PlaneDepth: Plane-Based Self-Supervised Monocular Depth Estimation

Ruoyu Wang, Zehao Yu, Shenghua Gao

Abstract—Self-supervised monocular depth estimation refers to training a monocular depth estimation (MDE) network using only RGB images to overcome the difficulty of collecting dense ground truth depth. Many previous works addressed this problem using depth classification or depth regression. However, depth classification tends to fall into local minima due to the bilinear interpolation search on the target view. Depth classification overcomes this problem using pre-divided depth bins, but those depth candidates lead to discontinuities in the final depth result, and using the same probability for weighted summation of color and depth is ambiguous. To overcome these limitations, we use some predefined planes that are parallel to the ground, allowing us to automatically segment the ground and predict continuous depth for it. We further model depth as a mixture Laplace distribution, which provides a more certain objective for optimization. Previous works have shown that MDE networks only use the vertical image position of objects to estimate the depth and ignore relative sizes. We address this problem for the first time in both stereo and monocular training using resize cropping data augmentation. Based on our analysis of resize cropping, we combine it with our plane definition and improve our training strategy so that the network could learn the relationship between depth and both the vertical image position and relative size of objects. We further combine the self-distillation stage with post-processing to provide more accurate supervision and save extra time in post-processing. We conduct extensive experiments to demonstrate the effectiveness of our analysis and improvements.

Index Terms—Monocular depth estimation, Self-supervised learning, Plane geometry.

I. INTRODUCTION

MONOCULAR depth estimation (MDE) is an ill-posed problem that tries to estimate a per-pixel depth map from a single image. However, many works could achieve good accuracy using deep neural networks under ground-truth supervision [1]–[5], suggesting that estimating depth from monocular cues is feasible. These well-trained models are widely used for autonomous driving and provide depth information as a sub-block in many other tasks.

Since it is hard to get dense and accurate ground-truth depth data, self-supervised learning from videos or stereo image pairs is essential to overcome this limitation. As analyzed by [6], many previous works got pretty results using depth regression [7]–[11]. They used the estimated source view depth to sample RGB from the target view. However, depth optimized by the local search method of bilinear interpolation in the target view easily falls into the local minimum. Recent influential works used depth classification, similar to multi-layer image (MPI), to train networks and got better results [12], [13]. These methods greatly alleviate the problem of local minima and predict sharper depth to the edges of objects by using pre-divided depth bins. Due to the advantage of the plane, these methods enable rendering multi-view as supervision. However, since they divide the depth space into fixed values, essentially dividing the scene into many planes parallel to the camera plane, they cannot predict continuous depth for continuous surfaces not parallel to the camera plane.

Especially, the ground is treated the same as other vertical objects. Generating discontinuous and vertical-prone depth for the ground will cause unnecessary limitations in other fields. Especially in autonomous driving, it is harmful to confuse the ground with other objects and predict discontinuous depth. We propose a novel plane-based representation beneficial for representing outdoor scenes. We add ground planes to the MPI vertical planes. The paralleled to the ground plane has a natural advantage in representing the ground, the most common plane in outdoor scenes. Therefore, our network can automatically segment the ground and predict continuous depth by choosing whether to use the ground planes.

Typically, classification tasks will choose one of the discrete candidate values as the final result. But in order to make the result smoother and more accurate, many previous depth classification methods use the expectation of depth as the prediction result [12], [13]. However, most self-supervised depth and multi-view stereo methods optimize predicted probabilities by weighted summation of projected colors to obtain a synthesized target view and compute losses. Due to the ambiguity of the weighted summation of colors, different weight combinations may result in the same synthesized color, which will generate an uncertain optimization objective, resulting in a non-trivial solution. Also, since the color space and depth space are not uniform, it is unreasonable to use the same weights for the summation of depth and color. Our method models the depth distribution as a mixture Laplace distribution, with each plane as the mean of a single Laplace distribution. During optimizing, we compute the photometric error independently on the color projected by each plane,
resulting in a certain optimization objective and no weighted summation of colors.

