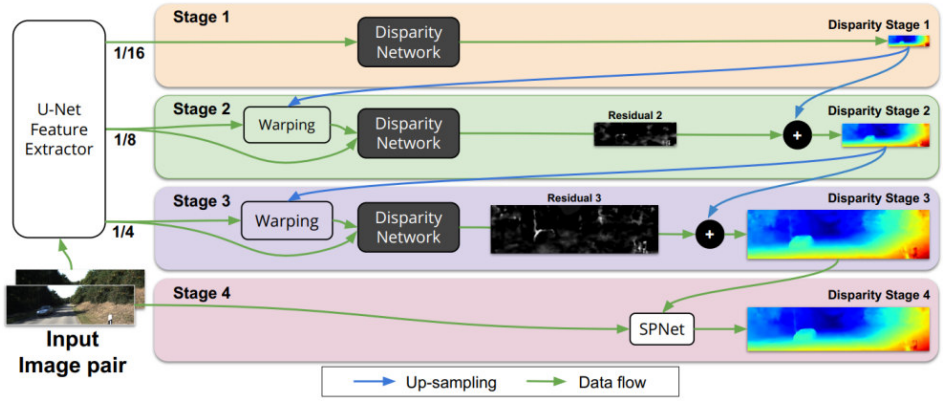


Figure 2.6.: AnyNet's Network Architecture [17].

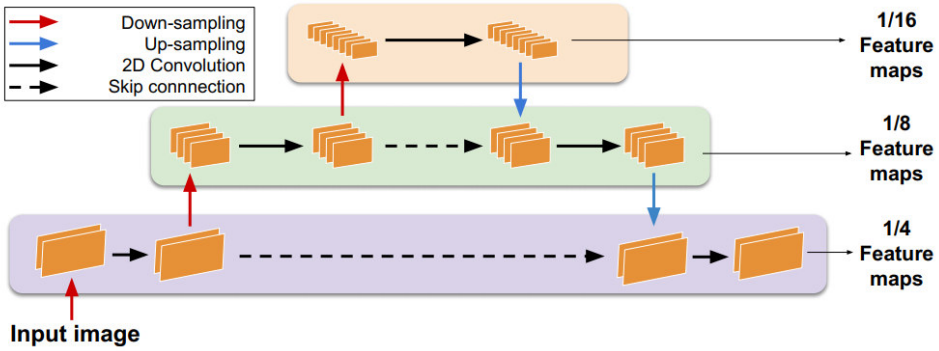


Figure 2.7.: Detailed structure of AnyNet's backbone [17].

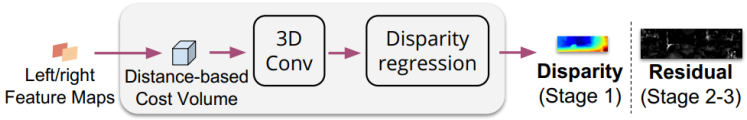


Figure 2.8.: Detailed structure of disparity Network [17].

2.3. Panoptic Segmentation

Panoptic segmentation has become a new research direction in recent years and an important research topic in computer vision. In computer vision, the task of semantic segmentation is to predict the semantic classification of each pixel. Instance segmentation is used to predict every pixel area that is included in each instance. Panoptic segmentation, which requires one semantic label and one instance ID in each pixel of an image, can be described as the combination of semantic segmentation and instance segmentation. In panoptic segmentation, the semantic label refers to the object type and the instance ID corresponds to a different number of similar objects.

Panoptic-DeepLab [7] is a panoptic segmentation model proposed in 2020. It achieves the goal of panoptic segmentation by combining semantic prediction, instance center prediction and instance center regression. The structure of Panoptic-DeepLab consists of four main components, as shown in Figure 2.10. From left to right: Backbone for semantic segmentation and instance segmentation, ASPP (atrous spatial pyramid pooling), decoder module for individual tasks and head for specific tasks. The backbone part uses ResNet. The ASPP module is used to extract multi-scale contexts. The decoder was designed and modified based on DeepLab V3 [18]. The structures for both semantic and instance segmentation are identical and the head part is an FCN (fully convolutional networks). At the end of the network, semantic prediction, instance center prediction and instance center regression are fused to generate the final panoptic segmentation result by the "majority vote" proposed by DeepLab [19].

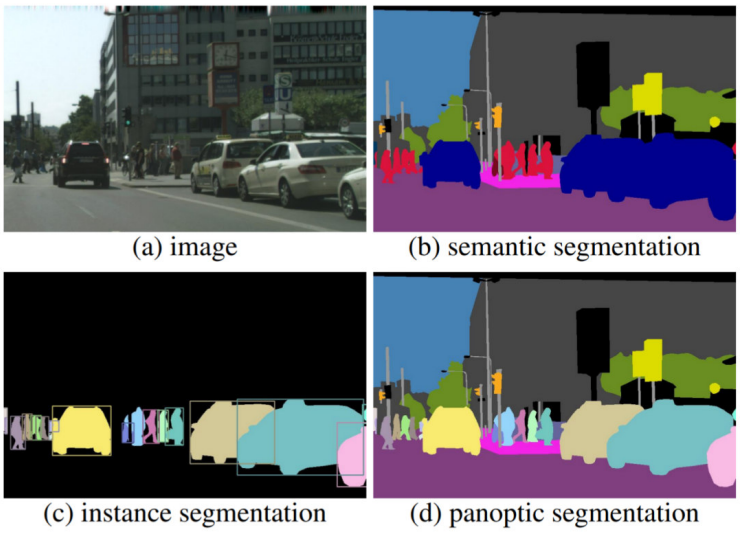


Figure 2.9.: The similarities and differences of semantic-, instance- and panoptic segmentation [7].

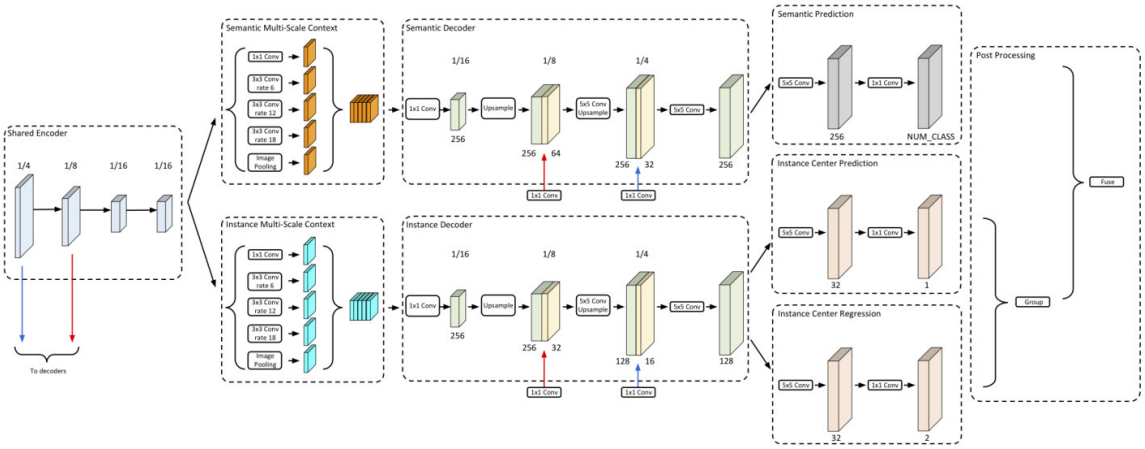


Figure 2.10.: The structure of Panoptic-DeepLab [7]. (See appendix for a larger image)

3. Related Works

In this chapter, the related works relevant for this work are reviewed. Since this work is a joint network of dense stereo matching and panoptic segmentation, studies related to dense stereo matching and panoptic segmentation are first reviewed in sections 3.1 and 3.2. In section 3.3, joint estimation geometry and semantic related to this work are presented.

3.1. Dense Stereo Matching

Recent studies on dense stereo matching are reviewed in this section. Traditional hand-crafted approaches are not discussed since the methodology presented in this work solely focuses on the development of deep learning-based approaches. These following approaches will be reviewed in turn:

GC-Net [20] is a representative method for end-to-end disparity prediction. It has had a profound impact on many subsequent studies. The features of input stereo image pairs are first extracted by 2D convolutions in backbone. These features are used to generate the cost volume. The cost volume is obtained by concatenating the left and the right feature maps of the corresponding channel in the disparity direction pixel-by-pixel. For a stereo image pair with height and width H and W, the dimension of the cost volume is H×W×D_{max}×C, where D_{max} is the preset maximum disparity and C is the number of channels in the feature map. The cost volume is then processed by 3D convolutions. The authors of GC-Net argue that a number of challenges faced by stereo matching algorithms can be improved by obtaining global semantic information instead of relying only on local geometric information. For example, for a reflective surface such as a car windshield, a stereo matching algorithm that relies only on the local representation of this reflective surface to compute geometric features is likely to be wrong. However, if the semantic information of this surface (which is part of the car) is understood, it is advantageous to infer the local geometric features [21]. Therefore, 3D convolutions are used in processing cost volume to obtain more contextual information in this network. Finally, GC-Net applies a soft argmax operation to the cost volume, which converts the matching cost into a normalized probability volume. By using the probability values as weights, the disparity map can be obtained by a weighted average along the disparity direction. The main contribution of GC-Net is that it does not use the traditional cost aggregation → disparity calculation → disparity optimization methods, but uses

a regression approach to predict disparity values end-to-end. Also, this deep learning-based feature extraction → cost volume → cost aggregation → disparity estimation process has had a profound impact on subsequent research. However, although GC-Net is trying to use global information, being less "local" does not mean global. GC-Net's "no-local" still has a maximum distance limit. In addition, GC-Net makes very little use of semantic information. In fact, GC-Net only considers geometric information. Semantics, i.e. class labels, are not included in the training data. Therefore, these semantics are not learned by the network.

The authors of PSM-Net [13] and GA-Net [22] basically followed the ideas of feature extraction and cost volume generation in GC-Net but modified the cost aggregation step. The authors of PSM-Net believe that there is still the potential for more utilization of contextual information. The main innovation of PSM-Net is the introduction of the SPP (spatial pyramid pooling) module and the stacked hourglass module. The SPP module uses adaptive mean pooling to compress the feature maps to 4 scales of 64x64, 32x32, 16x16 and 8x8. These feature maps are then convolved by 1x1 layers to reduce the dimension and are then upsampled by bilinear interpolation to recover the original image size. Finally, the feature maps of different levels are combined into a final feature map. The advantage of this is that the receptive field is expanded and the multi-scale features, which allows a better combination of local and global information. The stacked hourglass consists of several iterative coarse-to-fine and fine-to-coarse processes with intermediate layers of supervision. The feature maps are compressed and upsampled several times (like an hourglass) to make better use of contextual information. Although the accuracy of PSM-Net surpasses that of GC-Net, its number of parameters is nearly twice that of GC-Net.

GA-Net also proposes a new idea based on GC-Net, but does not require as great a number of parameters as PSM-Net. The authors of GA-Net propose two modules, SGA (semi-global guided aggregation) and LGA (local guided aggregation), to replace some of the 3D convolutional layers in GC-Net and combine them with the remaining 3D convolutional layers to form a cost aggregation module. By this scheme, the number of parameters of the model can be greatly reduced while the accuracy is guaranteed. In addition, the computational complexity of SGA and LGA is only 1% that of 3D convolutional layers according to the results of experiments [22]. Therefore, the inference speed of GA-Net is also improved. Similar works to reduce the computational effort of cost regression is also done by AA-Net [23]. However, the challenges posed by weak textures and light intensity for binocular stereo matching are still not well solved.

GWC-Net [24] and Cascade-MVS [25] have made a new attempt from the aspect of cost volume. A combined cost volume is proposed in GWC-Net, which consists of a concat volume and a group correlation volume. Concat volume is obtained directly by cascading left and right feature maps. The group correlation volume is obtained by dividing the unary features into many groups and calculating their volumes individually. The parameters can be reduced by the group correlation. The authors show experimentally that this concat volume and group correlation volume are consistent with each other.

Similarly, Cascade-MVS also tries to cascade multiple cost volumes and can obtain cost volumes of different scales based on the pyramidal encoder-decoder structure. The authors argue that the processing of cost volumes of different stages (scales) should not be completely fixed and identical. The cost volumes of the later stages should be able to narrow down the depth value or variability by the predicted results of the previous stage [25], i.e. the later cost volume has less depth/disparity hypothesis, which will reduce the computational effort.

From the above work, we can see that reducing the number of parameters and the computational effort of the model (reducing the hardware requirements of the model and increasing the inference speed) is a research priority as much as improving the accuracy. Although the PSM-Net algorithm has achieved high accuracy, its huge number of parameters and computational complexity have hindered its development in industry and academia. In addition to the above examples, a lot of work has been carried out in recent years [26, 27, 28, 29], but the hardware requirements of the current state-of-the-art algorithms are still very large compared to other computer vision fields (segmentation, target recognition, etc.). Therefore, it is still very important to reduce the number of parameters and the computational effort of the model while maintaining accuracy.

In addition, it is still difficult for the current method to make a breakthrough in weak textures and other difficult points. One important reason is that the information utilized by the above pure disparity calculation methods is very single and limited. As mentioned in GC-Net, semantic and disparity information are strongly correlated. The instance-level information allows the above correlation to be established not only at the semantic level, but also at the instance level in a more detailed way. Therefore, it is a promising direction to try to combine disparity estimation with semantic information, or even panoptic information (with additional instance-level information).

3.2. Panoptic Segmentation

Panoptic segmentation is a new computer vision task that has emerged in recent years. Since the concept of panoptic segmentation was introduced in 2019 by Kirillov, Alexander and He, Kaiming [7], there have been many excellent related works. These works can be broadly classified into two categories: top-down (instances detection first and then semantic segmentation) and bottom-up (semantic segmentation first and then instance generation). In this section, we briefly introduce these two ideas.

Panoptic-FPN [30] is an example of a top-down method. The current idea of top-down approaches is to add a semantic segmentation branch to the top-down based instance segmentation network (e.g., Mask-RCNN), i.e. these approaches are often based on object detection. In simple terms, Panoptic-FPN is Mask-RCNN with FPN. The input images are first extracted by a backbone with a pyramid structure. Then the feature maps are fed into the instance branch. The operations of this step are the same as those

in Mask-RCNN. The semantic segmentation also uses the feature maps of the pyramid structure. The final panoptic predictions are achieved by post-processing the results of instance segmentation and semantic segmentation, for example resolving overlaps between different instances based on their confidence scores and resolving overlaps between instance and semantic segmentation outputs in favor of instances [30]. The main drawback of the top-down methods is the long inference time, which is especially affected by the instance segmentation branch.

Panoptic-DeepLab is an example of a bottom-up method. The bottom-up methods also use a multi-branch structure similar to the top-down methods. But the main difference is that the bottom-up methods usually begin with performing semantic segmentation at the pixel level. Then the semantic segmentation results are used to distinguish different instances by clustering and metric learning. Finally, the semantic predictions and instance predictions are post-processed to generate the panoptic predictions. Panoptic-DeepLab was inspired by DeepLab [19]. The details of the algorithm are described in section 2.3. The complex object detection step similar to that in Mask-RCNN is not used, which results in the bottom-up methods having a fast inference speed. However, at the same time, they tend to have lower accuracy.

