Deep Learning-Based Incorporation of Planar Constraints for Robust Stereo Depth Estimation in Autonomous Vehicle Applications

Wei Qin Chuah, Ruwan Tennakoon, Reza Hoseinnezhad and Alireza Bab-Hadiashar, *Senior Member, IEEE*

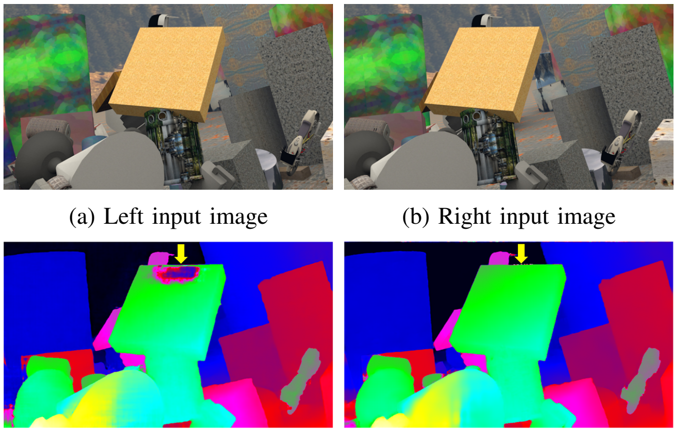
***Abstract*—In autonomous vehicles, depth information for the environment surrounding the vehicle is commonly extracted using time-of-flight (ToF) sensors such as LiDARs and RADARs. Those sensors have some limitations that may potentially degrade the quality and utility of the depth information to a substantial extent. An alternative solution is depth estimation from stereo pairs. However, stereo matching and depth estimation often fails at ill-posed regions including areas with repetitive patterns or textureless surfaces which are commonly found on planar surfaces.**

This paper focuses on designing an efficient framework for stereo depth estimation, using deep learning technique, that is robust against the mentioned ill-posed regions. With the observation that disparities of all pixels belonging to planar areas (scene plane) viewed by two rectified stereo images can be described using affine transformations, our proposed method predicts pixel-wise affine transformation parameters based on the depth information encoded in the aggregated cost volume. We also introduce a propagation term which enforces all pixels belonging to the same scene plane to be transformed using the same parameters. Disparity can then be computed by multiplying the predicted affine parameters with the corresponding pixel locations. The proposed method was evaluated on several benchmark datasets. We are able to obtain competitive results and at the same time reducing the processing time of common convolution neural network (CNN) in stereo matching by 50%. Analysis of the findings shows that our method can produce reliable results at the ill-posed regions which are challenging to the current state-of-the-arts methods.

***Index Terms*—stereo matching, depth estimation, disparity estimation, affine transformation, planar geometry constraint**

I. Introduction

ABILITY to perceive the world in three dimensions (3D) is an important aspect of several autonomous vehicle related tasks including localization and mapping [2], object detection and avoidance [3] and path planning [4]. Precision of depth map correlates with the safety of autonomous vehicles as, high depth precision allows accurate 3D object detection and avoidance [5], [6] and reliable cruise control [7]. Furthermore, accurate localisation of autonomous vehicles rely on having precise 3D positions of selected landmarks [8]. Absence of reliable and precise depth information can therefore lead to catastrophic road accidents.


(a) Left input image

(b) Right input image

(c) PSMNet [1]

(d) Ours

Fig. 1: Performance visualization. (a) and (b) are a pair of challenging input stereo images. (c) Result of the state-of-the-art method PSMNet [1]. (d) Result of our method. Our method is able to rectify erroneous matching and predict a smooth planar disparity surface (region pointed by the arrow).

Common methods employed to infer depth information from the scene include (1) Active sensors: time-of-flight (ToF) sensors such as Light Detection and Ranging (LiDAR) and Radio Detection and Ranging (RADAR) and (2) Passive sensors: dense stereo matching and triangulation from two or multiple images captured from RGB cameras. Although off-the-shelf RGB-D cameras, such as Kinect from Microsoft [9] and RealSense from Intel [10] have simplified capturing depth information in an indoor scene, these sensors often fail at challenging outdoor scenarios or under strong luminance (e.g. sun light) and only provide limited sensing range. Meanwhile, the LiDAR sensor is well known for its accuracy and precision where the captured depth only has errors of an order of centimetres and is currently used in many autonomous vehicles.

Active and passive sensors are often included in a complementary configuration in autonomous vehicles to accomplish various vision tasks, including motion prediction, obstacle detection and avoidance, and depth estimation. However, active sensors have some serious practical limitations: (1) the signal emitting from LiDAR may interfere with other active sensors including other LiDAR sensors [11] and can be absorbed by some materials, (2) there are applications (e.g. military and defence applications) where active sensing is ruled out, (3) the density of the depth map generated using LiDAR or RADAR

W. Chuah, R. Hoseinnezhad and A. Bab-Hadiashar are with the School of Engineering, RMIT University, Melbourne, Australia, (e-mail: wei.qin.chuah@student.rmit.edu.au, {rezah,abh}@rmit.edu.au).

R. Tennakoon is with School of Science, RMIT University, Melbourne, Australia, email: ruwan.tennakoon@rmit.edu.au

Manuscript received April xx, xxxx.

is sparse. For example, the Velodyne HDL-64E LiDAR sensor [12] with resolution of 64 scan lines can only cover about 6% of the total depths of image points which leads to missed detection of object, in particular small objects in a scene [13]. Missed detection can lead to failure in obstacle avoidance and cause tragic road accidents, (5) when compared to RGB cameras, LiDAR is relatively more expensive and it does not provide other additional information besides depth which may be useful. e.g. text on a traffic sign or color of a traffic light which are important when making decisions.

Depth information can be extracted from stereo images based on dense stereo matching algorithms [14]. Conventional dense stereo matching algorithms compute similarity scores between stereo patches in terms of various measures such as mean squared error (MSE), mean absolute difference (MAD) and normalized cross-correlation [14]. Since these methods rely on pixel intensities to compute the similarity scores, they often fail at ill-posed regions such as textureless surfaces or scenes with repetitive pattern or objects.

In recent years, convolutional neural networks (CNNs) have been employed to solve dense stereo matching problems. Instead of relying on RGB color intensities to calculate similarity, CNNs are commonly used to extract features from the patches and the similarity score is computed based on cross-correlation between the features [15], [16]. Methods designed based on such similarity scores outperform conventional methods such as the well-known Semi-Global Matching (SGM) [17] by a large margin. More recently, end-to-end learning models were proposed that can directly perform stereo matching and depth estimation [1], [18]–[22]. While end-to-end models are able to compute dense disparity maps with impressive accuracy, most of the models are computationally expensive. For example, GANet [20] requires 1.8 seconds to process one stereo pair from the KITTI dataset [23].

Despite the significant improvements achieved by end-to-end CNN models, stereo matching at the ill-posed regions remains challenging. Most man-made environments are characterized by planar surfaces, which often appear to be textureless (such as roads and rendered walls) or with repetitive patterns (brick walls) leading to ill-posed stereo matching problems. The illustration in Figure 1c shows an example where a state-of-the-art end-to-end depth prediction method has failed at an ill-posed region.

As demonstrated in [24], [25], transformation of scene planes between two rectified stereo images has three degrees of freedom: (1) scaling, (2) shearing and (3) translation. As such, we should be able to accurately represent the disparities on any planar region using a 3 DoF affine model.