As for the neglect of the relative size of objects, generally, the relative size and relationship between objects are important cues to predicate depth. Most of the previous MDE networks only focus on the vertical image position of objects, or the relative relationship of objects to the ground, while ignoring the cue of object size \[14\]. Some works use innovative data augmentations that alleviate this problem in stereo training \[11\]. Resize cropping is a widely used data augmentation in computer vision. It also is used in the stereo training setting of MDE \[12\], \[13\]. It destroys the relationship between the vertical image position and depth and encourages the networks to focus on the relative size of objects. However, previous works can only exploit the relative size cue in stereo training. We deeply analyze the effects of resizing cropping on world coordinates, get the resize cropping matrix, and extend this data augmentation to more settings, such as the monocular/video training. We then combine it with our method based on these analyses and propose a novel training strategy so that the network could learn both relative object size and vertical image position cues simultaneously.

The self-distillation stage is often used to improve prediction results \[11\], or to solve occlusion problems \[12\], \[13\]. Post-processing can also improve prediction results, but consume twice as much inference time. Therefore, we propose to combine post-processing with self-distillation. We use post-processing to improve the supervision of self-distillation, allowing the network to directly predict results similar to post-processing while solving the occlusion problem.

To summarize, our main contributions are:

1) We propose PlaneDepth, a plane-based monocular depth estimation network, which could separate the nearby ground from other objects, predicting smooth depth for the ground while preserving the sharp boundaries of other objects. This plane-based network could also retain the advantages of the plane in other tasks.

2) We propose to solve the problem of depth classification by optimizing the mixture Laplace distribution to fit the distribution of depth. This optimization method avoids the weighted summation of colors, which gives a certain optimization objective, leading to better and more stable results.

3) We are the first to have the network use object relative size cue in self-supervised monocular training. We deeply analyze the nature of resize cropping in depth estimation tasks and construct the resize cropping matrix to extend it to more settings. We further make it applicable to our network and improve our training strategy based on our analysis so that the network could learn the relationship between both relative size and vertical image position cues and depth.

4) We combine post-processing with self-distillation so that the original output of the network is closer to the post-processed output to save inference time while solving the occlusion problem.

II. RELATED WORKS

A. Supervised Monocular Depth Estimation

Many previous supervised monocular depth estimation works have achieved good performance using well-designed network structures and loss functions. \[15\] predicts coarse-to-fine depth maps using a multi-scale network. \[16\] uses a fully convolutional network combined with up-convolution blocks based on ResNet \[17\] to improve the performance. \[3\] uses the predefined depth space and ordinal regression prediction method. They discuss the impact of different depth partitions and carefully designed the loss and prediction method. This classification-like method greatly improves performance. \[5\] also uses the method of dividing the depth space and predicting the probability volume to obtain the depth. Instead of using the classical log space to divide the depth, they use a trainable division for every single image, achieving state-of-the-art performance combined with their well-designed network structure. However, these supervised methods require a large number of accurate labels, which take a lot of time to acquire with specialized equipment.
B. Self-supervised Monocular Depth Estimation

1) Depth Regression: Many previous methods achieve amazing results by directly regressing depth [7]–[11]. They use the predicted source view depth to sample colors from the target view, reconstruct the source view and compute loss with the source view image. However, since they use bilinear interpolation in the target view to make depth regression differentiable, depth could only be optimized with information of nearby pixels and falls into local minimum easily. [9] alleviated this problem by semi-global matching (SGM) supervision when trained with stereo pairs. [8] addressed the occlusion problem by selectively using multi-view information. [10] improves the prediction of textureless regions by using an auto-encoder. [11] proposed a data augmentation for stereo pairs to solve the problem that MDE only focuses on the vertical image position of objects. When trained with monocular sequences or videos, [8], [10] use an additional pose network to predict camera motion, while [18], [19] get camera motion via COLMAP [20]. The multi-scale output allows the network better to learn depth features [8], [11], while Self-distillation enables the student network to retain the best results synthesized from multiple outputs of the teacher network [11], [21]. However, due to the local search on the target view, these methods all suffer from the problem of local minima. Also, since continuous depth is hard to warp, these methods are difficult to address the occlusion problem in stereo training. Our classification-based method addresses the problem of local minima due to pre-divided depth space. We can also efficiently warp probability volume to solve the occlusion problem due to the advantage of plane.