Although top-down methods still have the advantage of accuracy, the development and progress of bottom-up methods are also evident. We have reasons to believe that bottom-up based panoptic segmentation has great potential and could produce exciting results, just as one-stage methods [31, 32, 33] in object detection and bottom-up methods in instance segmentation have done in recent years. In addition, the design of bottom-up methods starting with semantic segmentation is also more suitable for the processing of disparity estimation, which makes it structurally easier to combine with disparity estimation to form a multi-task structure.

3.3. Joint Estimation of Geometry and Semantic

Although there are many solutions for the joint estimation of geometry and semantic [34, 35, 36] , there are very few studies based on panoptic segmentation and binocular disparity estimation. There is consistency between semantic and disparity. For example, there is a sudden change in disparity and semantics between the edge of the car and the distant building as background and a smooth distribution of disparity and semantics in regions such as building and sky. This connection makes it possible for semantic segmentation and disparity estimation to positively influence each other. In addition, a similar connection can occur for different instances of the same semantics, e.g., different vehicles on a street that are far away from each other. Instance information allows the above associations to be established not only at the semantic level, but also at the instance level in a more detailed way. Therefore, the combination of panoptic segmentation and disparity estimation is worth investigating. This section only reviews the joint estimation of geometry and semantic based on binocular disparity estimation.

Most of the current studies on the joint estimation of geometry and semantic from binocular stereo images are based on semantic segmentation. SegStereo [34] guides the disparity estimation at the semantic level by embedding semantic feature maps into cost volumes. SegStereo first generates cost volumes by correlation. It then aggregates the left semantic segmentation maps obtained by the semantic branch with the cost volume. The aggregated concat-volume is then fed into an encoder-decoder with deconvolutions and output to obtain a disparity map. Meanwhile, the right semantic map is warped to the left view for per-pixel semantic prediction with softmax loss regularization and compared with the disparity map. The loss is then calculated. Both of these steps attempt to bootstrap the disparity prediction using semantic information. According to the results of experiments, the method is effective in improving the estimation results for local areas with insignificant texture. However, a limitation of the method is that the semantic segmentation results must be very accurate. Otherwise, the disparity estimation results will not be improved or will even become worse after being guided by semantic information.

SSPCV-Net [35] exploits the multi-scale spatial information by combining semantic cost volumes with multi-scale pyramid cost volumes. The semantic features of the left and right images at different disparity levels can be cascaded to obtain the semantic cost volume. At the same time, the pyramid cost volumes can be generated by using the disparity maps with different scales generated by the decoder. The authors obtained a total of four cost volumes, including a semantic cost volume with a resolution of 1/4 and three disparity cost volumes (pyramid cost volumes) with resolutions of 1/4, 1/8 and 1/16. Afterwards, these cost volumes are processed by their respective hourglass modules and fused by FFM (3D feature fusion module) in the order of lower to higher resolutions. The remaining cost volume is finally processed by an additional hourglass module to generate disparity predictions. In addition, SSPCV-Net also proposes a gradient-related loss to constrain the disparity estimation using semantic information. This loss assumes that the disparity should be flat for the same semantic region. The use of multiscale information and gradient-related loss allowed SSPCV-Net to become the state-of-the-art method at the time of publication. However, it should be noted that the number of parameters in this network is very large due to the extensive use of hourglass modules.

SG-Net [37] based on PSM-Net uses the confidence module and the residual module to fuse semantic and disparity information while using gradient-related loss similar to that of SSPCV-Net. The confidence module fuses semantic cost volume and disparity cost volume by a multiplication operation. It tries to refine the cost volume at the level of probability. The result of the fusion is further computed by 3D convolutions and directly linked to the result of the multiplication operation through the residual structure. The output of the confidence module is fed into the residual module along with the output of the semantic branch. The idea of the residual module is similar to SegStereo. The authors of SG-Net believe that the disparity in different semantic regions should be different. In addition, the disparity in the same semantic region should be similar. The residual module can use the prediction results of the semantic branch to guide the

disparity. Finally, the outputs of the residual module are corrected by the gradient-related loss. Since SG-Net is based on PSM-Net, its number of parameters is still very large. In addition, the multiplication operation in the confidence module, although concisely fusing information from semantic branches at the level of probability, also means that the fusion results are more likely to be influenced by semantic information. Both the correct part and incorrect part of the disparity cost volume may be amplified by multiplication. The experiments also demonstrate that the improvement of performance is mainly due to gradient-related loss.

Currently, there are few studies on joint estimation of geometry and semantic from binocular stereo images based on panoptic segmentation. PG-Net [38] is based on the same ideas as SG-Net and is made by the same team. PG-Net also uses the multiplication operation to fuse information from semantic, instance and disparity branches. The disparity maps are guided by the semantic and instance information in a similar residual module. The authors show experimentally that PG-Net can achieve higher accuracy than SG-Net. However, the previously mentioned drawbacks of SG-Net still exist. Although VIP-Deeplab [35] uses panoptic information to assist disparity prediction, it is based on monocular disparity prediction.

Through the previous studies we can see that the fusion of multi-branch features and the design of loss functions are the key to joint estimation of geometry and semantic. All of the above works have demonstrated that semantic information can be used to improve the accuracy of disparity estimation. In addition, PG-Net attempt of panoptic segmentation has also proved its feasibility to be combined with disparity estimation. As mentioned many times before, the current number of parameters of disparity estimation networks is still very large. Accuracy improvement should not be directly achieved at the cost of stacking the complexity and number of parameters of the network. Therefore, using multi-branch information in a multi-task framework, such as joint estimation of geometry and semantic in this work, to achieve higher or the same accuracy with the same or fewer number of parameters will be an important topic for future investigations. The use of semantic and panoptic information in the current studies is still inadequate First, although the semantic or panoptic information in the above works is involved in the estimation of disparity, it comes only from the final output stage. The semantic or panoptic information at this stage is often already the final outputs of the corresponding branch. In many network structures, the jump connections between encoder-decoder make the differences between feature maps of adjacent stages not only limited to the differences in resolution (e.g., caused by upsampling). The information contained between them may also be completely different. However, the information of these intra-branch feature maps is not well exploited by disparity prediction in the above works. Second, the current fusion methods for information from multi-branch are "predefined", such as the multiplication operation used in SG-Net for fusing cost volumes. These "predefined" approaches limit the possibility of the fusion of the multi-branch information and are not fully compatible with the original intention of "deep learning". Therefore, further research on the fusion of multi-branch information is still necessary.

Previous work has only focused on the improvement of disparity estimation using semantic or panoptic information and the results of semantic or panoptic segmentation have not been evaluated. Given the existence of a shared backbone component and the intervention of the semantic or panoptic branch in the disparity branch, it is reasonable to believe that the semantic or panoptic branch in this semi-dependent multi-branch structure is also influenced by the backpropagation. Is this influence positive or negative? To what extent does it reach? And is it possible to find new ways to make multi-branch mutually reinforcing? The answers to these questions need to be explored and studied.

4. Methodology

This chapter introduces the method proposed in this work. The conditions of application and the general structure of the method are presented in section 4.1. The other sections present the details of the components that make up the method, the motivation and the related variants.

4.1. Overview

The goal of the proposed method in this work is joint estimation of semantic map, instance map and disparity map, for which we consider both panoptic segmentation and disparity estimation. We focus on how to use semantic and instance information (panoptic information) to improve the quality of disparity estimation by means of feature fusion.

The input to this method is a binocular stereo image pair (left image, right image), as shown in Figure 2.2, where the left image is specified as the reference image. They are assumed to be taken simultaneously and partially overlapped so that the depth of the image can be determined according to the process shown in Figure 2.4. In addition, it is assumed that the calibration and planar rectification of the binocular stereo image pair are performed, so that the interior orientations of both cameras and the relative orientation between them is known and the epipolar lines that coincide with the image rows are resulted [1]. Under the above premise, the disparity of the pixels in the reference image (left image) is computable in the right image and the search range is only on the epipolar lines with the same y coordinate, as shown in Figure 2.5. The final output of the method is the panoptic segmentation and disparity estimation of the left image.

The structure of this method is shown in Figure 4.1. This method still follows the workflow of feature extraction $\rightarrow$ cost volume $\rightarrow$ disparity regression. In the feature extraction part, we refer to the network structure of Panoptic-DeepLab [8] and ViP-DeepLab [35] and use the same method in Panoptic-DeepLab to generate the panoptic segmentation. The encoder and decoder of semantic, instance and disparity branches are shared for the left and right input images. In addition, the three branches also share the same encoder for feature extraction. By extracting three different stages of feature maps with different resolutions in each branch, a pyramid cost volume can be generated, i.e. the semantic, instance and disparity pyramid DRA cost volume. These three cost volumes are fused into the fusion pyramid DRA cost volume by the AFM (Assign Fusion

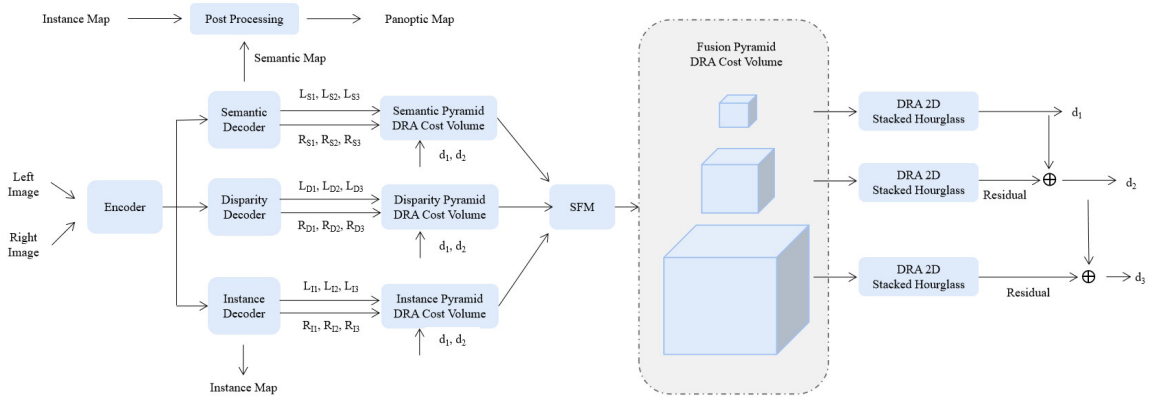


Figure 4.1.: The AA (all adapt) with AFM of the proposed method in this thesis. The features of left and right images are extracted by the shared encoder and decoders. L and R represent the features of the left image and the right image. The letters in their footnotes represent the branch they come from, and the numbers in the footnotes represent the stage they were extracted from. The three branches can generate a total of three pyramid cost volumes. These cost volumes can be fused into a new pyramid cost volume by the fusion module. The disparity prediction d of the network can be generated by the disparity regression of the cost volume. (See appendix for a larger image)

Module). The three cost volumes in the fusion pyramid DRA cost volume are fed into the respective DRA 2D Stacked Hourglass in the order of lower to higher resolutions. Note that the disparity maps of higher stages are obtained based on that of lower stages. In addition, disparity maps of stages 1 and 2 are involved through warping in the generation of the semantic, instance and disparity pyramid DRA cost volumes. These form a coarse to fine residual structure for disparity estimation. The remaining sections of this cheaper describe these parts in more detail.

4.2. Feature Extraction and Panoptic Segmentation

In this method, the design of feature extraction is more about how to combine multi-task more deeply. We refer to ViP-DeepLab, use the structure of Panoptic-DeepLab in semantic and instance branch and perform panoptic segmentation. And the decoder in the disparity branch uses a similar structure to the semantic decoder. For a detailed description of panoptic segmentation, see Panoptic-DeepLab [8].

A multi-task method should not be a simple combination and splicing of several individual methods. For this reason, the encoder and decoder of the semantic, instance and disparity branches of the left and right input images are shared. In addition, the three

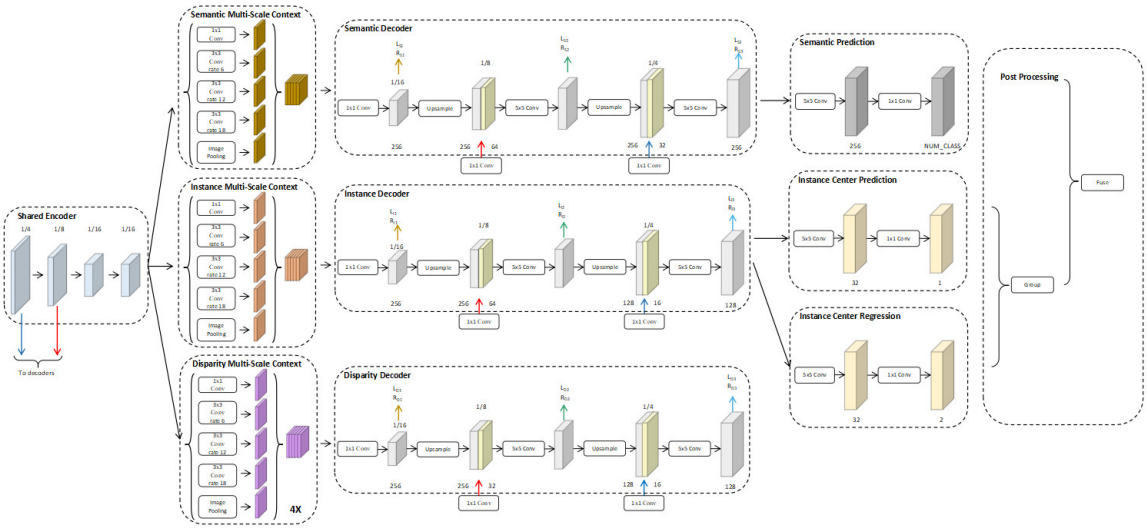


Figure 4.2.: The structure of panoptic branch and encoder-decoder for each branch. The panoptic process is identical to the Panoptic-DeepLab. The decoder of the disparity branch is consistent with the semantic branch. (See appendix for a larger image)

branches mentioned above also share the same encoder for the feature extraction. In contrast, although the encoder of ViP-DeepLab is shared, the encoder does not need to consider the next-frame t+1 (corresponding to the right image of this method) semantic and instance decoder. Besides, the two input images only share the next-frame instance decoder (corresponding to the disparity encoder of this method). And Panoptic-DeepLab as an algorithm for panoptic segmentation only considers the case of a single image input.

For each input binocular stereo image pair, a total of three pairs of feature maps from the decoder at different scales are extracted in each branch. Feature extraction and the generation of panoptic map is shown in Figure 4.2. The binocular stereo image pairs are first fed into the encoder. The encoder used in this paper is ResNet. Before the data flows into the three branches, the output of the encoder goes to the respective ASPP module of each branch and is extracted for multi-scale features. The ASPP output is processed by a 1x1 convolutional layer into a feature map with a fixed depth of 256 and an aspect of 1/16 of the original size of the stereo image pair (hereafter referred to as the original resolution), which is the extracted left and right feature maps (L_{S1}, R_{S1},L_{I1}, R_{I1},L_{D1}, R_{D1}) of the first level. Subsequently, the feature maps are upsampled to 1/8 of the original resolution and combined with the 1/8 original resolution feature maps from the encoder in a cascade manner to form a concat feature map. The concat feature maps are then convolved 5x5. The result of this processing is the second level of left and right feature maps (L_{S2}, R_{S2},L_{I2}, R_{I2},L_{D2}, R_{D2}) . Analogue, the feature maps are up-sampled, combined with the 1/8 original resolution feature maps from the encoder

and processed by 5x5 convolution to form the third level of left and right feature maps (L_{S3}, R_{S3},L_{I3}, R_{I3},L_{D3}, R_{D3}) .