In this work, we propose an end-to-end deep learning model to perform stereo matching and disparity estimation, that combines CNN and geometric constraints (it incorporates an affine disparity model). Affine disparity model is also referred as slanted surface [25] and in this paper we sometimes, when its helpful to the reader, refer to this transformation as the *planar constraint*. Our model takes a pair of stereo images as input and outputs a dense disparity map without any post-processing steps to refine the final output. Unlike [24]–[26], our method does not search for affine disparity models (f_p) explicitly. Instead, we predict f_p for each pixel using the depth information extracted from the stereo images. The proposed method employs a smoothness term which penalizes the disparities of pixels belonging to the same scene plane without following the same parameters f_p. The proposed method is able to resolve ambiguities in the ill-posed regions (see the example in Figure 1d).

The main contributions of this work are three-fold:

- The first point of novelty and significance is the idea of combining CNN and geometric constraints and its development in an end-to-end learning framework that is demonstrably capable of mitigating the effect of matching ambiguities in ill-posed regions.
- Another important contribution is the introduction of a novel smoothness term that penalizes the disparities of pixels belonging to the same scene plane but does not conform to the model having the same parameters f_p.
- The deep model design devised as our proposed solution is itself a major contribution as it effectively requires less burdensome 3D CNN layers and is able to produce better results than the other model with similar structure, GC-Net [18].

The remainder of this paper is organized as follows. Section II describes the related work in the field of learning-based methods for stereo matching and depth estimation (with and without planar constraints). Section III presents the proposed end-to-end learning model for stereo matching and depth estimation with planar constraints. Section III also describes the training method and hyper-parameter selection for the proposed network. Experimental results and discussions are presented in Section IV, and Section V concludes the paper.

II. Related Works

A. Learning-Based Stereo Matching and Depth Estimation

In contrast to traditional methods which rely on RGB color intensities in a local patch between stereo images to perform stereo matching, Zbontar and Lecun [15] proposed a method called MC-CNN that uses the features extracted by a CNN and a series of fully connected layers to compute similarity score or matching cost between the stereo patches. Compared to the traditional methods, the matching costs returned by MC-CNN are robust to ambiguities in photometric appearances while allowing to incorporate local context. Following this work, Luo *et al.* [16] proposed a more efficient solution for stereo patch matching based on computing the inner product of the extracted features from the input patches at each disparity level. However, these methods are inefficient as post-processing steps are often required to refine the initial disparity estimates.

To improve the efficiency, CNN-based end-to-end deep learning models were proposed to perform stereo matching and depth estimation within a single framework [18], [27]. These end-to-end learning models take a pair of rectified stereo images as input and predict a dense disparity map as output without the need for any post-processing. Mayer *et al.* [27] proposed two networks: (1) DispNet, which is inspired by FlowNet [28] that takes the stacked stereo image pairs as inputs and uses a encoder-decoder 2D convolutional network (hourglass) to predict dense disparity map and (2) DispNetC that first correlates left and right feature maps across disparity and the product is passed through an hourglass convolutional network to compute a dense disparity map. Both DispNet and DispNetC outperform MC-CNN and are roughly 1000 times faster [27]. Their results showed that CNN-based end-to-end learning model is able to improve the efficiency significantly without sacrificing the accuracy.

Perpendicular to the above direction, Kendall *et al.* [18] proposed a new method called GC-Net that uses shifted concatenation method to construct the initial cost volume and 3D CNNs to aggregate the cost. Instead of using another layer of CNN or argmax operation to convert the aggregated cost volume into dense disparity map, they used soft-argmax for this task. The methods that construct cost volume using correlation are more efficient than the shifted concatenation as 3D CNNs are more computationally expensive than 2D CNNs, but the latter outperforms the former by a large margin. This is because the cost volume constructed using shifted concatenation retains all the features, thus containing more information for the 3D CNNs to learn to compute matching costs from these features while correlation loses information due to dimension reduction [22].

Following Kendall *et al.* [18]'s footsteps, Chang *et al.* [1] proposed Pyramid Stereo Matching Network (PSMNet) that includes a spatial pyramid pooling (SPP) module to incorporate hierarchical context information [29] and a 3D stacked hourglass network to regularize the network. Zhang *et al.* [20] also proposed the GA-Net inspired by the popular traditional method for depth estimation, semi-global matching (SGM). GA-Net shares a network structure similar to the ones introduced in [1], [18] but it incorporates semi-global and locally guided aggregation layers that aggregate cost volume to incorporate global context while preventing the loss of fine details.

On the other hand, several methods were proposed to include context information such as edges and semantics to further improve the performance of disparity estimation. For instance, Song *et al.* [30] proposed EdgeStereo that incorporates edge information extracted from a separate network to improve the accuracy of stereo matching. Yang *et al.* [21] proposed SegStereo that incorporates semantic information extracted by the included semantic segmentation network. Similarly, Miclea *et al.* [31] proposed to improve the performance of a real-time traditional stereo matching method by including semantic features.

B. Stereo Matching and Depth Estimation with Planar Constraint

Disparity and scene points belonging to the same planar region in rectified stereo pairs are well explained by three degrees of freedom transformations (scaling, shearing and translation) [24]. Based on this observation, Bleyer *et al.* [25] proposed a new method called PatchMatch, in which stereo matching is solved using "slanted plane" algorithm [26]. Instead of searching and assigning plane parameters to each pixel from all possible planes whose size is infinite, random initialization is performed with the assumption that at least one pixel of the region is close to the correct one. Both spatial and view extension are then performed to propagate the correct plane hypotheses, between the neighbouring pixels as well as between left and right views. Matching cost at each pixel is computed using the disparity calculated from the assigned plane parameters. Moreover, if the input is a video sequence, temporal propagation can be performed between the current and previous frames.

Several works including [24], [32], [33], have been proposed based on the above method [25] by incorporating PatchMatch into global models and adding explicit smoothness terms to regularize the local neighbourhoods of 3D labels [34]. Also, PatchMatch was integrated into object extraction pipeline to improve the overall accuracy by leveraging disparity information as a prior [35]. Specifically, disparity map of a given stereo image pair is computed by fitting all image pixels to 3D planes using PatchMatch. Object proposals are generated using the computed disparity map. Lastly, a modified PatchMatch is utilized to refine the disparity map and generated object proposals. More recently, Duggal *et al.* [36] proposed DeepPruner, a CNN-based PatchMatch algorithm for stereo matching. DeepPruner models the traditional PatchMatch algorithm into CNN-based learning model, which allows the model to learn how to effectively propagate context information and compute highly accurate cost volume with less computational cost.

Rather than working on pixel level, many researchers also worked on assigning correct plane parameters to segments [37]–[40]. The reference image is first over-segmented into numerous non-overlapping segments based on the similarity in color space. Then, using the initial disparity information obtained from performing stereo matching between the reference and target images, slanted plane is fitted within each segment. Segment-based stereo matching method assumes that (1) the boundaries of the extracted segments coincide with the disparity boundaries and (2) disparity varies smoothly within a segment. Although the first assumption depends on the quality of the input image and the performance of the image segmentation method, it has yielded reliable results in many scenarios. Some prior works also focused on implementing global optimization using Markov Random Field (MRF) [37] or Graph Cuts [39], [40] to minimize the defined energy cost function. In these works, each segment is represented as a graph node.

While global optimization stereo matching methods can produce accurate results, these methods are computationally expensive and have large memory footprints. Alternatively, local stereo matching algorithms such as Block Matching (BM) are highly efficient. However, BM is prone to errors as the algorithm assumes that all disparities within a block must be constant. This assumption is only valid for fronto-parallel areas and is often violated in common scenes. To tackle this problem, Einecke *et al.* [41] proposed to find the best matches of pixels within a block on left image over multiple blocks on right image. For instance, given a left block of size 3×3, the best matches of three pixels (out of the nine pixels) may come from a right block with $disparity - 1$, another three pixels from block with $disparity$ and the rest from block with $disparity + 1$. Similarly, Muresan *et al.* [42] proposed to relax the fronto-parallel assumption by using multiple tilted matching blocks as descriptors rather than relying on square blocks that are parallel to the image plane. The tilted blocks relax the mentioned fronto-parallel constraint and thus able to achieve better accuracy than the traditional BM.