2) Depth Classification: Selecting depth from pre-divided bins can be seen as a classification task. [12], [13] achieved better performance using this method since it addressed the problem of local minima. They innovatively used resize cropping data augmentation and solved the occlusion problem on the stereo pairs. These methods can be seen as initializing some planes parallel to the camera plane, just like MPI. While [19] used MPI combined with homography and differentiable rendering to get good novel view synthesis and depth results. Self-supervision has been used in [12], [13] to solve the occlusion problem in the second training stage. However, these bins- MPI-based network will predict vertical-prone depth for the ground. Our method can automatically separate the nearby ground, distinguish it from vertical objects, and predict smooth depth for it. We use mixture Laplace loss to provide a more certain objective for optimization than previous indirect supervision of the probability volume. Besides, We extend resize cropping to be the first to address the problem of ignoring the relative size in monocular training, and we further improve the training strategy to exploit both cues simultaneously for monocular depth estimation.

C. Layered Geometry

There are also other plane-based methods used for depth estimation. [22] estimates the accurate geometry for indoor scenes by constraining the pixels within a superpixel to be in the same plane. Layered geometry is also widely used in novel view synthesis. In addition to the MPI-based classification methods [1], [19], there are also methods that combine regression and classification based on layered depth image (LDI) [23], [24]. Unlike pre-dividing the depth space, LDI-based methods first regress the multi-layered depth maps and then classify. These methods are usually based on expensive point cloud warping and z-buffer to synthesize novel view...
A. Plane Prediction

In previous works, inverse depth space \cite{12, 13} is often used to divide depth bins. \cite{12, 13} obtained good results using exponential depth space. These divisions are equivalent to pre-defining \( N \) planes parallel to the camera plane and then classifying each pixel to the plane it belongs. However, these MPI-based methods will predict vertical-prone depth, which is disadvantageous for continuous surfaces that are not parallel to the camera plane, especially the ground, the most common plane in outdoor datasets. In many tasks, especially autonomous driving, it is important to recognize the nearby ground and predict its smooth depth. To segment the nearby ground and generate continuous depth for it, we propose the ground plane.

Without resize cropping, since the camera’s x-axis (car horizontal direction) and z-axis (car heading direction) in outdoor datasets are often approximately parallel to the ground, the ground is a plane approximately perpendicular to the y-axis.

Define ground planes as (\( 1 \))

\[
\mathbf{n}^T \mathbf{w} - h_i = 0, \quad \mathbf{n} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad i = 0, 1, \ldots, N_g - 1
\]  

(1)

where \( h_i \) is the distance from the camera to the i-th plane along the y-axis, \( \mathbf{n} \) is the normal and \( \mathbf{w} \) is the point in the world coordinates. We divide \( N_g \) ground planes in linear space as (\( 2 \))

\[
h_i = h_{\text{min}} + \frac{i + r_{vi}}{N_g - 1} (h_{\text{max}} - h_{\text{min}}), \quad i = 0, 1, \ldots, N_g - 1
\]  

(2)

where \( h_{\text{min}} \) and \( h_{\text{max}} \) are the minimum and maximum camera height hyper-parameters, \( r \in (-0.5, 0.5) \) is plane residual which we will discuss later. In this way, we can use only one parameter \( h_i \) to represent the entire ground plane, just like vertical planes. Since the ground plane has the natural advantage of being parallel to the ground, each ground pixel will be classified into one of the ground planes, just like classifying vertical objects into vertical planes.