The motivation for extracting these feature maps at different resolutions and stages is to build pyramid cost volumes containing features at different scales and levels. The different levels of features have been shown to be complementary to each other in many works [36, 30, 13]. In addition, the second and third level features in this method are generated with the involvement of convolution and additional information from the encoder. Therefore, it is reasonable to believe that the difference between the information contained in the feature maps of these three levels is not only due to the difference in scale caused by upsampling.

4.3. Assign Fusion Module

The Assign Fusion Module (AFM) allows the method to select and fuse information from different branches in a data-driven manner at the cost volume, which is the most innovative point of this thesis. The motivation for AFM is to give the method the ability to select and use information from multi-branches by itself. Which parts of the information from the multi-branch are more useful? How should this useful information be used? The answers to these two questions should be given by the method to the extent possible by the machine, rather than by direct human design. Excessive intervention from human design can certainly improve the accuracy of the method in some cases, but it also limits the potential of the method, such as the confidence module in PG-Net [38], which has been introduced in section 3.3. Based on the above, AFM is designed to be data-driven to allow the method to find the optimal way to select and fuse multi-branch information on its own.

The processing flow of AFM is shown in Figure 4.3. The three cost volumes on the leftmost side of the figure are the inputs to the AFM. They are from different branches and have the same dimensions (resolution and depth). By setting the unit depth to 1, these cost volumes can be split into a total of *max disparity* individual feature maps along the depth direction, where *max disparity* is the depth of the input cost volume. Since these cost volumes are generated by correlation, this splitting makes the arrangement between these feature maps orderly (in the direction of increasing or decreasing disparity). Under this condition, feature maps from different branches can be cascaded by disparity level, which results in feature maps with total number of *max disparity* and depth of 3. Subsequently, these cascaded feature maps are further extracted by the respective 3x3 convolutional layers. The resolution of the processed feature maps is unchanged. The depth is expanded to a fixed value of 32. This value is set with reference to PSM, GC-Net and other structures. The optimal hyperparameter settings are not further investigated here. After further processing by 1x1 convolutional layers, the depth of these feature maps is restored to 1 and the resolution is kept constant. Finally these feature maps are cascaded in order of disparity level and recombined into

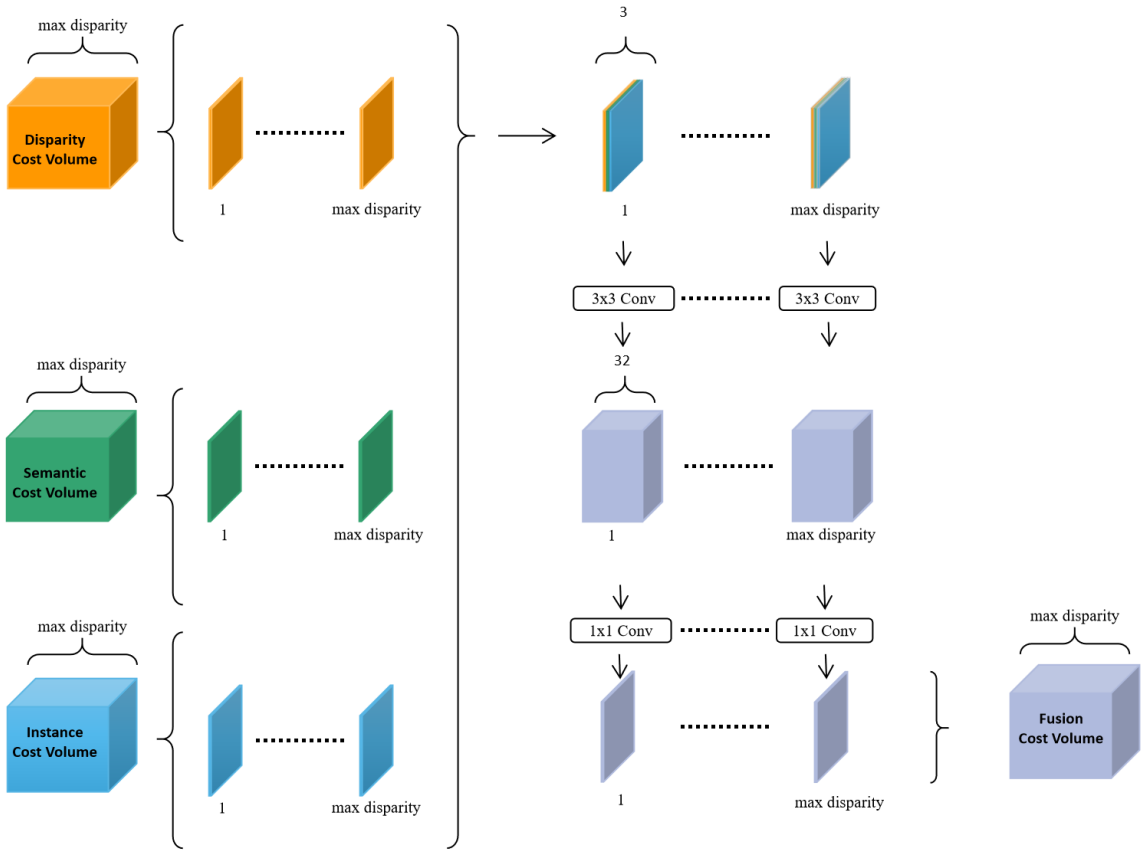


Figure 4.3.: The process of AFM. The cost volume from different branches on the left side can be split into feature maps along the disparity dimension. These feature maps are reorganized into feature maps of depth 3 in the order of disparity levels. After a series of subsequent convolutions, these feature maps are recombined into a fusion cost volume with the same dimension as the input cost volumes.

a cost volume with the same dimension as the input cost volume, i.e. the fusion cost volume.

In this work, we also explore other possibilities of fusing multi-branch information based on the same idea and compare them by experiments. Figure 4.4 shows another feature fusion module similar to AFM, which is called Weighting Fusion Module (WFM). The WFM follows the same concept as mentioned before in a more simple way. In contrast to AFM, the cascaded feature maps in WFM do not undergo a further intermediate step of feature extraction. These cascaded feature maps of depth 3 are directly fused into a feature map of depth 1 using a 3x3 convolution (without activation function) in a weighted summation manner. The rest of the WFM remains consistent with the AFM.

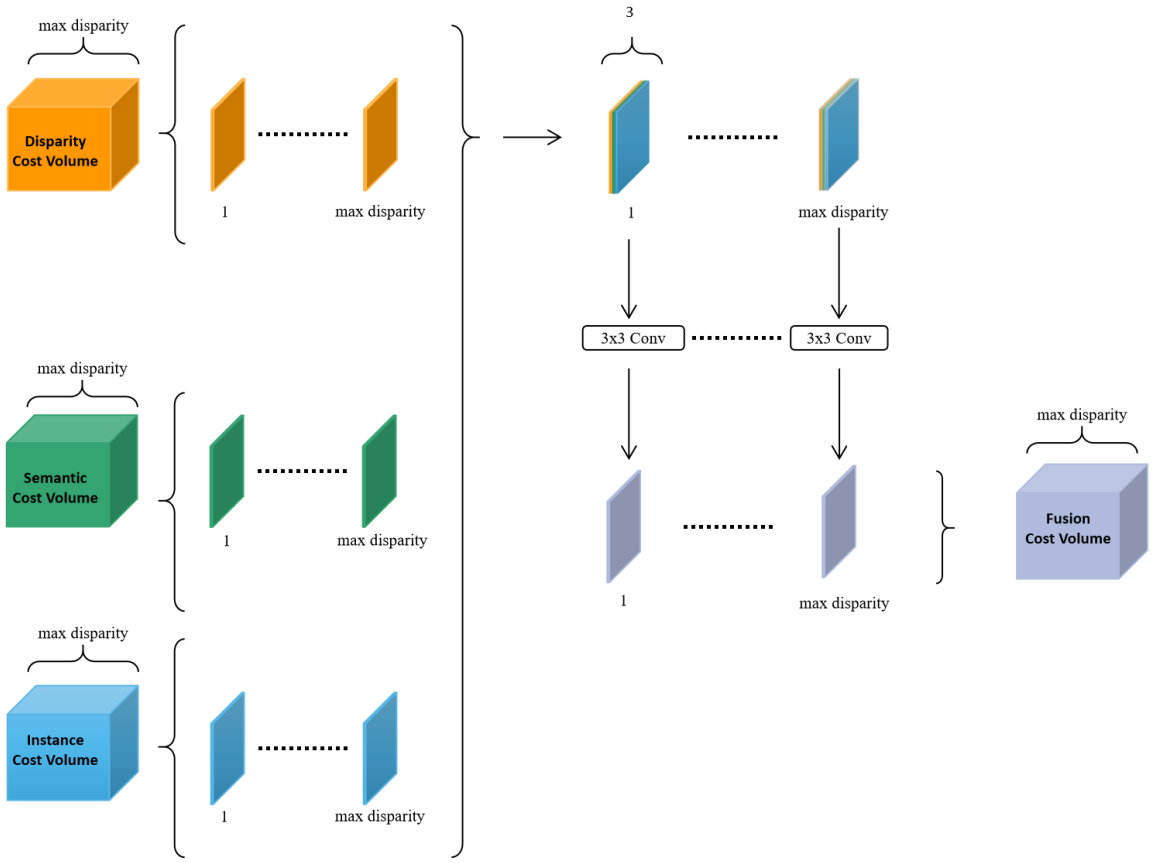


Figure 4.4.: The process of WFM. Its disassembly and recombination of the input cost volume is consistent with AFM. The recombined feature maps are processed by a convolutional layer without an activation function to achieve a effect like weighted summation.

4.4. Disparity Resolution Adaptive Structure

The idea of Disparity Resolution Adaptive Structure (DRA) in this method is used in cost volume and disparity regression (2D stacked hourglass). In short, this adaptive means that as the resolution changes, the search range of the disparity or the dimension of the convolution layer changes as well. They will be introduced in sections 4.3.1 and 4.3.2.

4.4.1. Disparity Resolution Adaptive Cost Volume

The generation process of the DRA pyramid cost volume proposed in this method can be divided into two steps: warping and DRA correlation, where warping is presented only in the generation of the cost volume of the second and third poles. Warping shifts and

reorganizes the pixels in the input left feature map horizontally by using the predicted disparity map from the previous stage (if it exists). If the disparity map is completely accurate, then the warped left feature map should be consistent with the right feature map. Conversely, if there are errors in the disparity map from the previous level of prediction, these errors are reflected as residuals in the subsequent cost volume.

The cost volume of this method is generated by cross-correlation. This approach is chosen mainly because of the small number of parameters and the fast training speed. Correlation is the process of translating the two input feature maps horizontally (disparity direction) and multiplying the overlapping parts. According to the design of correlation, the two feature maps need to be translated and multiplied by a total of *max disparity* in the horizontal direction, where *max disparity* is the maximum disparity value that can be predicted by the method. The dimensionality of the 3D cost volume obtained by correlation is W $\times$ H $\times$ *max disparity*. However, if the resolution of the feature map is already small, it does not make sense to calculate the *max disparity* as many times as possible. There are two main arguments for the above contention. First, when the resolution decreases, the disparity between the corresponding pixels in the two images decreases in the same proportion, which can be explained by the definition of disparity. Assuming that the reduced resolution image is still large enough, the disparity represented by each layer of the cost volume is actually the same proportional multiple of that of the original resolution cost volume, if the correlation is still set for the original resolution feature map. Assuming that the resolution of the feature map is changed to 1/2 of the original resolution, the disparityis actually twice that of the original resolution map when the same correlation calculation is performed as in the original resolution feature map. To match this, the *max disparity* of softmax+regression in the subsequent stacked hourglass must be doubled. This not only adds extra operations, but also directly affects the prediction results of the disparity map. Secondly, the reduction of the resolution of the feature map may make its length in the horizontal direction, i.e., the width of the image is smaller than the value of the *max disparity*. The disparity search range is meaninglessly expanded. This not only causes unnecessary calculations, but also makes it more difficult to find the right parameters during the training. All the above arguments are derived from the observation and analysis of the feature map in cost volume.

For these two reasons, DRA correlation is used in the cost volume generation process. Compared to the previously mentioned correlation, the *max disparity* is reduced proportionally while the resolution of the input feature map is reduced, which results in a proportionally smaller cost volume in terms of depth, as shown in Figure 4.5. In this way, DRA correlation allows to reduce the number of parameters and the computational effort of the network with almost no loss and to improve the training efficiency and network accuracy.

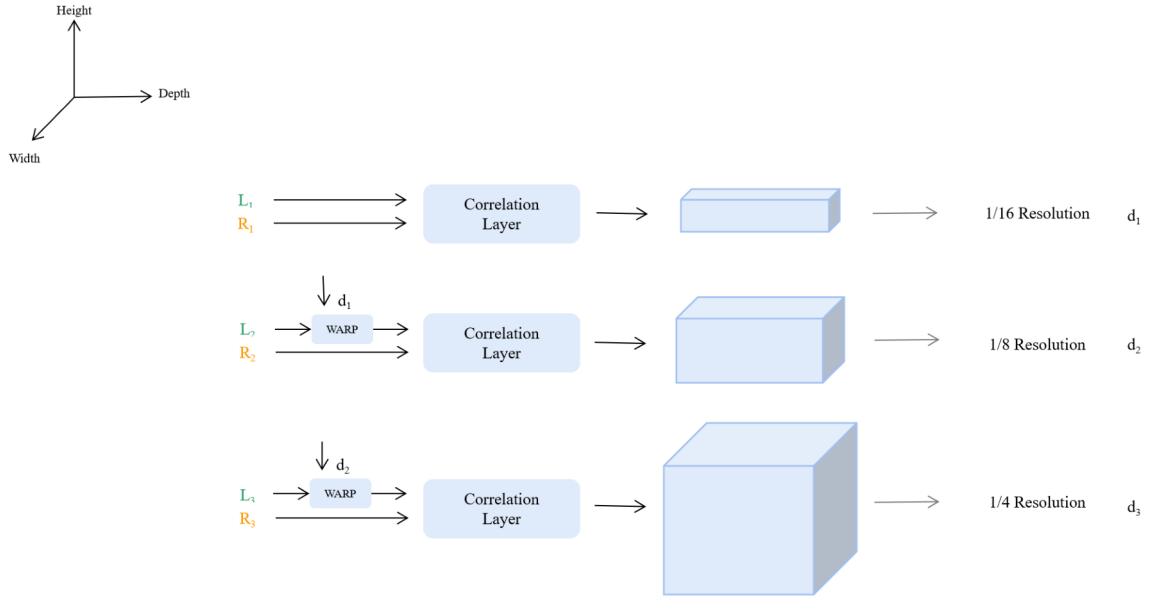


Figure 4.5.: The generation of pyramid cost volume without DRA. d is the disparity map of the network prediction. L and R are the input feature maps. The numbers in the lower corners are the stages from which they are derived. These feature maps are composed of three cost volumes with different resolutions through the correlation layer.