Conversely, instead of relying on exhaustive search for local correspondences between a stereo image pair using a sliding window along the full disparity ranges, Sinha *et al.* [43] proposed to estimate disparity for each pixel using local plane sweep. The proposed method utilizes sparse feature correspondences to propose local planes. These proposed plane hypotheses are then used to perform plane sweep to estimate disparity at each local region. By decomposing the stereo matching problem into multiple local problems can effectively improves the overall efficiency and encourages smooth local surface within each region.

Park and Yoon [44] proposed using the planar constraint to refine the initial disparity maps (irrespective of the method that generated the map). PSMNet [1] was implemented as the base model in their work. From the initial disparity map, both local and global plane hypotheses were generated. To further improve the results, a global optimization step, using hierarchical clustering to group interrelated plane hypotheses, was performed. However, this method strongly depends on the accuracy of initial disparity map generated from the selected network.

C. Combining geometry with CNN

Rather than treating deep neural network as a "black box" that can outperform conventional methods in most computer vision tasks, by instilling the geometry knowledge into the network, one can reason about what is occurring within the network. Moreover, geometry knowledge will act as a regularizer, which will further improve the performance of a given task. For instance, in the context of monocular depth estimation, prior works utilized auxiliary information such as pose estimation [45]–[47] or optic flow [48] to reason the geometry transformation of the scene between two consecutive time frames. A network was proposed in [49] that jointly predicts depth and surface normal, with the aim of embedding depth-surface-normal mutual transformation into constraint consistency between the two outputs. Furthermore, in the context of view synthesis, Liu *et al.* [50] incorporated homography into the proposed neural network to reason the transformation of planar surfaces in the scene and generate a realistic novel view from a single input image.

III. Proposed Method

While most stereo matching and depth estimation methods can produce high-quality and accurate depth estimates, these methods often fail to do so at ill-posed regions, including areas with repetitive patterns or textureless surfaces. These regions are commonly found on planar or flat surfaces (e.g. roads, buildings). This problem motivates us to design a robust solution against the mentioned ill-posed region without sacrificing the accuracy of other areas. We propose an efficient network that exploits the disparity information encoded within an aggregated cost volume to accurately predicts and assigns plane parameters (affine transformation parameters) to all pixels. We also proposed a novel propagation term that rectifies the incorrect predictions by enforcing all pixels belonging to the same scene plane to be transformed using the same parameters.

A. Slanted plane model in rectified stereo images

Given two 3D points that are on the same plane in camera coordinate system, namely point P and point Q, it is known that the respective transformation of those points between the left (p^l, q^l) and the right (p^r, q^r) cameras is related by a homography matrix, H, such that

$$p^r = Hp^l, \tag{1}$$

$$q^r = Hq^l, \tag{2}$$

where p^l and q^l are obtained by projecting P and Q to the image coordinates where the origin coincides with the center of left camera. Note that all projected points are represented in the homogeneous coordinates (e.g. $p^l = [p^l_x, p^l_y, 1]^\top$).

In a rectified stereo image pair, disparity is defined as the horizontal shift of a pixel when transformed from left (target) image to right (reference) image or vice versa. Using equation (1) we can derive:

$$p^l - p^r = (I - H)\, p^l, \tag{3}$$

where the first element of $p^l - p^r$ is the disparity for point P (d_p) which can now be written as:

$$d_p = (1 - h_{11})\, p^l_x - h_{12} p^l_y - h_{13}, \tag{4}$$

where $[h_{11}, h_{12}, h_{13}]$ is the first row of H and $p^l = [p^l_x, p^l_y, 1]^\top$. Similarly for point Q:

$$d_q = (1 - h_{11})\, q^l_x - h_{12} q^l_y - h_{13}. \tag{5}$$

For any points on a scene plane in rectified stereo image pairs, the transformation from image coordinates to disparity has three degrees of freedom. Therefore, for any point, *that lie on the same plane*, we can write the corresponding disparity, d, as:

$$d_p = f_p \cdot p^l, \tag{6}$$

where $f_p = [\theta_x, \theta_y, \theta_0]$ is a constant vector. The relationship between motion and depth is explained in [51] (chapter 3).

The above insight suggests that disparity of all pixels belonging to the same scene plane can be described using an affine transformation. Slanted plane formulation allows our method to accurately represent scenes with any planar surfaces. Furthermore, enforcing this planar constraint enables our model to generate smooth surfaces on planar regions, which is critical in 3D reconstruction tasks. This is fully explained in the following section.

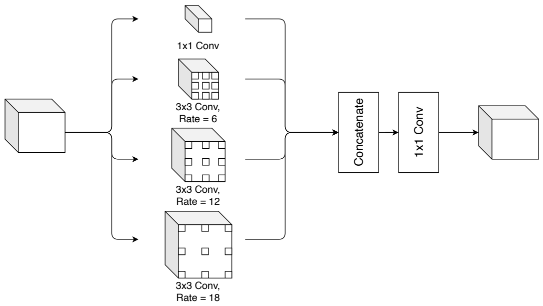


Fig. 2: Visualization of atrous spatial pyramid pooling module (ASPP) implemented in our network. ASPP is able to effectively increase the receptive field and capture global features from the given input which is critical in stereo matching.

B. Disparity regression via plane fitting

Given a pair of stereo images, I_l and I_r, the objective is to design a parametric model capable of performing depth estimation by generating an accurate dense disparity map. Our method first utilizes the depth information encoded in the aggregated cost volume to predict pixel-wise plane parameters, f_p. Next, we incorporate spatial propagation at each pixel to correct any erroneously predicted plane parameters. This correction is performed by enforcing that the disparity at pixel $p = [p_x, p_y, 1]$ (computed using the plane parameters f_q at its neighboring pixels) should be close to the ground truth disparity $\hat{d}_p$:

$$f_q \cdot [p_x, p_y, 1]^\top = \hat{d}_p \qquad \forall q \in \mathcal{N}_p. \tag{7}$$

A pixel $q = [q_x, q_y, 1]$ should be included in the neighbourhood of pixel p ($\mathcal{N}_p$) if both the points are on the same scene plane. We determine this using a weight based on the similarity in the color space between the two pixels in our implementation. More details regarding the spatial propagation loss will be discussed in Section IV (B).

C. Network Architecture

Figure 3 illustrates our proposed network, which consists of a 2D CNN encoder used as a feature extractor, an intermediate 3D CNN hourglass to perform stereo matching and cost aggregation, and a 2D CNN decoder that assigns pixel-wise plane parameters at multiple scales.

1) Feature Extraction and Cost Volume: In contrast to the common practice of computing stereo matching cost using raw pixel intensities, we employ CNNs to extract a deep representation of unary features. Similar to [1], [18], basic residual blocks [52] are implemented to extract the unary features from input stereo images. For feature extraction task, we initially apply a 5×5 convolutional layer, followed by a 3×3 layer without striding. We then include three blocks of stacked CNN layers with a stride of 2 to subsample the input. Each block consists of different numbers of CNN layers with more layers corresponding to the learning of feature representation at smaller scales. The selected number of stack CNNs for each block is $(3, 8, 12)$ with corresponding feature dimensions of $(32, 64, 128)$. Parameters are shared between the left and right feature extraction modules to ensure similar features are extracted from both left and right images.

We also included an Atrous Spatial Pyramid Pooling (ASPP) module to extract hierarchical contextual information and to increase the receptive field of the extracted features effectively. Compared to Spatial Pyramid Pooling [29] adopted in PSMNet [1], ASPP [53] is computationally more efficient at capturing global image context at multiple scales by utilizing atrous convolutional layers. As illustrated in Figure 2, the included ASPP consists of four convolution layers: one 1×1 and three 3×3 convolution layers with different dilation rates, $r \in [6, 12, 18]$.