The depth \( D \) and disparity \( d \) of each pixel on the ground plane can be obtained by (\( 3 \))

\[
D_{yi} = \frac{2h_i F_y}{g_y}, \quad d_{yi} = \frac{BW F_x g_y}{2h_i F_y}
\]  

(3)

where \( \mathbf{g} = (g_x, g_y) \in [-1, 1]^2 \) is the pixel grids, \( B \) is the baseline length of the stereo pair cameras. \( F_x = \frac{F}{w} \), where \( F_x \) is the horizontal focal length of the camera and \( W \) is the width of image.

We initialize the vertical planes similarly to \cite{12, 13} as (\( 4 \))

\[
d_{vi} = d_{\text{max}} (\frac{d_{\text{min}}}{d_{\text{max}}})^{\frac{i + r_{vi}}{N_v}}, \quad i = 0, 1, \ldots, N_v - 1
\]  

(4)

Since objects in the scene cannot all fall exactly on the \( N = N_g + N_v \) fixed candidate planes, taking only \( N \) values in the entire depth space will bring limitations. Therefore we predict independent initialization planes for each scene to ensure that the distribution of planes is the most appropriate. In order to maintain the excellent performance of exponential space, we just add a residual \( r \) to the initial integer level \( i \) so that our plane can theoretically be initialized at any value within the value range.

In the warping stage, we use homography to warp all planes to the novel view to synthesize the novel view image and compute loss. The homography correspondence between the pixel coordinates of the source view and the target view is as (\( 5 \))

\[
D_t \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = K (R + \frac{1}{s} (\mathbf{t} \mathbf{n}^T)) K^{-1} D_s \begin{bmatrix} u_s \\ v_s \\ 1 \end{bmatrix}
\]  

(5)

where \( \mathbf{p} = (u, v) \) is the pixel coordinates, \( K \) is the camera intrinsic parameter matrix and \( R \) and \( \mathbf{t} \) is the rotation translation of camera from the target view to the source view. We compute the corresponding coordinates of the source view for each pixel in the target view and warp planes by bilinear sampling.
The final depth estimated by our network \( \hat{D} \) is the expectation of depth distribution across all planes as \( \hat{D} = \sum_{i=0}^{N-1} p_i D_i \) \( \text{(6)} \), where \( N = N_g + N_v \) and \( p_i \) is the probability that the depth is on the \( i \)-th plane.

B. Resize Cropping in Depth and Egomotion

The relative size and the vertical image position of the object are both important cues in monocular depth estimation. Many MDE networks only predict depth from the vertical image position of the object, ignoring the effect of the relative size. In stereo training, \([11]\) proposed to graft two images vertically and \([12]\), \([13]\) used resize cropping to solve this problem. In fact, training with a fixed translation in the stereo setting without rotation is a special case of resizing cropping. We further extend this data augmentation and address the problem of ignoring relative size in monocular training for the first time.

Under the assumption that the camera parameters are constant, resize cropping causes a transformation of world coordinates, which encourages the network to predict different results, but also causes the ground to no longer be perpendicular to the y-axis. We will analyze these effects so that this data augmentation can be deeply integrated with our method.

In stereo vision, depth and disparity satisfy the relationship as in \( \text{(7)} \).

\[
D = \frac{BF_w W}{d} \quad \text{(7)}
\]

\( \text{(7)} \) shows that depth is only related to the disparity on the epipolar line when the camera system is constant. Compared to depth, disparity is uniform with the relative size of objects, which enables the network to handle inputs of different resolutions. As shown in Fig. 3, HR is three times the resolution of LR, and the depth of the white car in HR is three times that in LR, so the white cars in the two images are the same size at the pixel level. It is difficult for CNNs to predict different depth for white cars of the same size. Since the disparity is uniform with the relative size of the object, we only need to predict the same disparity \( d \) for the white cars, and the depth \( D_H \) will be three times of \( D_L \) because \( W_H \) is three times of \( W_L \). Since we take the relative size of objects as the main cue, all our networks take disparity as output.

Fig. 3. Objects at different depth in different resolution images have the same size and disparity.