4.4.2. Disparity Resolution Adaptive 2D Stacked Hourglass

As the disparity regression part of the processing flow, the DRA 2D Stacked Hourglass in this method is modified and adapted based on the 3D Stacked Hourglass of PSM. This modification is mainly in two aspects: the dimensionality of the convolution kernel (3D→2D) and the dimensionality of the convolution layers (DRA). The cost volume in PSM is 4D. Therefore, 3D convolution must be used in the disparity regression part to process it. In this method, the 3D cost volume is generated by correlation. However, this change should not be done just by changing the settings of the convolution kernel, because the input information changes in form, content and volume. Therefore, the number of channels in the 2D Stacked Hourglass in this method is reset to *max disparity* as default. Similar to the first aspect, the number of channels of the 2D Stacked Hourglass used to process the cost volume with scaled resolution is also changed in equal proportion to the number of channels from the perspective of information content.

Similar to the rationale for choosing correlation as the cost volume generation method, the motivation for this design is to minimize the number of parameters required by the method within a reasonable range. As mentioned in section 3.1, 3D Conv greatly increases the number of parameters for methods such as PSM. This is especially true for methods that use multi-level cost volume & disparity regression designs such as this

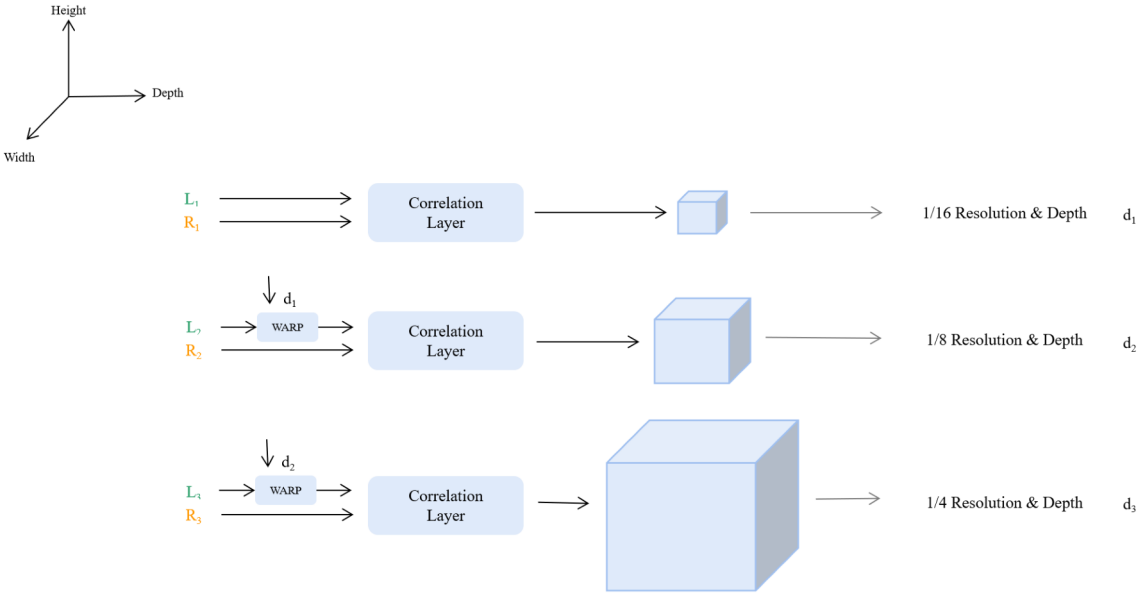


Figure 4.6.: The generation of DRA pyramid cost volume. d is the disparity map of the network prediction. L and R are the input feature maps. The numbers in the lower corners are the stages from which they are derived. These feature maps are composed of three cost volumes with different resolutions through the correlation layer.

one, where the effect of 3D Conv on the number of parameters is multiplied. The DRA in this method can further reduce the number of parameters in an interpretable way based on the 2D Stacked Hourglass.

4.5. Loss Function

The loss function used in this method consists of two components: panoptic loss and disparity loss. Panoptic loss is entirely derived from the design in Panoptic-DeepLab and can be described by the Equation 4.1 .

$$L_{panoptic} = \lambda_{sem} L_{sem} + \lambda_{heatmap} L_{heatmap} + \lambda_{offset} L_{offset} \tag{4.1}$$

Panoptic loss consists of weighted bootstrapped cross entropy loss for semantic segmentation head (L_{sem}), MSE loss for center heatmap head ($L_{heatmap}$) and L1 loss for center offset head (L_{offset}) . λ is the weight of each loss. This method directly uses the default setting of λ in Panoptic-DeepLab.

The disparity loss consists of the smooth L1 loss about disparity for different stages, Equation 4.2. Panoptic loss and disparity loss together form the final loss function of

this method, Equation 4.3.

$$L_{disparity} = \lambda_{stage1} L_{stage1} + \lambda_{stage2} L_{stage2} + \lambda_{stage3} L_{stage3} \tag{4.2}$$

$$L_{final} = L_{panoptic} + L_{disparity} \tag{4.3}$$

5. Experimental Setup

This chapter describes the datasets, traning and hyperparameters setting, evaluation metrics.

In order to obtain sufficient experimental data and compare the effects of different parts of the method, we divided the method into three models, Basic, volume adaptive (VA) and all adaptive (AA), according to the degree of DRA use. The VA indicates that only the cost volume uses DRA, while the AA indicates that both the cost volume and the disparity regression use DRA, i.e., the dimensionality of the 2D stacked hourglass in AA is also adapted with respect to the resolution (section 4.3.2). All three models have NOF (no fusion), CFM (confidence fusion module, shown in Figure 5.4, which is derived from PG-Net [38]), AFM and WFM four different variants for the feature fusion.

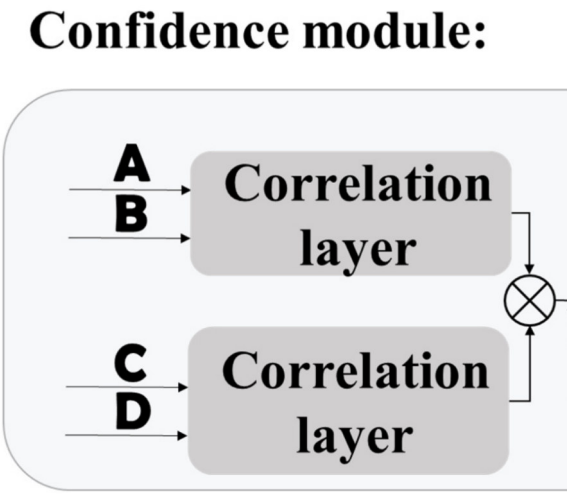


Figure 5.1.: Confidence module of PG-Net [38], where A & B and C & D are the stereo feature maps from different branches.

5.1. Datasets

Sceneflow FlyingThings3D [39], Sceneflow Driving [39] and KITTI 2015 [40] datasets are used in this work. Sceneflow FlyingThings3D and Sceneflow Driving are only used

for pre-training of the disparity branch, while KITTI 2015 is used for both the panoptic branch and disparity branch. Sceneflow FlyingThings3D is a synthetic dataset, which consists of random objects with random positions. The sample size is about 12280. It has a ground truth density of 100%, i.e. every pixel in every image. Sceneflow Driving is also a synthetic dataset. Its content is all street scenes with vehicles driving. Compared with Sceneflow Driving, Sceneflow FlyingThings3D is not only richer in content, but also has a more balanced distribution of the values of disparity in its ground truth Figure 5.2.

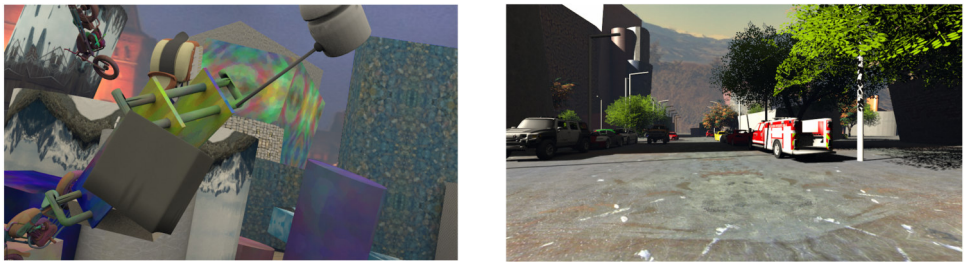


Figure 5.2.: Example for the samples in Sceneflow FlyingThings3D (left) and Sceneflow Driving (right).

KITTI 2015 presents a real street (outdoor) scenario. The sample size is 200. The density of its disparity ground truth is about 28% and its source is LiDAR [1]. The density of ground truth of its panoptic is 100% and is manually labeled. In this experiment, the maximum disparity is limited to 196. Therefore, the disparity data with disparity greater than 196 in the dataset are filtered out when training the disparity branch. Also, data with disparity less than or equal to 0 were filtered out as outliers.

The KITTI 2015 dataset was divided into training-set, validation-set and test-set in the ratio of 8:1:1. Since the dataset was numbered in temporal order, validation-set and test-set were created by systematic sampling to avoid the occurrence of similar sample sets. Similarly, Sceneflow FlyingThings3D and Sceneflow Driving are divided into training-set and validation-set. There is no test set, because the results of the pre-train are not evaluated.

The samples in the dataset are randomly cut to a size of 512×256. Input normalisation is performed before being fed into the network.

5.2. Training and Hyper-parameter Settings

This chapter introduces details of the training process settings.

Training and termination strategy

The experiments were performed twice in total, and they were denoted as 1.Training and 2.Training. The 2.training was performed only for VA and AA, and it was based on the Basic of 1.Training. In addition, we also compare the disparity maps predicted by each of the 12 variants at different stages in order to observe the performance of the disparity output at different stages under different configurations.

Variant Name	Training Phase	Training Behavior	Training Setting
NOF	Training phase 1	Pre-training disparity branch, encoder	Batch size: 6; LR: 0.0005
	Training phase 2	Training disparity branch, encoder	Batch size: 2; LR: 0.000167
	Training phase 3	Training only panoptic decoder	Batch size: 2; LR: 0.000167
	Training phase 4	Training all branch, without fusion module	Batch size: 2; LR: 0.000167
	Training phase 5	Training all branch, without fusion module	Batch size: 2; LR: 0.0000167
CFM	Training phase 1	Training all branch based NOF, with CFM	Batch size: 2; LR: 0.000167
	Training phase 2	Training all branch based NOF, with CFM	Batch size: 2; LR: 0.0000167
AFM	Training phase 1	Training all branch based NOF, with AFM	Batch size: 2; LR: 0.000167
	Training phase 2	Training all branch based NOF, with AFM	Batch size: 2; LR: 0.0000167
WFM	Training phase 1	Training all branch based NOF, with WFM	Batch size: 2; LR: 0.000167
	Training phase 2	Training all branch based NOF, with WFM	Batch size: 2; LR: 0.0000167

Table 5.3.: Experiment setting for variants.

In this work, the disparity branch and the panoptic branch are first trained separately, and then the whole network is trained together. In addition, since there is a multi-stage disparity map output in this method, it is also worth considering whether it is meaningful and necessary to train the disparity branches separately in different stages. So, three different training methods based on Basic-NOF are tried: all split (As), progressive (Pg) and direct (De). All split means that the parameters corresponding to the three disparity output stages are first trained individually. Progressive means that the parameters are trained progressively from stage 1 to stage 3. Direct means that all parameters are trained direct together. Table 5.4 shows the specific settings of these different training methods.

Early stopping is used as the training termination strategy and the best weights is used as the training results. Specifically, the training with Sceneflow FlyingThings3D, Sceneflow Driving and KITTI 2015 was stopped early after 4, 20 and 400 epochs, respectively. This setting is based on the number of samples in the dataset and is calculated in equal proportion to the setting of [1].

Training Strategies	Training Phase	Training Content	Training Setting
As	Training phase 1	Encoder, stage 1	Batch size: 2; LR: 0.000167
	Training phase 2	Stage 1	Batch size: 2; LR: 0.000167
	Training phase 3	Stage 2	Batch size: 2; LR: 0.000167
	Training phase 4	Stage 3	Batch size: 2; LR: 0.000167
	Training phase 5	Encoder, Stage 1,Stage 2, Stage 3	Batch size: 2; LR: 0.0000167
Pg	Training phase 1	Encoder, stage 1	Batch size: 2; LR: 0.000167
	Training phase 2	Encoder, stage 1, stage 2	Batch size: 2; LR: 0.000167
	Training phase 3	Encoder, stage 1, stage 2, stage 3	Batch size: 2; LR: 0.000167
	Training phase 4	Encoder, stage 1, stage 2, stage 3	Batch size: 2; LR: 0.0000167
De	Training phase 1	Encoder, stage 1, stage 2, stage 3	Batch size: 2; LR: 0.000167
	Training phase 2	Encoder, stage 1, stage 2, stage 3	Batch size: 2; LR: 0.0000167

Table 5.4.: Experiment setting for the training strateges.

Learning Rate and Batch Size

Two combinations of learning rate (LR) and batch size (BS) were chosen: BS: 6 & LR: 0.0005 and BS: 2 & LR: 0.000167. The combination of BS: 6 & LR: 0.0005 was used in the pre-training. The reason for choosing this batch size is mainly due to the hardware limitation. Usually, when we increase the batch size to N times the original size, the learning rate should be increased to N times the original size to ensure that the weights are equal after the same number of samples, according to the linear scaling rule [41]. The learning rate is set according to the above idea and is based on BS: 12 & LR: 0.001 in PSM, Panoptic-DeepLab and many other works. The combination of BS: 2 & LR: 0.000167 was set for similar reasons. This combination was used for training at KITTI 2015 after pre-training. Note that at the end of this training each variant was fine-true using KITTI 2015, where the batch size remained the same and the learning rate was 1/10 of the original, i.e. BS: 2 & LR: 0.0000167.

Parameter initialization and pre-training

The parameters of the instance and semantic branch of this method is pre-trained on cityscapes [42]. The parameters of the disparity branch are first initialized using PyTorch's default settings. Then the disparity branch and encoder are pre-trained on Sceneflow FlyingThings3D and Sceneflow Driving.

Hyperparameter setting

The hyperparameters involved in this method are mainly the weights of the components in the loss function. The ratio between the overall loss of the disparity branch and the overall loss of the panoptic branch is set to panoptic loss: disparity loss = 1:1. Inside the

panoptic loss, we keep the Panoptic-DeepLab setting unchanged. Inside the disparity loss, the 0.5:0.7:1.0 setting for the stacked hourglass from PSM is continued in the DRA 2D stacked hourglass in this method. The final disparity map of the three levels output by this method has a similar concept to the output of the stacked hourglass. Therefore, their weight in the calculation of the total loss is also set as stage 1: stage 2: stage3 = 0.5:0.7:1.0. The setting of hyperparameters is not further discussed and investigated in this work.

5.3. Evaluation Strategy and Criteria

This chapter describes the evaluation metrics used to assess the results of the experiments.