The extracted left and right unary features are used to construct the cost volume. Shifted concatenation proposed in [18] is performed to build our initial cost volume. The constructed cost volume has a dimension of $[2F, D, H, W]$ where F, D, H, W are the feature size, maximum disparity range, height and weight of the cost volume. Then, a 3D aggregation network is used to aggregate the constructed cost volume.

2) 3D Aggregation Network: Although the 3D CNNs can learn more information-rich context in the cost volume and significantly improve stereo performance compared to the 2D CNNs, the additional dimension is a burden on the computation time for both training and inference [18]. Despite this, state-of-the-art stereo matching networks have many layers of 3D CNNs, either connected in series [20] or stacked hourglass structure [1], [22]. As a result, these methods are capable of producing accurate results at the expense of computational efficiency.

In contrast, we propose to include a lite version of the 3D aggregation network in our model. Our 3D aggregation network only consists of 13 layers of 3D CNNs as compared to 28 layers in PSMNet [1] and GwCNet [22] and 15 layers in GANet [20]. Most of the 3D CNNs in GANet were replaced by the local-guided aggregation (LGA) and semi-global aggregation (SGA) modules, which was proposed in their work. The computation time of the mentioned works for one image is at least 0.4 seconds, while our lightweight implementation is able to reduce the runtime by half. Although our model does not always produce the best results, it delivers a rewarding trade-off between accuracy and efficiency.

The proposed 3D aggregation network consists of two initial cost aggregation blocks (each block has two 3D convolution layers) followed by a 3D hourglass, composed of nine 3D convolution layers. Skip connections are included in the 3D hourglass to retain high-level details, which may be discarded by downsampling the cost volume. The included skip connections are $1 \times 1 \times 1$ 3D convolution layers that do not add much to the overall computation cost. The overall structure of our 3D aggregation network implemented in the proposed network is illustrated in Figure 4. The kernel size of all the 3D convolution layers is $3 \times 3 \times 3$. These layers are followed by batch normalization and rectified linear activation (ReLU) except the last layer of the network.

Also, our 3D aggregation network takes the cost volume constructed via shifted concatenation as mentioned in the previous section as input and output an aggregated cost volume.

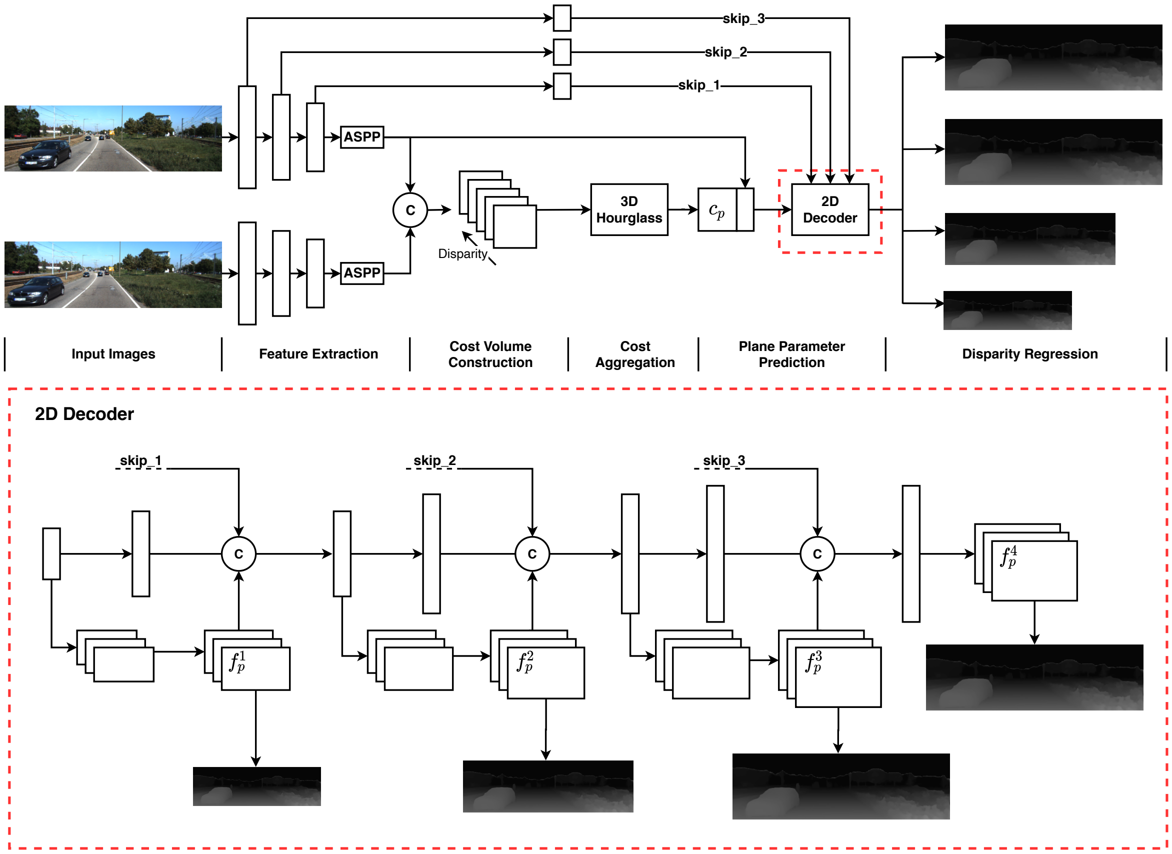


Fig. 3: The visualization of our model. Our model takes a stereo image pair as input and perform feature extraction, cost volume construction and aggregation, pixel-wise plane parameters prediction and disparity regression in sequence and produce multi-scale dense disparity maps as outputs. During inference, the model only outputs the largest scale disparity map. Skip connections from earlier layers are added to the plane parameters prediction phase (2D decoder) to retain high level details in the final outputs. At each scale s, the 2D decoder computes a parameter map with size of $[3, \frac{H}{s}, \frac{W}{s}]$ where H and W are the height and weight of the input image. Disparity map is computed by multiplying the predicted parameter map with pixel locations as shown in equation (6).

The final layer of the 3D aggregation network reduces the dimension of the cost tensor from 5D to 4D, before passing it to the 2D CNN decoder. An initial disparity map is also produced at the end of the 3D aggregation network and is used to compute the initial prediction loss, which is described in Section IV (B).

The initial disparity map is computed from the cost volume using differentiable argmax or soft argmax proposed in [18]. We first convert the predicted similarity score (negative of matching cost, c_p) of each pixel p to probability distribution using softmax operation $\sigma(\cdot)$. We then compute the initial disparity value by taking the sum of each disparity d weighted by the normalized probability $\sigma(-c_p[d])$. Therefore, the initial disparity at pixel p is defined mathematically as follows:

$$d_p^{3D} = \sum_{d=1}^{Dmax} d \times \sigma(-c_p\,[d]), \tag{8}$$

and the softmax operation is defined as

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{D_{max}} e^{z_j}}, \tag{9}$$

where D_{max} is the user-defined maximum disparity range.

Performing disparity regression using soft argmax is able to produce disparities with sub-pixel accuracy as compared to the conventional winner-take-all (WTA) approach that can only output disparities with discrete values [18]. Also, unlike WTA, soft argmax is fully differentiable, allowing the proposed network to be trained in an end-to-end manner.

3) Plane Parameters Prediction (2D Decoder): As previously mentioned, disparity described using affine transformation is robust against erroneous matching in ill-posed regions. As such, we propose to refine the initial disparity map predicted by 3D aggregation network, utilizing the affine transformation. We designed our network to leverage depth information encoded in the cost volume output from 3D aggregation network to predict and assign affine parameters f_p to each pixel. Specifically, the aggregated cost volume is first concatenated with skip features of the same scale. The concatenated cost volume is then fed into the plane parameter prediction network as input. The plane parameter prediction network maps the predicted matching costs of each pixel c_p to an affine parameter f_p. Furthermore, the network also progressively upsamples the predicted affine parameter maps by a factor of 2.