From the above analysis, it seems that the network can handle inputs of different resolutions simply by learning the relationship between the relative size of objects and disparity. However, without special data augmentation, TABLE \([11]\) shows that networks has difficulty handling inputs at different resolutions, which together with \([14]\) indicate that networks tend to learn the vertical position of objects and ignore the relative size. Since resize cropping can disrupt the vertical position cue of object, we will use it to solve this problem.

We now define the pixel grid \( g = (g_x, g_y) \in [-1, 1]^2 \) to represent the position of the pixel after the resize cropping in the original image. We can get the scaling factor from \( g \), so the changed disparity is given by \( \text{(8)} \).

\[
d^n = \frac{2d}{\Delta g_x} \quad \text{(8)}
\]

where \( d^n \) is the disparity after resize cropping, \( \Delta g_x = \max(g_x) - \min(g_x) \) implies twice the horizontal scaling factor.

However, in monocular vision, the horizontal direction is no longer special, and we should keep the normal aspect ratio for objects. Define the image scaling factor \( f \) as \( \text{(9)} \).

\[
f = \frac{\Delta g_x}{2} = \frac{\Delta g_y}{2} \quad \text{(9)}
\]

Therefore the disparity \( d^n \) and depth \( D^n \) after resize cropping is given by \( \text{(10)} \).

\[
d^n = \frac{d}{f}, D^n = f D \quad \text{(10)}
\]

In addition to disparity and depth, resize cropping changed the pixel coordinates \( p = (u, v) \). Define \( g^c = \frac{\max|g| + \min|g|}{2} \) to indicate the center of the selected resize cropping area in the original image. The new pixel coordinates \( p^n \) are given by \( \text{(11)} \).

\[
p^n = \frac{2p - S \odot (g^c - 1 + f)}{2f} \quad \text{(11)}
\]

where \( S = (W, H) \) is the size of image.

Therefore, both camera coordinates \( w, w^n \) can be obtained by \( \text{(10)-(11)} \).

\[
w = dK^{-1}p \quad \text{(12)}
\]

where \( K \) is the camera intrinsic parameter matrix. Then we can get resize cropping matrix \( R_C \) as \( \text{(13)} \), which represents the linear transformation of the world coordinates by resize cropping, i.e. \( w^n = R_C w \).

\[
R_C = \begin{bmatrix}
1 & -\frac{g_x}{2f} + \frac{c_y W}{2f} (1 - f) \\
1 & -\frac{g_y}{2f} + \frac{c_x H}{2f} (1 - f)
\end{bmatrix} \quad \text{(13)}
\]

Where \( (c_x, c_y) \) is the pixel coordinate of the camera principal point. We assume that the principal point is at the center of the image, then the \( R_C \) matrix can be simplified to \( \text{(14)} \).

\[
R_C = \begin{bmatrix}
1 & -\frac{g_x}{2f} \\
1 & -\frac{g_y}{2f}
\end{bmatrix} \quad \text{(14)}
\]

Thus we can use the \( R_C \) matrix to represent the depth and translation changes of objects in the image under the assumption that the camera system is constant.
In multi-view vision, the pixel coordinates \((u, v)\) of source and target views are related as in (15).

\[
D_t \begin{bmatrix} K^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_t \\ v_t \end{bmatrix} = \begin{bmatrix} R & t \end{bmatrix} D_s \begin{bmatrix} K^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_s \\ v_s \end{bmatrix}
\]

where \(R\) is the rotation and \(t\) is the translation from the source view to the target view. After resizing cropping, The rotation \(R^n\) and translation \(t^n\) need to be changed as (16) to satisfy \((15)\). Since \(R^n\) is not orthogonal, it is not a canonical rotation matrix.

\[
R^n = R_C R C^{-1}, t^n = R_C t
\]

(10) and (16) show the changes in the depth and pose estimated by the network compared to the original image when using resize cropping as data augmentation.

The equation of planes is also modified as (17) by resize cropping.

\[
\frac{n^t R C^{-1}}{||n^t R C^{-1}||} w^n - \frac{s}{||n^t R C^{-1}||} = 0
\]

where \(s\) is \(h\) in the ground planes or \(d\) in the vertical planes.