5.3.1. Disparity Error Metrics

Since only a single comovement quantity, disparity, needs to be evaluated in disparity prediction, Mean Absolute Error (MAE) , Root Mean Square Error (RMSE) and Pixel Error Rate (PER) are used in this thesis in order to evaluate the disparity prediction for quantitative evaluation of disparity prediction. They can be described by Equation 5.1, Equation 5.2 and Equation 5.3:

$$MAE = \frac{1}{|D|} \sum_{p \in D} |d_p - \hat{d}_p| \tag{5.1}$$

$$RMSE = \sqrt{\frac{1}{|D|} \sum_{p \in D} (d_p - \hat{d}_p)^2} \tag{5.2}$$

$$PER_\tau = \frac{|\{\hat{p} | \hat{p} \in D \wedge |d_{\hat{p}} - \hat{d}_{\hat{p}}| > \tau\}|}{|D|} \tag{5.3}$$

where d and $\hat{\mathrm{d}}$ denote the disparity prediction and ground truth values for pixel p, respectively. And D denotes the set of all pixels with reference disparity. MAE reflects the actual situation of the prediction error. The absolute value is useful because it can avoid the offset between positive and negative errors. The calculation of RMSE is similar to that of standard deviation, but the object and purpose of the statistics are different. The standard deviation is used to measure the degree of dispersion of a set of numbers, while the RMSE measures the deviation between the observed and true values. Compared to MAE, RMSE is more sensitive to outliers, i.e., if there is a predicted value that differs significantly from the true value, then RMSE will be large. The t in PER specifies the threshold from which the disparity estimation is considered incorrect.

By setting different thresholds, a more refined analysis of the prediction results with different levels of accuracy can be achieved. 1, 3 and 5 pixels are used as values of t in the evaluation of this work.

5.3.2. Region Masks

Region mask is a number of regions that are extracted separately. By evaluating the results in these regions, we can better understand the characteristics of the methods. The region masks used in this thesis can be divided into two main categories: characteristics-based masks and semantic-based masks. Region masks are used in this work to evaluate disparity maps.

We divide the regions according to two characteristics: textureless and occlusions. The principle of cost volume in this method is actually to compute the disparity by finding the corresponding pixels in the two images. Given the nature of the binocular stereo image pair, there is bound to be a part of the left image where no corresponding pixel is found in the right image. How does this method and its variants behave in these occluded regions? This is a question worth exploring. In addition, the estimation of disparity in low-texture regions is an ongoing challenge for binocular stereo matching algorithms, which is also interesting to evaluate.

The semantic-based region masks were designed according to the categories in the cityscapes dataset, which include construction, human, flat, nature, object, and vehicle. The disparity evaluation of these region masks allows us to gain a deeper understanding of the method and its variants.

5.3.3. Panoptic Error Metrics

This thesis uses the evaluation criteria proposed by the FAIR research team [7] for panoptic segmentation: PQ (panoptic segmentation), SQ (segmentation quality) and RQ (recognition quality), which can be expressed by Equation 5.4:

$$PQ = SQ \times RQ = \frac{\sum_{(p,g)\in TP} IOU(p,g)}{|TP|} \times \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{5.4}$$

IoU (Intersection over Union) is one of the most common metrics used in target detection and semantic segmentation. It calculates the ratio of the intersection of prediction and ground truth at the pixel level to the union. The TP, FP and FN in the equation are computed in an instance-oriented manner. The instances in prediction are determined as TP only if the category is the same as the corresponding part of the ground truth and its IOU with the ground truth region is strictly greater than 0.5. FP represents the sample with IOU greater than 0.5 but the wrong category is predicted. FN represents

the samples that are not predicted. The similarities and differences between TP, FP and FN can also be expressed in Figure 5.5.

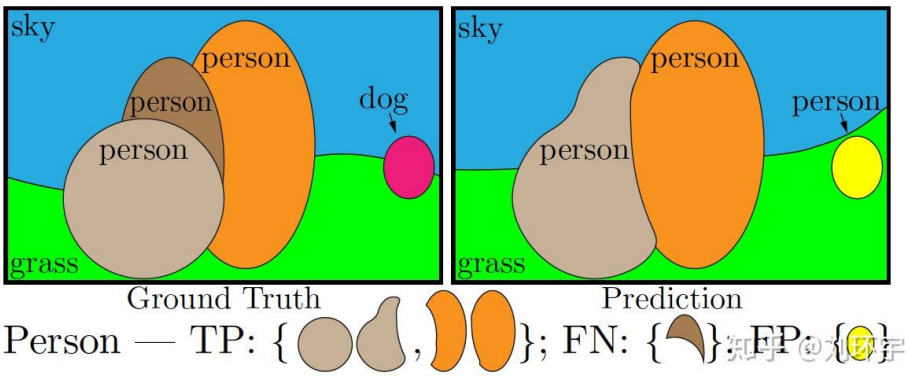


Figure 5.5.: Examples of how TP, FP and NP are determined in the PQ calculation [7].

In this theis, PQ, SQ and RQ are used to evaluate the prediction results at Things and Stuff. The contents of Things and Stuff are listed in the Table 5.6. RQ is actually the widely used F1-score. This metric allows to evaluate the performance of the algorithm in terms of instance detection. The higher the RQ value of a category, the better it is at detecting/identifying instances. SQ calculates the average IoU of the samples of TP. It measures the accuracy of the mask segmented by the algorithm. The larger the value of SQ, the closer the shape and edge of the predicted mask to the ground truth, i.e., the better the segmentation of the mask. PQ, as a combination of RQ and SQ, can make a comprehensive evaluation of the algorithm for panoptic segmentation.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{5.5}$$

Category	Contents
Things	Human
	Vehicle
Stuff	Object
	Construction
	Nature
	Sky
	Flat

Table 5.6.: The contents of Things and Stuff.

6. Results and Discussion

In this chapter, the experimental results are presented and analyzed. Section 6.1 presents the effect of training strategy and dataset on disparity prediction. The properties of different variants are discussed in Section 6.2. The evaluation of panoptic segmentation and its structural compatibility with disparity prediction are discussed in section 6.3.

6.1. The Effect of Training Setting on the Disparity Estimation

This section presents the effect of training strategy and dataset on disparity prediction.

6.1.1. Effect of Different Pre-Training Datasets

Since the density of disparity ground truth in KITTI 2015 is only 28%, a significant fraction of the pixels in this dataset are not available for training. The predicted disparity values for the corresponding parts are also uncontrollable, which is more evident in the continuous parts of the image without ground truth, such as the sky. Therefore, finding a better disparity prediction for these parts is also one of the reasons for pre-training.

In this work, the disparity branch is pre-trained using Sceneflow FlyingThings3D and Sceneflow Driving datasets. The results of the subsequent training based on these two pre-trained weights at KITTI 2015 are shown in Figure 6.1

In general, the difference between the two pre-training datasets on the results is mainly in the area of the sky. Although Sceneflow Driving is more similar to KITTI 2015 in terms of scenes, the single scene still makes the model unable to predict the sky very well. On the contrary, the results of Sceneflow FlyingThings3D are much better in predicting the sky. Of course, there are still obvious errors in the disparity prediction of the sky. By observing the results of several sets of predictions, we found that the values of the wrong disparity in the sky area in the disparity map are mainly in the range of [5,8]. In this case, even if the method is applied directly to an autonomous driving environment, the error in the prediction of the sky area does not affect the usage very much. In the remainder of this paper, Sceneflow FlyingThings3D is used for pre-training by default, unless otherwise specified.

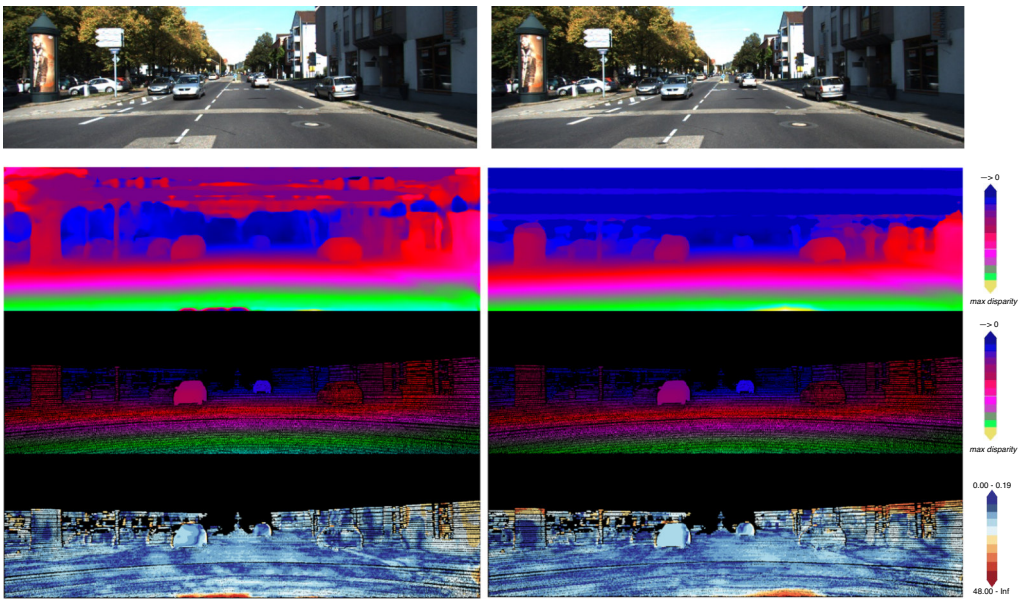


Figure 6.1.: Results obtained using different pre-trained datasets.The left side is using Sceneflow Driving, the right side is using Sceneflow FlyingThings3D. The first row is the reference image. The second row is the predicted disparity map by this method, where the darker color means the smaller disparity value. The third row is the ground truth. The fourth row is the difference between the predicted disparity map and the ground truth.

In addition to using a more comprehensive pre-trained dataset, a potentially more effective approach to this problem is to complement the existing dataset with a manual process. In addition to existing deep learning-based disparity estimation methods that have already been trained, traditional binocular stereo matching methods can also be used to perform a "pre-processing" of the dataset. More specifically, existing dataset with a low density of disparity ground truth are first fed into these methods and a disparity map is obtained. Then, the disparity values of the pixels not covered by the ground truth in this dataset are complemented by this disparity map. Although there is outliers in the ground truth thus obtained, it is sufficient for the sky or very distant regions.

6.1.2. Impact of Different Training Strategies

This section shows the training results and analysis of disparity branch based on basic-NOF according to the training method shown in Table 5.4. The experimental results show that there is almost no difference between the final training results of progressive (Pg) and direct (De), while the results of all split (AS) are significantly worse.

Table 6.2 shows the training results of the Pg strategy. In the first phase of training, the

accuracy of the output of stage 1 is greater than that of stages 2 and 3 because only the lowest resolution is involved in the training. Due to the pre-training of the model and the correlation between the different stages of the method, the outputs of stage 2 and 3 are not completely disordered. In the second phase of training, the accuracy of stage 2 and 3 increases significantly. This is reasonable and expected, because as the parameters are unfrozen, the trained layers naturally perform better than the untrained ones. However, it is important to note that the accuracy of stage 1 is also increased. This means that the simultaneous training of multiple stages helped the parameters trained in the previous stage to find a new optimum, i.e., to leave the local optimum. This situation is maintained until the final stage. The result of the third phase becomes worse compared to that of the second phase. Given that the highest accuracy of this strategy is very close to that of the second stage, This situation could be due to the oscillation of the parameters close to the optimum. The accuracy of this training strategy reaches its maximum after the final training phase.

Training Phase	**Output**		
	Stage 1	Stage 2	Stage 3
Phase 1: Encoder, stage 1	12.13	14.44	14.44
Phase 2: Stage 2	11.35	10.75	10.75
Phase 3: Stage 3	11.61	10.79	10.79
Phase 4: Encoder, Stage 1,Stage 2, Stage 3	11.51	10.74	10.74

Table 6.2.: Evaluation results of the disparity output training with Pg strategy. Metric: PER-3 (%)

The results of the De training strategy are shown in Table 6.3. The optimal accuracy of this strategy is almost indistinguishable from that of Pg. However, from the output of stage 1, the accuracy of the Pg strategy is higher. This phenomenon can be explained by the weights of the losses of each part of the loss function: the weights of the loss of these three stages in this experiment are set as stage 1: stage 2: stage 3 = 0.5:0.7:1.0. When more than one stage is involved in the training, the prediction results of the lower stages are more ” disregarded” because the weights of the higher stages are higher. This results in a worse prediction result for stage 1. It also explains a similar phenomenon in the Pg strategy. This conjecture can be verified by setting the weights to stage 1: stage 2: stage 3 = 1.0:1.0:1.0. However, this experiment is designed to compare the difference in accuracy (optimal accuracy) of stage 3 for different strategies. Therefore, this phenomenon will not be further investigated in this work.

The results of the AS strategy are shown in Table 6.4. The changes in the results of training stages 1 to 3 are basically as expected. In the results of training phase 4, the prediction accuracy of stage 1 decreases. This phenomenon is the same as that shown in Pg and De, which again confirms the conjecture of the reason presented in the previous paragraph. What is more interesting is that the optimal accuracy of the training results

Training Phase	Output		
	Stage 1	Stage 2	Stage 3
Phase 1: Encoder, stage 1, stage 2, stage 3	12.27	10.60	10.60
Phase 2: Encoder, stage 1, stage 2, stage 3. with 0.1 x LR	12.27	10.60	10.60

Table 6.3.: Evaluation results of the disparity output training with De strategy. Metric: PER-3 (%)

of this strategy is significantly smaller than that of Pg and De. In fact, a similar comparison is made in this work for the training process of AFM. The split training (first training only the parameter of the AFM and then training all parameters) has a smaller accuracy than that of direct training (training all parameters together directly). This phenomenon could be caused by the "parameters trapped in local minima", i.e. although focusing on a certain part of the training can make the parameters in that part reach the "current optimum", it also tends to make the parameters in that part fall into local minima. Since this problem is not the focus of this paper, its specific causes are not discussed. The rest of the experimental results are considered to be obtained by training the De strategy unless otherwise specified.

Training Phase	Output		
	Stage 1	Stage 2	Stage 3
Phase 1: Encoder, stage 1	11.94	13.82	13.82
Phase 2: Stage 2	11.98	13.89	13.89
Phase 3: Stage 3	11.99	13.86	13.86
Phase 4: Encoder, Stage 1,Stage 2, Stage 3	14.34	13.04	13.04

Table 6.4.: Evaluation results of the disparity output training with AS strategy. Metric: PER-3 (%)

6.2. Evaluation of Disparity Estimation

This section evaluates the disparity estimation. First an general analysis of the results is performed in section 6.2.1. Subsequently, we compare and analyze each fusion module in more detail.

6.2.1. General Comparison among Performances of all Variants

This section provides a general analysis of the experimental results obtained according to configure in Table 5.4 (section 5.2). The purpose of this section is to get an overview of the disparity performance of each variant and to introduce subsequent sections by comparing and analyzing the experimental results with the expected ones. Therefore, this section only focuses on analyzing the average MAE for the disparity output of highest resolution (stage 3) of different variants. The results of this section are shown in Table 6.5.

			Average MAE
1.Training	Basic	NOF	1.61
		CFM	1.37
		AFM	1.53
		WFM	2.92
	VA	NOF	1.49
		CFM	1.62
		AFM	1.41
		WFM	1.42
	AA	NOF	1.79
		CFM	1.95
		AFM	1.76
		WFM	2.72
2.Training	VA	NOF	1.36
		CFM	1.48
		AFM	1.45
		WFM	1.77
	AA	NOF	1.55
		CFM	1.82
		AFM	1.81
		WFM	1.66

Table 6.5.: Results obtained using different variants. The smaller the value, the higher the accuracy. Unit: pixel.