Thus, our network produces an affine parameters map with the size of $[3, H, W]$ as final output where H and W are the height and weight of the input image. Our network also produces multi-scale outputs, which allow it to learn in a "from-coarse-to-fine" manner [54]. The disparity map is computed by multiplying the predicted affine parameters map with the corresponding pixel locations as shown in Equation (6). To retain high level details such as sharp edges, skip connections from the feature extractor layers are added into the network. The architecture of our 2D decoder is inspired by [27] and is illustrated in Figure 3.

D. Loss Function

We formulate our loss function as

$$\mathcal{L} = \sum_{i=1}^{4} \alpha_i (\mathcal{L}_r^i + \lambda \mathcal{L}_p^i) + \beta \mathcal{L}_{init}, \tag{10}$$

where $\mathcal{L}_{init}$ is the initial prediction loss, $\mathcal{L}_r$ is the multi-scale disparity regression loss and $\mathcal{L}_p$ is the propagation loss. The parameters α and β are the loss weights and λ is to balance the influence of the propagation loss term.

1) Disparity Regression Loss, $\mathcal{L}_r$: We adopt the smooth L_1 loss function as the disparity regression loss to train the purposed network. Smooth L_1 loss function has low sensitivity to outliers and is robust at disparity discontinuities, as compared to L_2 loss function. The disparity regression loss term is defined as:

$$\mathcal{L}_r(d, \hat{d}) = \frac{1}{N} \sum_{p=1}^{N} smooth_{L_1}(d_p - \hat{d}_p), \tag{11}$$

where N is the number of valid pixels, d is the predicted disparity and $\hat{d}$ is the ground truth disparity, and

$$smooth_{L_1}(x) \triangleq \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5. & \text{otherwise} \end{cases} \tag{12}$$

2) Initial Prediction Loss, $\mathcal{L}_{init}$: This loss function aims to explicitly train the 3D CNN in our network to learn how to perform the stereo matching task.

$$\mathcal{L}_{init}(d^{3D}, \hat{d}) = \frac{1}{N} \sum_{p=1}^{N} smooth_{L_1}(d_p^{3D} - \hat{d}_p). \tag{13}$$

In the Section VI, we show that by including this loss, our network is able to produce better results. This is largely due to the fact that by including this loss function, the network is able to compute accurate cost volume and provides a better initialization to our 2D decoder. We also adopt the smooth L_1 loss function as the initial prediction loss.

3) Propagation Loss, $\mathcal{L}_p$: As the initial predictions of plane parameters may be erroneous, we propose a novel propagation term to further improve the performance of plane parameters prediction. The propagation term rectifies any erroneous prediction by referring to its related neighboring pixels. Specifically, the propagation term constraints the disparity of all pixels belonging to the same plane to be defined using the same plane parameters. Our propagation loss is defined as:

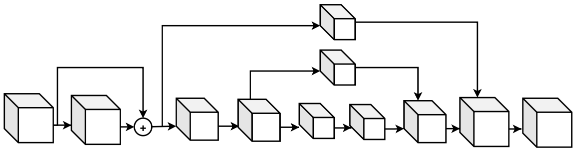


Fig. 4: Visualization of the 3D aggregation network implemented in our work. Our 3D aggregation network only consists of 13 layers of 3D CNNs which effectively reduces the computational burden of the proposed method.

$$\mathcal{L}_p = \frac{1}{N} \sum_{p=1}^{N} \sum_{q \in \mathcal{N}_p} \omega_{pq} |\hat{d}_p - f_q \cdot [p_x, p_y, 1]^\top |, \tag{14}$$

where $\mathcal{N}$ represents the neighbouring pixels of p along four different directions, namely top, bottom, left and right. Therefore, the proposed propagation loss function can be seen as spatial propagation with its neighbouring pixels in four directions along horizontal and vertical directions.

If the target and neighbor pixels are closely related (belong to the same plane), then the predicted parameters should be the same. In other words, disparity at pixel p can be described with plane parameters predicted at pixel q and vice versa. We define the affinity between the two pixels using color similarity weight, ω_{pq} which is defined as:

$$\omega_{pq} \triangleq \exp\left(-||I_l(p) - I_l(q)||\right). \tag{15}$$

E. Implementation Details

All models were trained end-to-end with Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$). Batch size of 16 was used for Scene Flow dataset [27] and KITTI 2015 dataset [23], [55]. Color normalization was included in data preprocessing. Input images were randomly cropped to size $H = 256, W = 512$ during the training phase. The maximum disparity was set to D_{max} = 192. We trained our model on Scene Flow dataset for 19 epochs with learning rate of $1e^{-3}$ for the first 10 epochs, decreased it to $7e^{-3}$ for the remaining epochs. For KITTI 2015, we fine-tuned our model pre-trained on Scene Flow dataset for 600 epochs. The learning rate was set to $1e^{-3}$ for the first 400 epochs then decreased to $1e^{-4}$ for the remaining epochs. We prolonged the training to 1200 epochs to obtain the final model and the test results for KITTI 2015 submission. The parameters α and β are set as $\alpha_1 = 0.6$, $\alpha_2 = 0.8$, $\alpha_3 = 0.9$, $\alpha_4 = 1.0$ and $\beta = 0.6$. The selection of the parameters is accomplished by balancing the contribution of loss function at different scales. For example, α is weighted with the intuition that the largest scale i=4 in equation 10 should be the most important loss to minimize as compared to the losses of smaller scales, $i < 4$. Similar principle also applies to β. All weights have been empirically tested in our experiments.

IV. Experimental Results and Discussion

A. Experimental Settings

We evaluated our method with different settings using Scene Flow and KITTI 2015 datasets. The proposed network was

TABLE I: Evaluation of the proposed method with different settings.

Loss Function			Scene Flow
$\mathcal{L}_r$	$\mathcal{L}_{init}$	$\mathcal{L}_p$	EPE
✓			1.46
✓		✓	1.35
✓	✓		1.28
✓	✓	✓	**1.25**

implemented using PyTorch [56]. The training process took about 13 hours for Scene Flow dataset and 6 hours for KITTI 2015 dataset. We performed ablation studies using both Scene Flow and KITTI 2015 datasets with different settings to evaluate the performance made by affine transformation in disparity prediction, different values of the loss parameter λ and different combinations of loss functions.

For Scene Flow dataset, the end point error (EPE) is used as the evaluation metric (EPE measures the mean average disparity error in all pixels). For KITTI 2015, the percentage of outliers $D1$ is evaluated for background, foreground, and all pixels. The outliers are defined as pixels whose disparity errors are larger than $max(3px, 0.005d^*)$, where d^* denotes the ground-truth disparity. As the KITTI 2015 dataset does not provide the ground truth labels for the testing set and is required to upload the final results to the evaluation server for benchmarking, we performed training by splitting the 200 training images into 160 images for training and 40 for validation.

TABLE II: Performance comparison with Scene Flow dataset. For our proposed method, $\lambda = 1.0$.

	GANet-15	PSMNet	DispNetC	GCNet	Our
EPE	**0.84**	1.09	1.68	2.51	1.25

B. Evaluation on Scene Flow Dataset

1) Dataset: Scene Flow dataset is a large collection of synthetic stereo dataset with dense disparity ground truth. Scene Flow comprises three subsets of datasets with different settings, FlyingThings3D, Driving and Moonkaa. This dataset consists of 35,454 training and 4,370 testing images. The size of each image is 960×540. As the maximum disparity in this dataset is larger than our pre-defined maximum disparity value, D_{max}, any pixel with disparity larger than the D_{max} is neglected in the loss computation.