Acco\(rding to (16)(17)\), homography matrix after resize cropping (18) is also established with new camera pose and plane parameters.

\[
H^n = KR_C(R + \frac{1}{s}(t n^T)) R C^{-1} K^{-1}
\]

\[
= K(R_C R C^{-1} + \frac{||n^t R C^{-1}||_2}{s} R C t n^t R C^{-1} ||n^t R C^{-1}||_2) K^{-1}
\]

Essentially, the network predicts depth of vertical objects directly by predicting disparity, and predicts depth of ground indirectly by predicting camera height. (17) shows that resize cropping does not change the normal direction of vertical planes but only the depth, but changes the normal direction of the ground. It shows that the network needs to accurately predict the inclination of the ground caused by resize cropping before accurately predicting the camera height. To help the network get this information, we following [13] use neural positional encoding (NPE) to encode \(g\), which is the only variable in \(R_C\). Since our encoder is designed to extract image features and often uses weights pre-trained on image classification tasks, we only pass the NPE results into each layer of the decoder, as shown in Fig. 2. This step greatly improves the network’s prediction of the ground plane. The results are shown in Fig. 4.

Similarly, since \(R^n\) is not orthogonal and we use the pose network similar to [8], which cannot output non-orthogonal rotation matrices, we let the network output \(R\) and explicitly convert it to \(R^n\) according to (16). However, since we only use resized cropped images as inputs, the network should implicitly convert \(R^n\) to \(R\) first, indicating that it needs the information of \(R_C\) as input. We following [13] use neural positional encoding (NPE) to encode \(g\) as we do in the depth network. We simply pass the NPE results into the third layer from the end of the pose network, which resolves the ambiguity on rotation \(R\) in the network output, as shown in TABLE V.

C. Loss Functions

The previous works use weights \(p\) to linearly sum all warped planes to synthesise the target view image and compute the \(L_1\) loss with the ground truth image [12], [13]. The final predicted depth \(\hat{D}\) is obtained by weighted summation with the same weight \(p\). However, due to the ambiguity of weighted summation over colors, different weights for different colors can all get the same color as ground truth, resulting in a non-trivial solution.

Inspired by [25], we use a mixture Laplace distribution to fit the depth distribution. We treat the depth of each plane as the mean of single Laplace distribution as (19),

\[
p(D) = \sum_{i=0}^{N-1} \pi_i e^{-\frac{|D - d_i|}{2\sigma_i}}
\]

where \(\pi_i, \sigma_i\) are the weight and variance of the \(i\)-th peak, respectively. We consider training our model by minimizing the negative log-likelihood of depth distribution. Since \(D\) denotes the ground truth depth, which is not available during training. We simply use \(L_1\) photometric error instead of

---

Fig. 4. Predicted Ground. Our method can automatically segment the nearby ground to provide help for other tasks such as autonomous driving. NPE improves the area and confidence of the detected ground.
The final mixture Laplace loss is defined as (20),

\[ L_{ph} = - \log \frac{\pi_i \exp(-\frac{1}{2\sigma_i^2} ||\hat{y} - \hat{\pi}_i||^2)}{\sum_{j=0}^{N-1} \pi_j \exp(-\frac{1}{2\sigma_i^2} ||\hat{y} - \hat{\pi}_j||^2)} \]  

where \( \hat{\pi}_i \) are the warped color, weight and variance of the \( i \)-th plane respectively. Note that \( \hat{\pi}_i \) is obtained by the softmax results of the warped logits. The probability \( p_i \) of each plane is obtained by (21).

\[ p_i = \frac{p(D_i)}{\sum_{j=0}^{N-1} p(D_j)} \]  

Perceptual loss (26) is shown to be effective in novel view synthesis and self-supervised depth estimation [12], [19], which constrains the Euclidean distance of the images in the feature space as (22).

\[ L_{pc} = ||\phi(I_1) - \phi(I_t)||_2^2 \]  

where \( I = \sum_{i=0}^{N-1} p_i \hat{c}_i \) is the synthesis result of novel view and \( \phi \) is the first \( I \) maxpool layers of a VGG19 [27] pre-trained on the ImageNet [28]. We also use this loss to supervise our novel view synthesis indirectly.