Under the "multi-stage" training of this method, the optimal training result of the previous stage has a negative impact on the next stage of training. This argument is obtained by combining the analysis in section 6.1. It has been shown in section 6.1.2 that the results obtained by using the De strategy are worse. So, this phenomenon

could also be caused by "parameters trapped in local minima". This argument can also explain the case shown in Table 6.5. The variants with fusion module are trained based on NOF. In training results of VA and AA, the overall average MAE of AFM in 1.Training is smaller than that of 2.Training, while in contrast, the average MAE of NOF in 1.Training is better than that of 2.Training. Moreover, in the training results of Basic, the CFM results are higher than the AFM for the first time and reach the highest precision of all variants with fusion module in the available data, while the precision of NOF in this group is lower (the second lowest precision in the available data on NOF). Although the Basic has only been experimented once, the accuracy of NOF is already lower than the NOF in AA of 2.Training, which is theoretically very unlikely to happen. So, we prefer to consider the results in the Basic as "non-optimal training results". All the above evidence proves that higher accuracy can be obtained by using lower NOF as the starting point.

As for the experimental data in Table 6.5 that do not exactly match the argument in the previous paragraph, such as the CFM in the training results of VA and AA, they can be explained by the principle of the corresponding fusion module. As mentioned before in section 3.3, the CFM is more sensitive to the error information in the input cost volume. At a heigher MAE of NOF, i.e. NOF in VA and AA in 1. Training, the proportion of error information in disparity cost volume must be greater. Then, it is conceivable that CFM outputs a worse disparity map due to the greater interference from the error information. Due to time and hardware constraints, the training part is not further investigated in this work. However, it is still a question of how to deal with the drawbacks brought by this "multi-stage" training model.

One possible solution is to move away from NOF and use a more comprehensive dataset to pre-train variants with fusion modules. This requires a dataset with both panoptic and disparity ground truth and a large enough amount of data, e.g. KITTI 360. Another possible solution is to perform dynamic and automatic training. The dynamics is not only in the different stages of the individual variants, but also in the transition from NOF→variants with fusion module. Automatic means that instead of manually dividing the training stages, the different training stages are linked in a functional way (e.g. similar idea is focal loss).

In general, the variants using the fusion module (either CFM, AFM or WFM) hardly reach the accuracy level of NOF in the same group. In the two experimental results of VA, the highest accuracy was found in the NOF variant of 2.Training. Although the AFM of 1.Training exceeds the NOF of 1.49 with 1.41 in average MAE, it is still lower than the NOF of 2.Training (average MAE=1.36). A similar situation occurred in the data of AA. The results of Basic are very different from that of VA and AA. In the case of Basic, the highest precision is found in the CFM results. In addition, the precision of AFM (average MAE=1.53) is also higher than that of NOF (average MAE=1.61). However, it is important to note that only one experiment was performed in the Basic group. Given the randomness in training exhibited by the data of VA and AA, one experiment is less representative than two experiments. Therefore, the author prefers to

argue that with the available data, it is difficult to achieve the accuracy level of NOF by using the variant of fusion modules.

For the reason of the phenomenon shown in the previous paragraph, two speculations are proposed: 1. Due to the inappropriate training method and the limitation of the data set, it is difficult to find better parameters for the model with fusion modules. 2. With the current design of the method, the features between the panoptic and disparity branches are still incompatible.

The point about the training method in the first speculation has been mentioned several times in this paper. The conjecture that "parameters are trapped in local minima" can explain the difference in results using different training strategies and the difference in results based on different precision levels of "starting points", which have been introduced in section 6.1.2 and the second paragraph of this section, respectively. All variants with fusion module are based on the training results of the same group of NOF. One of the original purposes of this design is to provide a good initial value for the training of variants with fusion modules, which is similar to the motivation of pre-training. However, one of the differences between this operation and pre-training is that pre-training is performed by using different datasets for meta-feature transfer. Therefore, "parameters trapped in local minima" may also be a reason to limit the fusion module approach. On the other hand, the parameters in the fusion module are never pre-trained. This is because there is a lack of pre-trained datasets with both disparity and panoptic ground truth. Even if the variant with fusion module is trained directly based on the results of pre-training in Table 5.3, i.e. the results of training phase 1, the parameters in the fusion module are still not initialized by pre-training. Therefore, the limitation of the dataset may also have a negative impact on the training of the parameters of the fusion module.

The compatibility between feature maps is difficult to quantify. Directly comparing the accuracy of the output with and without fusion module is the easiest and most intuitive way to determine compatibility. As far as the available data are concerned, the highest output accuracy is still achieved by NOF. In addition, the difference in information from different branches in this method provides the possibility of complementing information from multiple branches on the one hand, and does make the fusion more difficult on the other. The compatibility problem of direct fusion between feature maps at different scales has been pointed out by Zhenli Zhang, Xiangyu Zhang in ExFuse [43]. From these two points of view, the compatibility between features from different branches in this method is indeed questionable. However, the positive effect of CFM in this part of feature fusion has been demonstrated in SG-Net [37] and PG-Net [38]. Moreover, from the previous analyses on training strategies and methods, it is clear that the accuracy of the variants with fusion module in the available data is likely to be limited. Therefore, although the author recognizes that the three fusion modules in this paper do not achieve the expected good results in exploiting multi-branch features, there is no sufficient evidence that the features from different branches in this method are not compatible with each other.

VA makes it easier to achieve higher accuracy. AA has the highest accuracy/parameter ratio. Although the overall precision of AA is significantly lower than the other two groups, its advantage in terms of number of parameters is very obvious. In contrast, the results from the basic data are more complicated. On the one hand, the CFM has the highest precision, and on the other hand, most of the results in this group are not as good as that in VA. All three groups, Basic, VA and AA, exhibit the expected properties (see chapter 4 for details).

The descending ranking of the three fusion modules in terms of accuracy is: AFM≈CFM>WFM. The lowest ranking of WFM can be clearly identified from the Table 6.5. Except for AA in 2. Training, the MAE of WFM is the largest in the rest of the groups. Compared to the other four groups, the good performance of WFM in AA in 2.Training alone is not enough to show that it can reach the same level of accuracy as AFM and CFM. Therefore, with the available data, WFM can achieve the lowest accuracy among the three fusion modules. The comparison of AFM and CFM is more complicated. From four groups data of VA and AA, the accuracy of AFM is consistently higher than CFM in the same group. This difference is particularly evident in 1.Training. This evidence suggests that AFM seems to be more accurate than CFM. However, it cannot be ignored that in the Basic group, the accuracy of CFM is higher than that of AFM in any other group. Despite the randomness in training mentioned in the previous paragraph, the fact that this variant achieves high accuracy should not be ignored. The AFM could be improved by multiple experiments. However, the possibility that CFM can achieve higher accuracy than AFM cannot be ruled out with the present data alone. Therefore, in the analysis of this section, the accuracy of AFM and CFM is considered to be the same for the time being. Further analysis of the three fusion modules will be presented in section 6.2.2 and 6.2.3.

6.2.2. Comparison and Analysis between AFM and CFM

We evaluated the different regions of the scene using the region masks. The evaluation results are shown in Table 6.6. The purpose of this section is to find out the difference between the performance of AFM and CFM in different regions of the scenes. Therefore, this evaluation metric is more intended for comparison between AFM and CFM. It does not measure the absolute performance of the models. It is obvious from the data that the performance of AFM and CFM in different areas is indeed different. And there is a strong pattern in this difference.

In general, AFM tends to have better accuracy for disparity prediction in human and vehicle regions. CFM tends to perform better in the flat (including road) and object regions. The difference in performance between the two fusion modules is not significant in the remaining areas not mentioned. The superiority of AFM in predicting disparity of human regions is very obvious. AFM outperforms CFM in the human regions results regardless of the data set, and this difference in accuracy is very large. In the disparity

			Construction	Flat	Human	Nature	Object	Occlusions	Road	Textureless	Vehicle
1.Training	Basic	CFM	30.79	2.46	8.42	11.97	26.13	24.59	2.65	8.83	5.48
		AFM	31.66	1.61	**5.64**	14.05	31.95	26.92	1.69	10.55	6.39
	VA	CFM	31.71	2.26	11.46	17.04	39.93	30.73	2.22	10.84	6.04
		AFM	25.65	1.66	**8.95**	12.10	32.02	23.51	1.59	9.13	**3.96**
	AA	CFM	35.37	3.79	23.59	21.38	38.80	36.36	3.79	16.54	10.45
		AFM	31.87	2.91	**12.66**	16.73	36.24	28.40	3.03	14.49	**6.69**
2.Training	VA	CFM	30.77	**2.75**	10.65	13.13	22.80	20.33	**2.85**	9.90	5.07
		AFM	26.35	2.98	**4.54**	12.12	27.05	22.26	3.09	9.37	**3.94**
	AA	CFM	30.36	**2.80**	14.04	16.96	33.94	30.17	**2.17**	14.31	6.82
		AFM	29.55	6.03	**9.00**	14.97	34.81	27.08	6.35	14.00	**5.52**

Table 6.6.: Evaluation results of CFM and AFM using region mask. The marked values are "anomalous", i.e., they have lower overall accuracy but a better local accuracy. Metric: PER-3 (%).

prediction of vehicle regions, AFM brings higher accuracy by a large margin in most cases. E.g. in VA and AA of 2.Training, AFM outperforms CFM by 22.25% and 19.03% respectively. Although CFM in basic exceeds the accuracy of AFM in vehicle regions, it is still much lower than the results of AFM in VA of 2.Training, which is with smaller overall accuracy. Therefore, the performance of AFM in the vehicle region is considered to be better. The argument that CFM tends to perform better in the flat (including road) and object regions is similar to the above. Figure 6.7 illustrates an example of the above phenomenon. The reason why "the difference between the performance of these two fusion modules in the remaining regions not mentioned is not significant" is that it is more noteworthy that the data with higher accuracy in individual regions appear in the overall lower accuracy results. The overall accuracy of AFM in VA and AA of 2.Training is higher than CFM. Therefore, it is not surprising that its accuracy is higher in the construction, nature, occlusions and textureless regions. The data for the human, vehicle, flat and object regions are clearly "anomalous".

The results of the evaluation with RMSE and PER for AFM and CFM (Table 6.8) show that the accuracy of AFM is more discrete and "inhomogeneous" on the test set. The AFM in AA of 2.Training has a lower average MAE than that of CFM counterpart, but performs worse on average RMSE, average PER-3 and average PER-5. From the mathematical definition of RMSE, higher RMSE means that the variance between the accuracy of different pixels in the disparity map is larger, i.e., the disparity accuracy of pixels is more inhomogeneous. Higher average PER-3 and average PER-5 imply that the disparity maps predicted by the AFM are in general more different from the ground truth. However, it is worth noting that the average PER-1 of AFM is consistently better than that of CFM. Even though the other metrics of CFM in Basic are much better than that of AFM, the average PER-1 of AFM is still lower than that of CFM. This phenomenon indicates that more disparity of pixels are correctly predicted with a very high degree of accuracy in the AFM prediction results. Combining the results

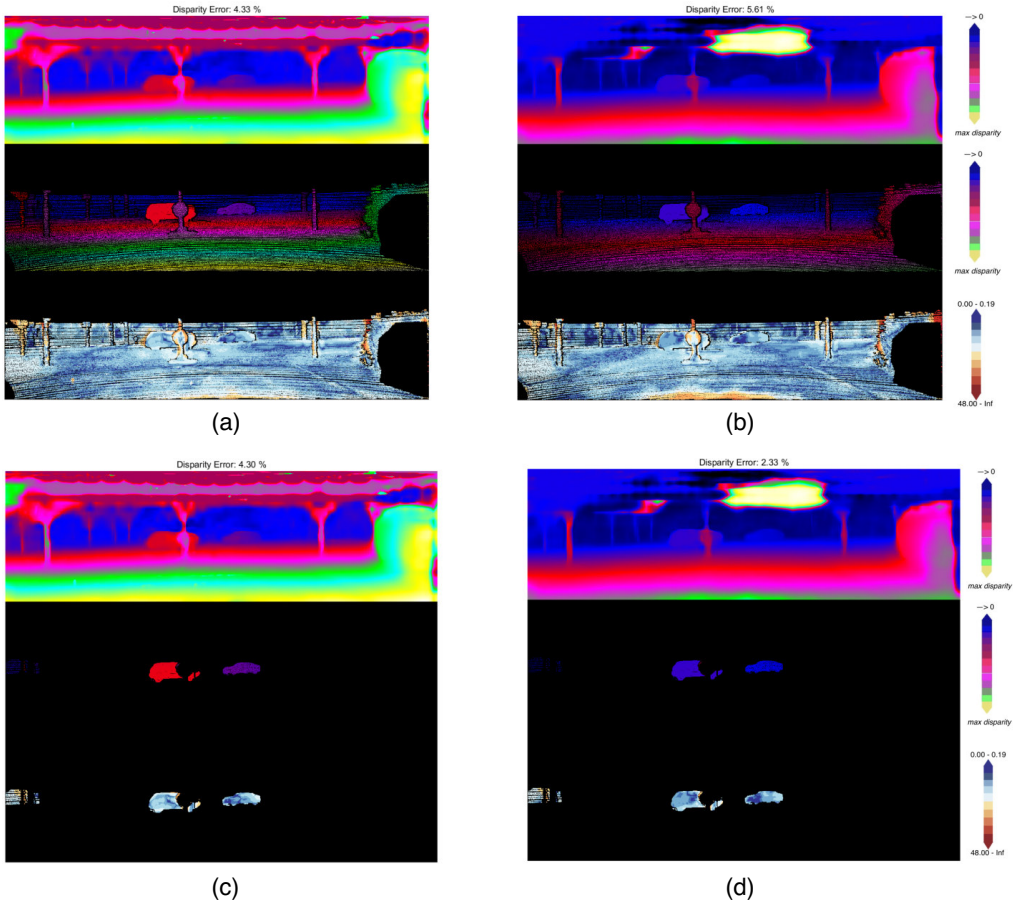


Figure 6.7.: Comparison of the disparity map output by CFM (left) and AFM (right). (a) and (b) are the overall disparity error (PER-3) for CFM and AFM, with values of 4.33% and 5.61%. (c) and (d) are that of CFM and AFM in the vehicle region with the values of 4.30% and 2.33%, respectively. All images in (a)-(d) from top to bottom are: the predicted disparity map, the ground truth and the difference between the predicted map and the ground truth. Compared with CFM, AFM has significantly more errors in the road regions. Meanwhile, although the overall error rate of AFM is slightly higher than CFM, its error rate in the vehicle region is much lower than CFM.

of RMSE and PER-1, 3, and 5, we can conclude that, compared to CFM, a portion of the disparity values predicted by AFM are correctly predicted with a high degree of accuracy, but a larger portion are far from the ground truth, i.e., the prediction accuracy of AFM is more discrete and "inhomogeneous".