2) Disparity estimation: The performance of the proposed method is compared with several state-of-the-art approaches including GC-Net [18], DispNet [27], PSMNet [1] and GANet [20]. Table II compares the methods using End-Point Error (EPE). Our method outperforms most of the methods except PSMNet and GANet. As the testing set (subset of FlyingThings3D) contains many complex objects that are difficult to be described using planes such as motorcycles and headphones, our method performs poorly when these objects are present. However, as shown in Figure 5, our method outperforms PSMNet in examples without the mentioned complex objects. Our method is able to predict accurate disparity for the background which is a planar structure with random images. This shows that our method can reliably predict disparities on planes.

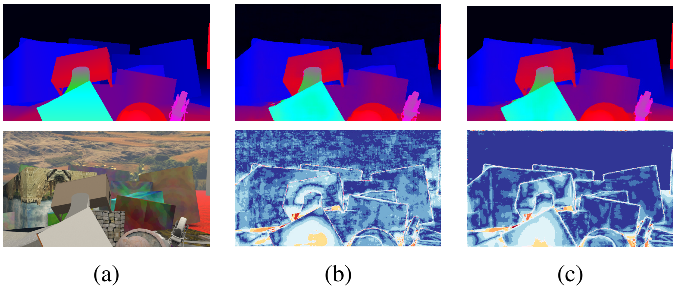


Fig. 5: Performance visualization on Scene Flow dataset. (a) are the left input image (top) and the ground truth (bottom). Top row of (b) and (c) are the outputs of PSMNet [1] and our proposed method. Bottom row of (b) and (c) are the error maps generated using KITTI 3-px metric. Our method is able to learn and predict excellent disparity for planar regions such as the wall of the boxes and the background.

C. Evaluation on KITTI 2015 Dataset

1) Dataset: KITTI 2015 stereo dataset consists of real-world imagery as the data were collected in city and rural area and highways in Karlsruhe, Germany. It contains 200 training stereo image pairs with sparse ground truth disparities collected using LiDAR sensor and 200 testing image pairs without ground truth disparities. The size of each image is 376×1240. KITTI allows performance evaluation by submitting final results to their evaluation server.

2) Fine-tune pre-trained model: The KITTI 2015 dataset is relatively small and consists of only 200 images. Thus, any deep learning model will easily overfit if trained from scratch using this dataset. To mitigate this problem, the model pre-trained on Scene Flow dataset was used to fine-tune on the KITTI 2015 dataset.

3) Disparity estimation: Experimental results demonstrate that our method outperforms other deep learning stereo depth estimation methods as listed in Table III. Although the quantitative results in Table III show that our proposed method does not have the best results as compared to GANet-15 [20] or has slight improvement as compared to the GC-Net [18], our method is able to produce more accurate and smoother disparities and has lower processing time. Qualitative evaluation of KITTI 2015 stereo results[1] is illustrated in Figure 6.

2) Propagation loss function: We conducted several experiments with different propagation loss weight, $\lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ to evaluate the performance of our model. As illustrated in Figure 7, when evaluated in KITTI 2015 dataset, in one case the performance improves as the λ increases (top row) while in another, the performance worsens due to over smoothing (bottom row). As shown in Table IV, λ of 1.0 yielded the second best performance, which has an error rate of 2.44% on KITTI 2015 validation set. Although

[1] Link to our KITTI 2015 submission: PCStereo

TABLE III: KITTI 2015 performance comparison. The included results are obtained from the KITTI 2015 leaderboard. The results show the percentage of pixels with errors of more than three pixels or 5% of disparity error from all test images.

Method	All (%)			Noc(%)			Runtime (s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
DispNetC [27]	4.32	4.41	4.34	4.11	3.72	4.05	**0.06**
SGM-Net [57]	2.66	8.64	3.66	2.23	7.44	3.09	67
Displets v2 [58]	3.00	5.56	3.43	2.73	4.95	3.09	265
MC-CNN-acrt [15]	2.89	8.88	3.89	2.48	7.64	3.33	67
GC-Net [18]	2.21	6.16	2.87	2.02	5.58	2.61	0.9
PSMNet [1]	1.86	4.62	2.32	1.71	4.31	2.14	0.41
GANet-15 [20]	**1.55**	**3.82**	**1.93**	**1.40**	**3.37**	**1.73**	0.36
Ours (PCStereo)	2.39	4.98	2.82	2.23	4.65	2.63	0.2

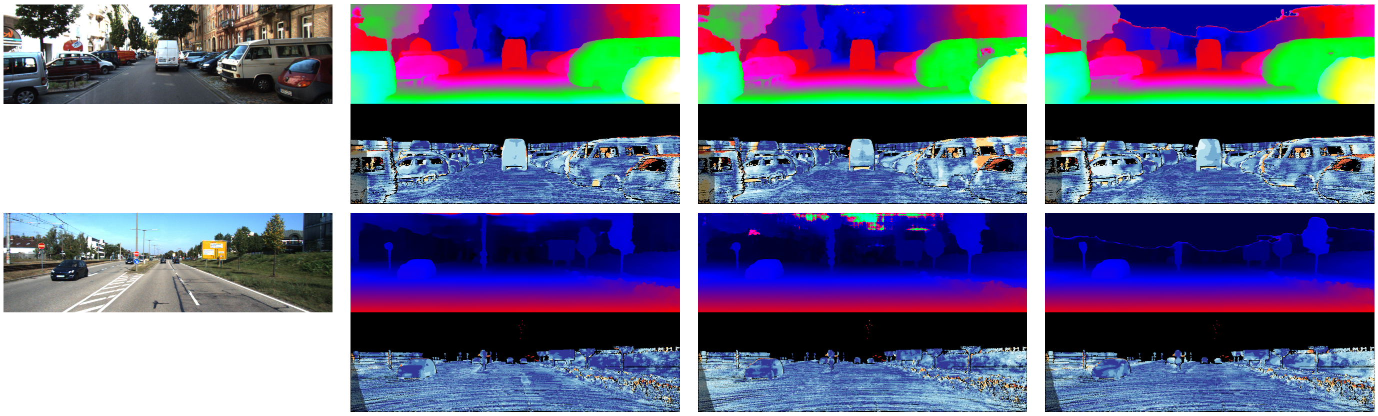

(a) (b) GANet [20] (c) GC-Net [18] (d) Ours

Fig. 6: KITTI 2015 test results. The left column shows the left input image of stereo image pair. For the top example, our method produced better disparity prediction on the surface of the van that is located on the right of the image. For the bottom example, our method produced better and smoother disparity prediction on the car that is located on the left of the image.

TABLE IV: Performance evaluation with different propagation loss weight values. The error rate (%) represents the three-pixel-error on KITTI 2015.

Lambda	KITTI 2015	Scene Flow
	Error Rate (%)	EPE
0.0	**2.40**	1.28
0.2	2.55	1.21
0.4	2.72	**1.11**
0.6	2.50	1.58
0.8	2.47	1.43
1.0	2.44	1.25

the final performance of setting $\lambda = 1.0$ is slightly behind $\lambda = 0.0$, by including the loss propagation term, our model is able to produce a more piece-wise like results as demonstrated in Figure 7 which is important in applications including 3D reconstructions. Furthermore, we demonstrate that our model is capable of generating much smoother flat surfaces as compared to PSMNet [1] by visualizing the generated results in 3D which is shown in Figure 9.