Smooth loss (7) is also effective in many depth estimation works [8]-[10], [12], [13], [19]. We follow [12], [13] to use the smooth loss with a regular term \( \gamma = 2 \) added as (23).

\[ L_s = ||\partial_x \hat{d}_s||_{\gamma} + ||\partial_y \hat{d}_s||_{\gamma} \]  

Therefore, our final loss is (24),

\[ L = L_{ph} + \lambda_1 L_{pc} + \lambda_2 L_s \]  

which is averaged over pixel, view and batch. \( \lambda_1 \) and \( \lambda_2 \) are the weight hyper-parameters.

**D. Training Strategy**

Previous works optimize network outputs or address the occlusion problem with two-stage training and self-distillation loss [11]-[13]. We find that improving the training strategy can help MDE exploit depth cues better. Also, we combine post-processing [7] and mirror loss [12] to provide better supervision for the network self-distillation while solving the occlusion problem. Post-processing is a classic method to improve MDE results, which consumes twice the inference time and gets better depth from the original and the flipped images.

As aforementioned, the relative size and the relationship between objects are important cues for depth. In outdoor datasets, since the position of the ground is almost fixed, the relative relationship between the object and the ground, which could be seen as the vertical image position of the object, is important information for predicting depth. However, since we use resize cropping to break the relationship between depth and vertical image position of objects and limit the network to focus on relative size, we only exploit one cue as previous works.

After the first stage of training, our network will learn the relationship between the relative size and depth of objects. Since MDE will tend to predict depth from object vertical image position when the network is not limited by data augmentation [14]. We propose a strategy to let the network learn the relative size first and then the vertical image position. After completing the aforementioned training, we cancel resizing cropping and use the full resolution to fine-tune for a small number of epochs. The fine-tuning stage preserves the relative size cue which the network has learned, and introduces the vertical image position cue. TABLE II shows that this strategy greatly improves the performance of the network.

In addition, we consider the feasibility of introducing post-processing into self-distillation loss. The premise of using mirror loss is that the depth map will only produce artifacts on one side of all foregrounds. The post-processing flips the image, causing artifacts to appear on both sides of all foregrounds. Suppose we always produce artifacts on the left side of foregrounds in the left view and the right side in the double flipped left view, which is common in stereo training. We refer to mirror loss to obtain occlusion mask on left side of foregrounds in the left view \( M^l_1 \) and right side in the right view \( M^r_1 \). If we treat the left view as the right view, we can get the right side occlusion mask of the left view. However, since we do not have a more left view, we simplify the mirror loss so that the occlusion mask can be obtained only from the logits of the single source image as (25),

\[ M^l_i = \min \left( \sum_{i=0}^{N-1} W_{r \rightarrow l}(p^r_{l_i}, d_i), 1 \right) \]  

\[ p^r_i = S(\mathcal{W}_{r \rightarrow l}(l_i, d_i))_{i=0}^{N-1} \]  

where \( \mathcal{W}(\cdot, d) \) is a function of horizontal warp using disparity \( d \), \( S \) is softmax and \( l_i \) is the logit of the \( i \)-th plane. The right occlusion mask for the source view can be obtained by
TABLE I
Comparison of performance on KITTI Eigen test set [15]. The best is in bold and the second best is underlined in each metric.