The results of the above experiments are interpretable. CFM fuses the features at the probability level by multiplication. If the quality of these feature maps is good, this simple and identical calculation for all pixels can easily improve the disparity prediction. If the disparity map is flawed, the disparity prediction in the flawed areas will not improve or even get worse. A clear example of this is the difference in the quality of the

			MAE	RMSE	PER-1	PER-3	PER-5
1.Training	Basic	CFM	1.37	3.00	37.86%	7.72%	3.46%
		AFM	1.53	6.25	36.98%	11.11%	6.70%
	VA	CFM	1.62	3.85	37.50%	9.18%	4.73%
		AFM	1.41	3.33	33.01%	7.46%	3.83%
	AA	CFM	1.95	4.08	50.84%	12.99%	6.25%
		AFM	1.76	3.94	43.02%	10.76%	5.45%
2.Training	VA	CFM	1.48	3.33	38.24%	8.29%	4.02%
		AFM	1.45	3.52	34.11%	8.03%	4.20%
	AA	CFM	1.82	4.31	43.20%	10.52%	5.25%
		AFM	1.81	4.13	42.62%	11.42%	5.95%

Table 6.8.: Evaluation results of CFM and AFM.

predictions for the human and road regions in Table 6.6. In panoptic map of all variants, the accuracy of the road regions is often very high. This means that the information about road in the output CFM is rich and correct. Therefore, the disparity prediction of CFM in the road region is also very good. The accuracy of the panoptic segmentation of the human regions is much lower than that of the road regions (see section 6.3.1 for details). Therefore, it is not surprising that the disparity prediction of CFM in the human regions is much worse than that of AFM.

In contrast, AFM has the ability to filter and further process information. Even if the input information is not very accurate, AFM is able to select the useful information from it and is less affected by the noise in it. Therefore, it has a very good performance in the human regions, which is in line with our original intentions and expectations of designing AFM, i.e., to let the machine decide how to use the input features. However, with the current experimental results, this "decision making" capability of AFM is still far from mature. Theoretically, AFM can achieve as good results in regions such as the road region as in the human region. And in fact, AFM in basic does achieve the highest accuracy among all variants in the flat part. However, there is no data to show that AFM can predict all regions with high accuracy at the same time, i.e., it has the ability to "make decisions" for all regions at the same time. This explains the high performance of AFM on PER-1 and the "anomalous" performance on RMSE, PER-3 and PER-5. As for the reason, there are two conjectures: 1. Due to its more complex structure, the training of AFM is very difficult, so much so that it does not learn the optimal parameters with the existing data set and training methods. 2. The two convolutional layers in AFM are not enough to make it have sufficient "decision making" ability. Due to time constraints, these two conjectures will not be further investigated in this work.

6.2.3. Evaluation For WFM

In general, the performance of WFM tends to be consistent with that of AFM. For comparison purposes, we also include the CFM data as a reference in the comparison. As can be seen in Table 6.9, WFM still has a very significant higher accuracy in the human and vehicle parts. In the regions of construction, nature, occlusions and textureless, the accuracy of WFM is similar to that of AFM. Even the WFM in AA of 2.Training shows that the accuracy of the flat region is very accurate while the accuracy of the other regions decreases, which is the same as the case of basic AFM. These experimental results also support the analysis in section 6.2.2.

			Construction	Flat	Human	Nature	Object	Occlusions	Road	Textureless	Vehicle
1.Training	Basic	CFM	30.79	2.46	8.42	11.97	26.13	24.59	2.65	8.83	5.48
		AFM	31.66	1.61	5.64	14.05	31.95	26.92	1.69	10.55	6.39
		WFM	28.18	10.82	11.21	14.24	32.11	30.69	12.38	12.39	6.98
	VA	CFM	31.71	2.26	11.46	17.04	39.93	30.73	2.22	10.84	6.04
		AFM	25.65	1.66	8.95	12.10	32.02	23.51	1.59	9.13	3.96
		WFM	26.24	2.25	5.53	11.82	30.09	28.51	2.10	9.25	4.61
	AA	CFM	35.37	3.79	23.59	21.38	38.80	36.36	3.79	16.54	10.45
		AFM	31.87	2.91	12.66	16.73	36.24	28.40	3.03	14.49	6.69
		WFM	30.51	19.46	16.35	16.59	33.83	29.44	21.42	17.72	7.84
2.Training	VA	CFM	30.77	2.75	10.65	13.13	22.80	20.33	2.85	9.90	5.07
		AFM	26.35	2.98	4.54	12.12	27.05	22.26	3.09	9.37	3.94
		WFM	25.98	6.93	5.96	12.97	26.24	21.76	7.55	10.28	3.82
	AA	CFM	30.36	2.80	14.04	16.96	33.94	30.17	2.17	14.31	6.82
		AFM	29.55	6.03	9.00	14.97	34.81	27.08	6.35	14.00	5.52
		WFM	28.70	3.06	6.30	13.73	34.55	26.96	2.68	12.39	5.30

Table 6.9.: Evaluation results of CFM, AFM and WFM using region mask.

The experimental results of WFM and AFM show that the idea of "letting the machine decide how to fuse features" is possible. The consistent and interpretable performance of WFM and AFM suggests that their high accuracy disparity prediction in human and vehicle regions is not a coincidence. Moreover, the overall better performance of AFM compared to WFM suggests that giving the fusion module a stronger "decision making" capability by adding convolutional layers is a possible direction for optimization.

6.2.4. Comparison and Analysis between Fusion Modules and NOF

The accuracies achieved by AFM and CFM in their respective dominant regions are still not surpassed by NOF. Here we have selected the data of VA in 2.Training, which with the best NOF performance of all the results. As shown in Table 6.10, although NOF is much better than AFM and CFM in terms of global accuracy, its accuracy in the vehicle and flat regions is still not significantly better than AFM and CFM. In the human and object regions, NOF is even outperformed by AFM and CFM by a large margin. This

indicates that the fusion module does have a significant positive effect on the prediction of disparity in some regions.

		Construction	Flat	Human	Nature	Object	Occlusions	Road	Textureless	Vehicle
2.Training VA	NOF	25.72	1.82	4.13	11.48	25.48	20.73	1.78	8.81	4.25
	CFM	30.77	2.75	10.65	13.13	**22.80**	20.33	2.85	9.90	5.07
	AFM	26.35	2.98	4.54	12.12	27.05	22.26	3.09	9.37	**3.94**
	WFM	25.98	6.93	5.96	12.97	26.24	21.76	7.55	10.28	3.82

Table 6.10.: Comperation of variants with region masks. (Metric:PER-3)

6.2.5. Analysis between different Stages

The higher stage outperforms the lower stage in general and in most regions. In order to most directly evaluate the structural design of the network at the disparity branch, the data evaluated here are from NOF in 2.Training. Stage 3 performs better than stage 1 in terms of MAE, RMSE, PER1, PER3, and PER5, as shown in Table 6.11. The data in Table 6.12 also shows that the accuracy of disparity prediction is greater for higher stages than that for lower stages in most regions. However, it should be noted that stage 2 is sometimes "negatively optimized" for stage 1.

		MAE	RMSE	PER-1	PER-3	PER-5
VA	stage 1	1.40	3.30	33.49%	7.48%	3.76%
	stage 2	1.39	3.28	32.98%	7.39%	3.72%
	stage 3	**1.36**	**3.25**	**32.26%**	**7.08%**	**3.58%**
AA	stage 1	1.58	3.52	39.86%	9.65%	4.88%
	stage 2	1.58	3.52	39.89%	9.65%	4.87%
	stage 3	**1.55**	**3.47**	**38.98%**	**9.21%**	**4.67%**

Table 6.11.: Evaluation results for different stages of disparity estimation. Part 1

		Overall	Construction	Flat	Human	Nature	Object	Occlusions	Road	Textureless	Vehicle
VA	stage 1	7.39	27.92	1.87	4.94	12.00	29.44	23.07	1.81	8.89	4.87
	stage 2	7.30	27.44	1.86	4.99	11.81	28.52	22.81	1.81	8.87	4.87
	stage 3	**6.99**	25.72	1.82	**4.13**	11.48	**25.48**	20.73	1.78	8.81	**4.25**
AA	stage 1	9.52	30.50	2.71	13.20	14.02	40.75	30.02	2.32	12.49	6.58
	stage 2	9.51	30.48	2.72	13.20	13.98	40.72	30.02	2.32	12.49	6.57
	stage 3	**9.08**	28.42	2.75	**8.24**	13.80	**35.24**	26.37	2.35	12.35	**5.29**

Table 6.12.: Evaluation results for different stages of disparity estimation. Part 2

Compared with the low-level stages, the accuracy improvement of higher stages is more obvious in small-sized regions, such as human, object and vehicle regions. This phenomenon can be explained by the difference of the scale of the feature maps used in

different stages. The input feature maps of the lower stages are very small, e.g., the cost volume in stage 1 is 1/16 of the original size. In this case, the size of the already small regions of human, object and vehicle is further reduced. This limits the feature representation. As the feature map size is recovered in higher stages, the disparity prediction in this region is improved very much.

Higher stages do not significantly improve the results of lower stages in large size and low texture areas, such as flat, nature and textureless regions. In addition to the size of the feature maps mentioned in the previous paragraph, i.e. different scales of feature maps do not have much effect on large size regions. Another possible reason is that the disparity prediction is interpolated (upsampled) before output. The mathematical definition of interpolation dictates that the interpolated disparity does not "jump" too much in value compared to the disparity of its neighboring pixels, i.e., it is also smooth. Compared to these interpolated disparity values, the disparity values directly predicted by the network have more uncertainty in these regions. In addition, more pixels in the output of the low-stage disparity are interpolated than those of the high-stage disparity prediction maps. Therefore, the performance of low stage in these regions is not lower than that of higher stage. Regions with "repetitive textures" such as vegetable have disparity jumps, i.e., they are not completely smooth, but the distribution of disparity is generally "uniform". Therefore, the interpolation does not have a significant negative impact on the disparity results in these regions.

6.3. Panoptic Segmentation and its Compatibility with Disparity Estimation

This chapter evaluates the quality of the panoptic maps output by different variants and discusses the structure of the network based on the data.

6.3.1. Evaluation of Panoptic Segmentation

This section focuses on comparing the differences between panoptic segmentation in different variants. The aim is to evaluate the impact of the disparity branch with fusion module on the panoptic segmentation due to feature fusion. Since the focus of this work is on the improvement of disparity estimation using panoptic information, we do not evaluate the absolute quality of panoptic segmentation. The experimental data on panoptic segmentation are shown in Table 6.13.

In general, the quality of panoptic segmentation does not differ significantly among variants. This conclusion can be drawn by comparing the coefficient of variation of different variants in PQ-All, SQ-All and RQ-All. In addition, there is no absolute positive or negative correlation between the accuracy of disparity prediction and panoptic

			PQ-All	PQ-Things	PQ-Stuff	SQ-All	SQ-Things	SQ-Stuff	RQ-All	RQ-Things	RQ-Stuff
1. Training	Basic	NOF	30.46	10.77	44.78	51.61	27.85	68.89	37.75	14.40	54.73
		CFM	30.61	12.01	44.14	53.59	32.51	68.91	38.78	17.77	54.05
		AFM	30.79	11.46	44.84	51.50	27.20	69.17	38.53	16.17	54.80
		WFM	31.20	12.51	44.79	51.33	27.31	68.80	39.25	17.74	54.88
	VA	NOF	32.40	16.40	44.04	52.85	38.44	63.33	41.82	25.17	53.93
		CFM	31.28	12.95	44.61	53.11	31.88	68.54	40.13	19.91	54.83
		AFM	32.45	15.30	44.92	56.44	39.39	68.85	41.74	23.38	55.09
		WFM	31.01	11.89	44.92	54.67	34.92	69.04	38.83	16.53	55.05
	AA	NOF	31.43	9.76	47.19	51.54	27.21	69.24	39.09	12.93	58.11
		CFM	31.50	11.42	46.11	53.11	31.69	68.68	40.09	16.86	56.99
		AFM	32.38	12.55	46.80	55.52	36.32	69.49	40.56	17.18	57.56
		WFM	32.65	12.07	47.61	54.30	34.37	68.80	41.56	17.22	59.27
2. Training	VA	NOF	31.91	12.82	45.79	51.30	27.21	68.82	40.21	18.05	56.32
		CFM	30.33	9.75	45.30	51.57	27.66	68.96	37.75	12.94	55.79
		AFM	31.83	13.69	45.02	51.34	27.29	68.82	40.24	19.57	55.28
		WFM	30.89	10.96	45.38	52.45	27.97	70.26	38.09	14.63	55.15
	AA	NOF	32.56	15.62	44.88	52.84	30.85	68.83	42.35	24.14	55.59
		CFM	28.23	6.46	44.06	47.09	17.23	68.81	35.15	8.72	54.37
		AFM	31.04	11.86	44.99	51.45	26.35	69.70	39.19	17.21	55.17
		WFM	30.89	11.22	45.20	51.07	26.02	69.28	39.19	16.43	55.75
Standard Deviation			1.03	2.21	0.98	1.98	5.13	1.34	1.70	3.90	1.38
Average			31.29	12.07	45.27	52.43	29.98	68.76	39.51	17.35	55.63
Coefficient of Variation			3.29%	**18.32%**	2.17%	3.79%	**17.12%**	1.95%	4.31%	**22.47%**	2.49%

Table 6.13.: Evaluation results for all variants of panoptic segmentation.

segmentation. For example, compared with AFM in VA of 1.training, AFM in VA of 2.training and AFM in AA of 1.Training is better and worse in disparity prediction, respectively, while their panoptic segmentation quality is lower. Similar examples can be found in other variants of the data.

It should not be ignored that the data show that the quality of the panoptic segmentation of the Things varies considerably between variants. Is the panoptic segmentation of Things strongly influenced by the fusion module? In order to answer this question, we have performed separate statistics for all variants (Table 6.14). The data show that even without the use of the fusion module, the results of the panoptic segmentation of the Things still show large fluctuations. Therefore, even though PQ-Things, SQ-Things and RQ-Things have very high coefficient of variation (more than 15%), it is still not possible to prove that the fluctuations are caused by feature fusion. However, it should be also noted that the coefficient of variation of CFM is significantly larger compared to AFM and WFM. Given the property that "CFM is more dependent on the quality of the input original feature maps", it is reasonable to speculate that CFM has a greater impact on panoptic branch.

In general, none of the variants had a significant impact on the results of panoptic segmentation. However, compared with other variants, CFM has a relatively greater impact on panoptic segmentation. In addition, the accuracy of panoptic segmentation in the Things region is significantly lower than that in the Stuff region. If future work involves the improvement of panoptic segmentation, then improving the accuracy of the Things region will be an important direction. Besides changing the structure of the panoptic branch alone, using the information in the disparity branch may also be a

		NOF			CFM			AFM			WFM		
		PQ-Things	SQ-Things	RQ-Things	PQ-Things	SQ-Things	RQ-Things	PQ-Things	SQ-Things	RQ-Things	PQ-Things	SQ-Things	RQ-Things
1. Training	Basic	10.77	27.85	14.40	12.01	32.51	17.77	11.46	27.20	16.17	12.51	27.31	17.74
	VA	16.40	38.44	25.17	12.95	31.88	19.91	15.30	39.39	23.38	11.89	34.92	16.53
	AA	9.76	27.21	12.93	11.42	31.69	16.86	12.55	36.32	17.18	12.07	34.37	17.22
2. Training	VA	12.82	27.21	18.05	9.75	27.66	12.94	13.69	27.29	19.57	10.96	27.97	14.63
	AA	15.62	30.85	24.14	6.46	17.23	8.72	11.86	26.35	17.21	11.22	26.02	16.43
Standard Deviation		2.91	4.79	5.55	2.55	6.42	4.44	1.55	6.08	2.90	0.63	4.20	1.18
Average		13.07	30.31	18.94	10.52	28.20	15.24	12.97	31.31	18.70	11.73	30.12	16.51
Coefficient of Variation		22.26%	15.79%	29.33%	24.27%	22.78%	29.10%	11.98%	19.42%	15.50%	5.41%	13.93%	7.14%

Table 6.14.: The further evaluation results of the panoptic segmentation of all variants on Things.

feasible direction.