D. Discussion

1) Loss function combinations: In this section, we explore the effectiveness of different combinations of loss functions. Our experiments show that our model is able to produce better results with the *Initial Prediction Loss*, $\mathcal{L}_{init}$ included. The $\mathcal{L}_{init}$ improves the initial prediction in our 3D CNN thus producing a less erroneous cost volume that is then used to predict pixel-wise plane parameters. Our experiments also demonstrate that including the *Propagation Loss*, $\mathcal{L}_p$ further improves the performance of our model. The $\mathcal{L}_p$ loss function enforces local smoothness by constraining the neighbouring pixels to have the same plane parameters as the target pixel when their color similarity is high.

2) Efficiency of the proposed network: In this section, we discuss the efficiency of our proposed network. The proposed plane parameters prediction module is not only robust to challenging scenarios, as mentioned in the previous section, it is also highly efficient. The module requires a processing time of only $10ms$ for each stereo image pair. Meanwhile, the 3D aggregation network is considered the bottleneck of the proposed network as it requires a processing time of approximately $180 - 210ms$ for one stereo image pair. This is due to the costly 3D convolution layers that have cubic computational complexity and high memory consumption. However, the 3D aggregation network can generate accurate initial disparity, which provides reliable initialization for the proposed plane parameters prediction module.

3) Performance of the proposed network on planar surfaces: In this section, we discuss the performance of the proposed network on disparity estimation, specifically on planar surfaces using the KITTI dataset. These planar surfaces include

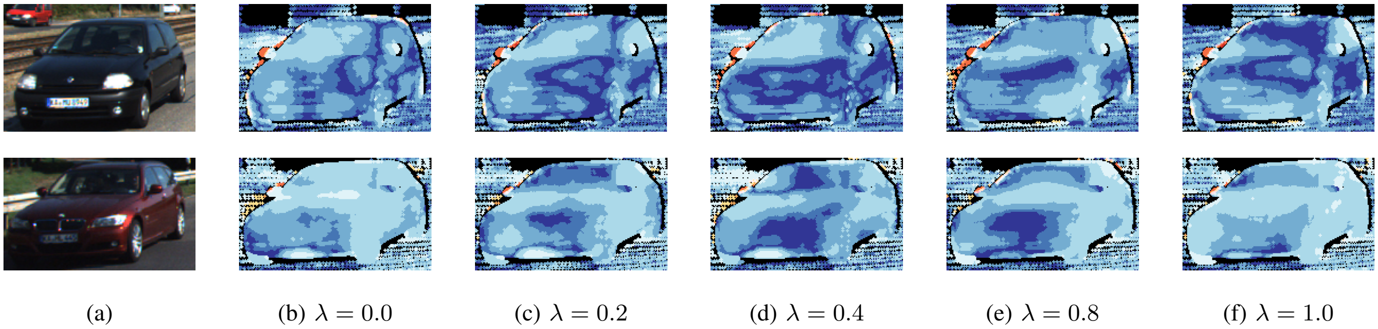
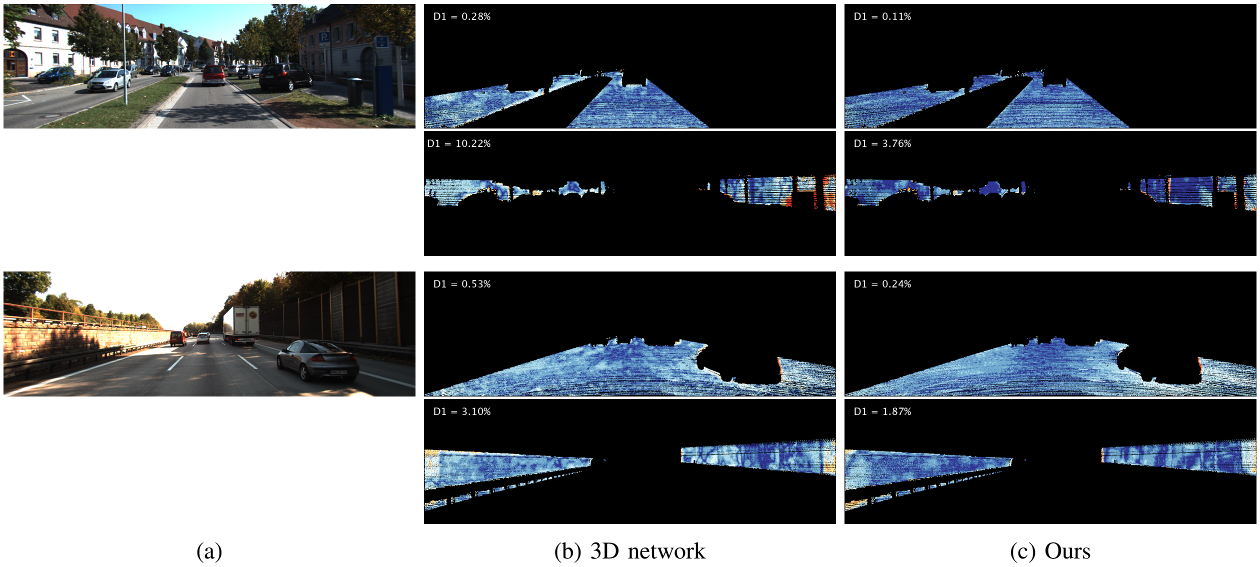

(a) (b) $\lambda = 0.0$ (c) $\lambda = 0.2$ (d) $\lambda = 0.4$ (e) $\lambda = 0.8$ (f) $\lambda = 1.0$

Fig. 7: Visualization of error maps produced using different λ propagation loss weights. Top row: Increase in λ parameter improves disparity prediction. Bottom row: Increase in λ parameter leads to oversmoothing and deteriorate the results.



(a) (b) 3D network (c) Ours

Fig. 8: Visualization of error maps on planar surfaces. These surfaces are extracted using semantic maps provided by KITTI [59]. The results of a network with only 3D aggregation network are shown in the second column and the results of the proposed network are shown in third column. The comparison shows that our proposed network can accurately estimates disparity for planar surfaces. The corresponding D1 error is also included at the top left corner of all error maps.

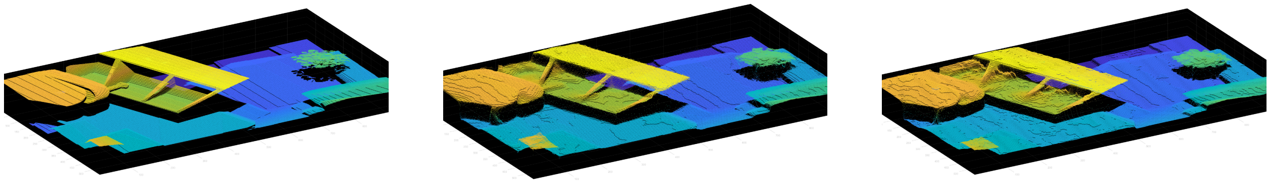

(a) Ground Truth (b) Our (c) PSMNet [1]

Fig. 9: 3D illustration of the predicted depth. Our method is able to produce smoother flat surfaces such as the side wall of the shelf and the background as compared to the PSMNet.

walls of buildings and roads. We compare the results of two different networks: (1) the proposed network and (2) the proposed network without the plane parameters prediction module (feature extraction module and 3D aggregation network only). As shown in Figure 8, our proposed network has better performance as it can constantly generate disparity estimates with higher accuracy on the planar surfaces. Also, by including the proposed smoothness loss function, the network can produce significantly smoother disparity estimates on the planar surfaces.

V. Conclusion

In this paper, we proposed an end-to-end learning model for stereo matching, which incorporates geometry planar constraint in its framework. Instead of focusing on designing better matching algorithm or aggregation of cost volume, we seek to accurately model planar regions in the scene and in disparity space using the affine transformation (slanted plane).

We demonstrated that our method is particularly useful when the visual driving system is used in places that include many planar objects (e.g. urban landscape). We have designed a novel network architecture specifically to predict and assign the transformation parameters to each pixel. We also proposed a novel propagation loss function that ensures local smoothness by enforcing same parameters are assigned to all pixels belonging to the same plane. Experiments demonstrated the effectiveness of our proposed method on the Scene Flow and the KITTI 2015 datasets.