6.3.2. Structural Compatibility of Joint Training

This section discusses the impact of the multi-task framework on individual tasks in this method. In section 6.2.1 and 6.3.1, the compatibility between feature maps from multiple branches in fusion module and the impact of fusion module on panoptic segmentation are discussed respectively. In contrast, this section discusses the compatibility of panoptic segmentation and disparity estimation on the network structure, i.e., the impact of shared encoder on the network.

	VV			AA		
	Panoptic only	**Disparity only**	**Panoptic and Disparity (NOF)**	**Panoptic only**	**Disparity only**	**Panoptic and Disparity (NOF)**
PQ-all	30.92	/	**31.91**	27.74	/	**32.56**
SQ-all	52.89	/	51.30	45.36	/	**52.84**
RQ-all	39.82	/	**40.21**	35.47	/	**42.35**
MAE	/	1.35	1.36	/	1.55	**1.55**
RMSE	/	3.24	3.25	/	3.47	3.47
PER_5	/	3.54%	3.58%	/	4.66%	4.67%

Table 6.15.: Evaluation results about the difference between single and joint output.

Experiments show that the shared encoder is capable of satisfying the needs of multiple branches in this method simultaneously. Table 6.15 shows the results of training the disparity branch and panoptic branch separately, as well as training all branches at the same time. We can see that training all branches simultaneously does not have a significant negative impact on the accuracy of any of the branches. With the current

data, the best data for disparity estimation and panoptic segmentation are even found in the joint training. Although these data may not be optimal due to hardware and data set limitations, there is still no evidence that the design of shared encoder is a bottleneck limiting disparity estimation and panoptic segmentation.

7. Conclusion and Outlook

In the present work, we propose a method that can simultaneously perform disparity estimation and panoptic segmentation based on the task requirement of jointly estimate geometry and semantic. The method is proposed by taking the advantages of Panoptic-DeepLab, PSM and AnyNet work. The design of each part of the method is not only theoretically sound but also unified, lightweight and well-structured in its entirety. Experimentally, we also demonstrate the feasibility of the approach in terms of task requirements and structural compatibility with multiple tasks. This makes the method a strong reference and a starting point for subsequent research. On this basis, we designed two feature fusion modules, AFM and WFM, and further investigated the use of panoptic information to improve the disparity estimation by feature fusion. In addition, through experiments on training strategies, datasets, and structural compatibility, we have obtained a lot of first-hand information on the above topic, which will guide the direction of the subsequent research.

In general, the possible improvement of accuracy by feature fusion in a multi-task framework is still highly desirable. In this work, we experimentally analyze the effects of CFM, AFM and WFM on disparity prediction and compare them with the results without fusion module. The data show that feature fusion can indeed bring a breakthrough to the original single-branch disparity estimation in some aspects. The positive impact of AFM, WFM and CFM on disparity estimation differs due to their different design philosophies. This difference also demonstrates the potential of feature fusion in terms of accuracy improvement. How to cleverly design the fusion module so that all the positive effects can be seen simultaneously can certainly be an important direction for further work.

The experimental results demonstrate that the effect of dataset and training strategy on the results can be very significant. In this thesis, we have mentioned the concept of training strategy several times. As a multi-task network with multi-level structure and multiple branches, this method also exhibits much more complexity in the training setup than a single-task network. A possible solution to deal with this situation may be to use a dynamic training strategy, i.e., by designing functions that incorporate all branches into a dynamic system, such as the focal loss. However, how to further set or even define a new training strategy for a multi-task network is an issue that cannot be ignored in the subsequent work.

The present method still has the possibility of structural optimization. The experiments on DRA show a variety of possibilities for the trade-off between the number of parameters

and the accuracy of the method. How to find a well-founded and interpretable optimal design? This is an interesting question. Moreover, observing the experimental results on multi-stage disparity estimation, it can be seen that the accuracy improvement of stage 2 over stage 1 is not significant. Currently, the multi-scale information of many studies has been designed with 3 levels, while there is no reason to suggest that this is necessary. Is it possible to make some changes, such as keeping only stage 1 and stage 3, to further improve this pyramid structure?

There are still many different perspectives for improving disparity estimation in a multi-task framework. In addition to feature fusion, the use of panoptic information on the loss function is also a promising direction. Methods such as PG-Net have been used to guide the loss function with panoptic information and have achieved good results, which is very informative.

In addition, the improvement of panoptic segmentation using multi-branch information is also a valuable research direction. This work focuses only on the improvement of disparity estimation due to time constraints. Due to the structural symmetry of this method, similar feature fusion schemes can be easily implemented and experimented on panoptic branches.

Bibliography

[1] Max Mehltretter. Uncertainty estimation for dense stereo matching using bayesian deep learning. 2001.

[2] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002.

[3] T Kanade, H Kano, S Kimura, A Yoshida, and K Oda. Development of a video-rate stereo machine, proc. IROS.

[4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[5] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 15–18. IEEE, 2006.

[6] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007.

[7] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[8] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020.

[9] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2021.

[10] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018.

[11] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, 2021.

[12] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *European Conference on Computer Vision*, pages 108–126. Springer, 2020.

[13] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5418, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Prof. Dr.-Ing. habil. Christian Heipke. Photogrammetric computer vision. In *Photogrammetric Computer Vision*, 2020.

[16] Hamid Lesani. Frontiers of information technology & electronic engineering.

[17] Yan Wang, Zihang Lai, Gao Huang, Brian H. Wang, Laurens Van Der Maaten, Mark Campbell, and Kilian Q Weinberger. Anytime stereo image depth estimation on mobile devices. *arXiv preprint arXiv:1810.11408*, 2018.

[18] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[19] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

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

[21] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[22] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019.

[23] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

and Pattern Recognition, pages 1959–1968, 2020.

[24] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019.

[25] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2020.

[26] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2019.

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

[28] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33:22158–22169, 2020.

[29] Xiao Song, Xu Zhao, Liangji Fang, Hanwen Hu, and Yizhou Yu. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *International Journal of Computer Vision*, 128(4):910–930, 2020.

[30] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[33] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[34] Junming Zhang, Katherine A Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. Dispsegnet: Leveraging semantics for end-to-end learning of disparity estimation from stereo imagery. *IEEE Robotics and Automation Letters*, 4(2):1162–

1169, 2019.

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

[36] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Semantic stereo matching with pyramid cost volumes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7484–7493, 2019.

[37] Shuya Chen, Zhiyu Xiang, Chengyu Qiao, Yiman Chen, and Tingming Bai. Sgnet: Semantics guided deep stereo matching. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[38] Shuya Chen, Zhiyu Xiang, Chengyu Qiao, Yiman Chen, and Tingming Bai. Pgnet: Panoptic parsing guided deep stereo matching. *Neurocomputing*, 463:609–622, 2021.

[39] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[40] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[41] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[42] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[43] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 269–284, 2018.

A. The Structure of the Method Mentioned in this Thesis

A.1. The structure of Panoptic-DeepLab

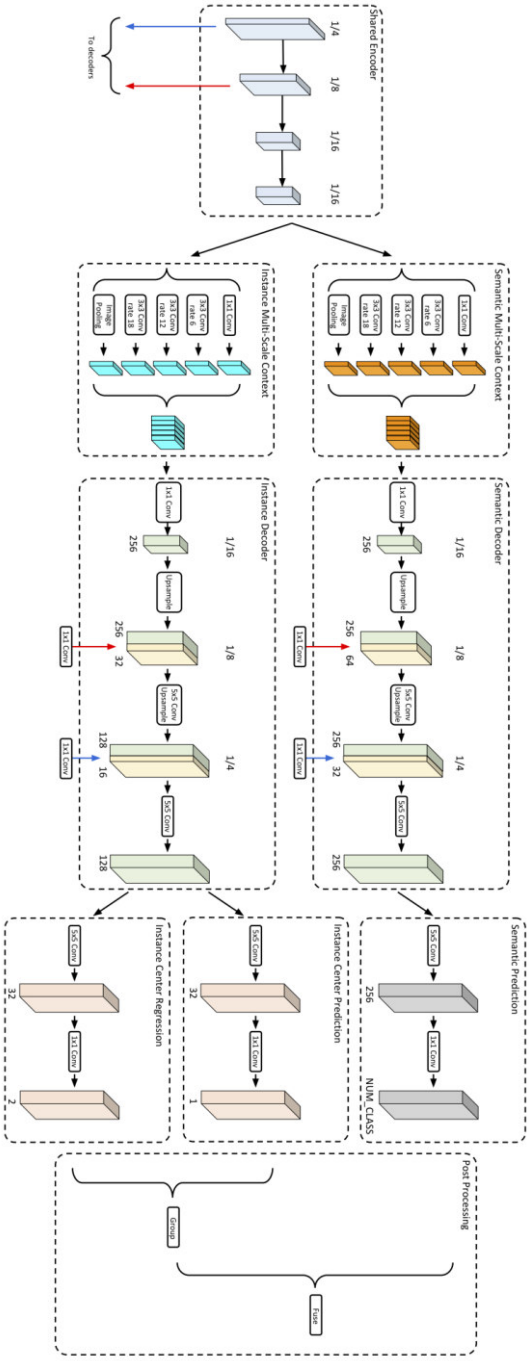


Figure A.1.: The structure of Panoptic-DeepLab [7].

A.2. The structure of the Variant AA-AFM

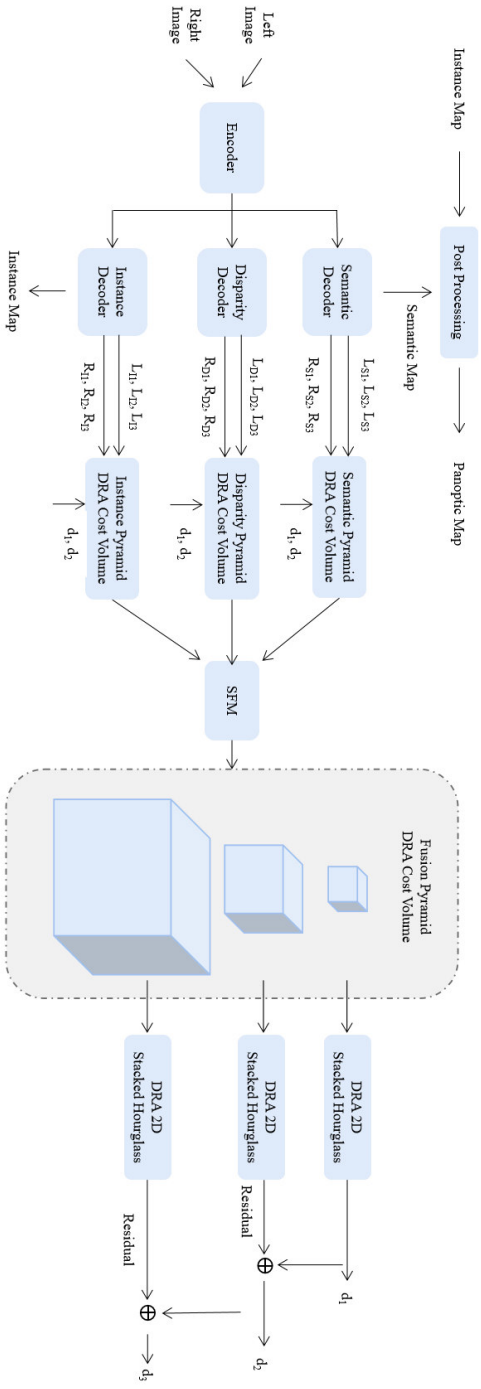


Figure A.2.: The AA (all adapt) with AFM of the proposed method in this thesis. The features of left and right images are extracted by the shared encoder and decoders. L and R represent the features of the left image and the right image. The letters in their footnotes represent the branch they come from, and the numbers in the footnotes represent the stage they were extracted from. The three branches can generate a total of three pyramid cost volumes. These cost volumes can be fused into a new pyramid cost volume by the fusion module. The disparity prediction d of the network can be generated by the disparity regression of the cost volume.

A.3. The structure of panoptic branch and encoder-decoder

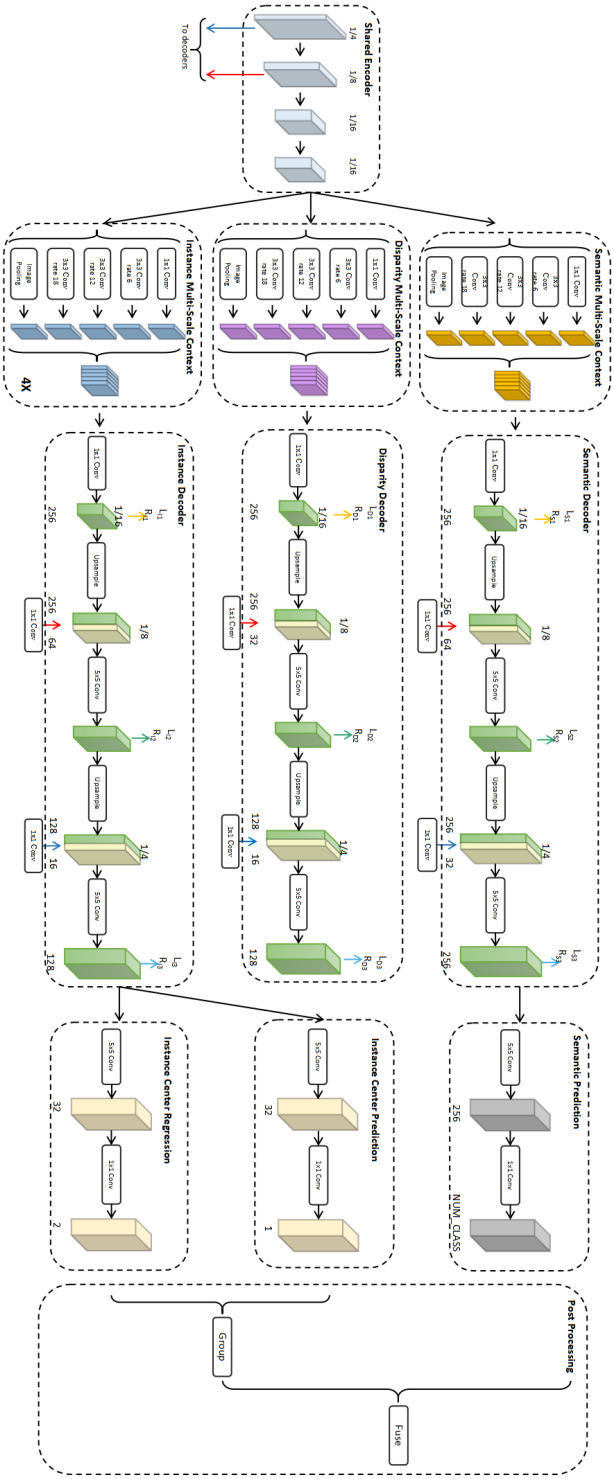


Figure A.3.: The structure of panoptic branch and encoder-decoder for each branch. The panoptic process is identical to the Panoptic-DeepLab. The decoder of the disparity branch is consistent with the semantic branch.