References

[1] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5410–5418.

[2] H. Lategahn, A. Geiger, and B. Kitt, "Visual slam for autonomous ground vehicles," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1732–1737.

[3] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara, "Obstacle avoidance and path planning for humanoid robots using stereo vision," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 592–597.

[4] T. Cao, Z.-Y. Xiang, and J.-L. Liu, "Perception in disparity: An efficient navigation framework for autonomous vehicles with stereo cameras," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2935–2948, 2015.

[5] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8445–8453.

[6] S. Nedevschi, S. Bota, and C. Tomiuc, "Stereo-based pedestrian detection for collision-avoidance applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 380–391, 2009.

[7] Y. Fang, I. Masaki, and B. Horn, "Depth-based target segmentation for intelligent vehicles: Fusion of radar and binocular stereo," *IEEE transactions on intelligent transportation systems*, vol. 3, no. 3, pp. 196–202, 2002.

[8] A. De la Escalera, J. M. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," *Image and vision computing*, vol. 21, no. 3, pp. 247–258, 2003.

[9] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[10] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1–10.

[11] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: common practices and emerging technologies," *arXiv preprint arXiv:1906.05113*, 2019.

[12] D. S. Hall, "High definition lidar system," Jun. 28 2011, uS Patent 7,969,558.

[13] K. Park, S. Kim, and K. Sohn, "High-precision depth estimation using uncalibrated lidar and stereo fusion," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[14] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[15] J. Žbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *The journal of machine learning research*, vol. 17, no. 1, pp. 2287–2318, 2016.

[16] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5695–5703.

[17] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 807–814.

[18] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 66–75.

[19] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 887–895.

[20] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "Ga-net: Guided aggregation net for end-to-end stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 185–194.

[21] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "Segstereo: Exploiting semantic information for disparity estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 636–651.

[22] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise correlation stereo network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3273–3282.

[23] M. Menze, C. Heipke, and A. Geiger, "Object scene flow," *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[24] P. Heise, S. Klose, B. Jensen, and A. Knoll, "Pm-huber: Patchmatch with huber regularization for stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2360–2367.

[25] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo-stereo matching with slanted support windows." in *Bmvc*, vol. 11, 2011, pp. 1–11.

[26] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," in *ACM Transactions on Graphics (ToG)*, vol. 28, no. 3. ACM, 2009, p. 24.

[27] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[28] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[30] X. Song, X. Zhao, L. Fang, H. Hu, and Y. Yu, "Edgestereo: An effective multi-task learning network for stereo matching and edge detection," *International Journal of Computer Vision*, pp. 1–21, 2020.

[31] V.-C. Miclea and S. Nedevschi, "Real-time semantic segmentation-based stereo reconstruction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1514–1524, 2019.

[32] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "Pmbp: Patchmatch belief propagation for correspondence field estimation," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 2–13, 2014.

[33] C. Olsson, J. Ulén, and Y. Boykov, "In defense of 3d-label stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1730–1737.

[34] C. Zhang, Z. Li, R. Cai, H. Chao, and Y. Rui, "As-rigid-as-possible stereo under second order smoothness priors," in *European Conference on Computer Vision*. Springer, 2014, pp. 112–126.

[35] M. Bleyer, C. Rhemann, and C. Rother, "Extracting 3d scene-consistent object proposals and depth from stereo images," in *European Conference on Computer Vision*. Springer, 2012, pp. 467–481.

[36] S. Duggal, S. Wang, W.-C. Ma, R. Hu, and R. Urtasun, "Deeppruner: Learning efficient stereo matching via differentiable patchmatch," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4384–4393.

[37] T. Yan, Y. Gan, Z. Xia, and Q. Zhao, "Segment-based disparity refinement with occlusion handling for stereo matching," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3885–3897, 2019.

[38] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 15–18.

[39] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, 2004, pp. I–I.

[40] M. Bleyer and M. Gelautz, "Graph-based surface reconstruction from stereo pairs using image segmentation," in *Videometrics VIII*, vol. 5665. International Society for Optics and Photonics, 2005, p. 56650U.

[41] N. Einecke and J. Eggert, "Block-matching stereo with relaxed fronto-parallel assumption," in *2014 Ieee Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 700–705.

[42] M. P. Muresan, S. Nedevschi, and R. Danescu, "A multi patch warping approach for improved stereo block matching," in *International Conference on Computer Vision Theory and Applications*, vol. 7. SCITEPRESS, 2017, pp. 459–466.

[43] S. N. Sinha, D. Scharstein, and R. Szeliski, "Efficient high-resolution stereo matching using local plane sweeps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1582–1589.

[44] M.-G. Park and K.-J. Yoon, "As-planar-as-possible depth map estimation," *Computer Vision and Image Understanding*, vol. 181, pp. 50–59, 2019.

[45] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5667–5675.

[46] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3828–3838.

[47] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 340–349.

[48] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992.

[49] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "Geonet: Geometric neural network for joint depth and surface normal estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 283–291.

[50] M. Liu, X. He, and M. Salzmann, "Geometry-aware deep network for single-image novel view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4616–4624.

[51] M. Subbarao, "3 - formulation," in *Interpretation of Visual Motion*, ser. Research Notes in Artificial Intelligence, M. Subbarao, Ed. Morgan Kaufmann, 1988, pp. 21 – 31. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780273087922500084

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[53] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[54] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous crfs as sequential deep networks for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5354–5362.

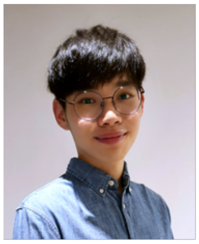
[55] M. Menze, C. Heipke, and A. Geiger, "Joint 3d estimation of vehicles and scene flow." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 2, 2015.

[56] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[57] A. Seki and M. Pollefeys, "Sgm-nets: Semi-global matching with neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 231–240.

[58] F. Guney and A. Geiger, "Displets: Resolving stereo ambiguities using object knowledge," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4165–4175.

[59] H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision (IJCV)*, 2018.



WeiQin Chuah received the B.Eng.(Hons) degree in Adv Manufacturing and Mechatronics in 2018, and he is currently pursuing the Ph.D. degree in engineering at the Royal Melbourne Institute of Technology (RMIT). His research interests include computer vision, stereo matching and depth estimation systems, machine learning and deep learning, autonomous driving, and related applications.

Ruwan Tennakoon obtained his PhD degree in computer vision from Swinburne University of Technology, Australia (2015) and his BSc degrees (with fists class honours) in Electrical & Electronics Engineering from University of Peradeniya, Sri Lanka (2007). From 2015, Ruwan worked as a Research Fellow at RMIT School of Engineering developing computer vision based driver assist technologies for industrial vehicles. He has also worked as a Research Scientist at IBM-Research Australia. His main research interests include computer vision, machine learning and medical image analysis.

Reza Hoseinnezhad received his PhD in 2002 then held various positions at Swinburne University of Technology, The University of Melbourne, and RMIT University, where he is Associate Dean (Mechanical & Automotive Engineering). His main research interests include statistical information fusion, random finite sets, multi-object tracking, deep learning, and robust multi-structure data fitting in computer vision.

Alireza Bab-Hadiashar received the B.Sc. and M.Eng. degrees in mechanical engineering and the Ph.D. degree in robotics from Monash University. He has held various positions in Monash University, the Swinburne University of Technology, and RMIT University, where he is currently a Professor of mechatronics and leads the Intelligent Automation Research Group. His main research interests include intelligent automation in general, robust data fitting in machine vision, deep learning for detection and identification, and robust data segmentation.