| Methods       | PP  | Network          | Resolution | Train | Abs Rel | Sq Rel | RMSE | RMSE log | A1 | A2 | A3 |
|---------------|-----|------------------|------------|-------|---------|--------|------|----------|----|----|----|
| Monodepth2 [8] | ✓   | ResNet-50        | 1024x320   | S     | 0.097   | 0.793  | 4.533| 1.81     | 0.896 | 0.961 | 0.983 |
| DepthRef [9]  | ✓   | ResNet-50        | 1024x320   | S     | 0.096   | 0.710  | 4.393| 1.85     | 0.890 | 0.962 | 0.981 |
| PCFDNet [6]   | ✓   | ResNet-50        | 1024x320   | S     | 0.091   | 0.646  | 4.207| 1.76     | 0.901 | 0.966 | 0.983 |
| OFDNet [6]    | ✓   | ResNet-50-A      | 1280x384   | S     | 0.091   | 0.576  | 4.036| 1.74     | 0.901 | 0.967 | 0.984 |
| FALNet [10]   | ✓   | FALNet           | 1280x384   | S     | 0.097   | 0.590  | 3.991| 1.77     | 0.893 | 0.966 | 0.984 |
| ResNet-50     | ✓   | FALNet           | 1280x384   | S     | 0.093   | 0.564  | 3.973| 1.74     | 0.897 | 0.967 | 0.985 |
| PLADENet [15] | ✓   | PLADENet         | 1280x384   | S     | 0.092   | 0.626  | 4.046| 1.75     | 0.896 | 0.965 | 0.984 |
| PLADENet [15] | ✓   | PLADENet         | 1280x384   | S     | 0.089   | 0.590  | 4.008| 1.72     | 0.900 | 0.967 | 0.985 |
| Ours stage1   | ✓   | ResNet-50-A      | 1280x384   | S     | 0.090   | 0.604  | 4.151| 1.79     | 0.898 | 0.964 | 0.983 |
| Ours stage1   | ✓   | ResNet-50-A      | 1280x384   | S     | 0.087   | 0.577  | 4.063| 1.75     | 0.903 | 0.965 | 0.983 |
| Ours stage1   | ✓   | ResNet-50-A      | 1280x384   | S     | 0.085   | 0.562  | 4.053| 1.72     | 0.908 | 0.967 | 0.983 |
| Ours stage1   | ✓   | ResNet-50-A      | 1280x384   | S     | 0.085   | 0.553  | 4.024| 1.72     | 0.908 | 0.967 | 0.984 |

Improved Eigen Test Set [29]

| Methods       | PP  | Network          | Resolution | Train | Abs Rel | Sq Rel | RMSE | RMSE log | A1 | A2 | A3 |
|---------------|-----|------------------|------------|-------|---------|--------|------|----------|----|----|----|
| Monodepth2 [8] | ✓   | ResNet-18        | 1024x320   | S     | 0.104   | 0.775  | 4.562| 1.91     | 0.878 | 0.959 | 0.981 |
| DepthRef [9]  | ✓   | ResNet-18        | 1024x320   | S     | 0.098   | 0.702  | 4.398| 1.83     | 0.887 | 0.963 | 0.983 |
| FeatureNet [6] | ✓   | ResNet-50        | 1024x320   | S     | 0.099   | 0.697  | 4.427| 1.84     | 0.889 | 0.963 | 0.982 |
| Ours stage1   | ✓   | ResNet-50-A      | 1280x384   | S     | 0.092   | 0.601  | 4.188| 1.84     | 0.893 | 0.961 | 0.981 |
| Ours stage1   | ✓   | ResNet-50-A      | 1280x384   | S     | 0.090   | 0.584  | 4.130| 1.82     | 0.896 | 0.962 | 0.981 |

We also use occlusion-avoided photometric and perceptual losses in stage3 training like [12], [13]. We get the occlusion mask of the target view by

\[ M_t = \min\left( \sum_{i=0}^{N-1} W_{s \rightarrow t}(p_i, d_i), 1 \right) \] (29)

Occlusion-avoided photometric and perceptual losses is given by

\[ L_{ph} = M_t \odot \left( - \log \frac{1}{\pi_{t,t_i}^2} \frac{\|\phi_h(I_t) - \phi_h(M_t \odot I_t + (1 - M_t) \odot I_t)\|_2^2}{2\hat{\sigma}_{t,t_i}^2} \right) \] (30)

IV. Experiments

A. Implementation Details

We implemented our network using PyTorch [30] and trained it using Adam [31] with \( \beta_1 = 0.5, \beta_2 = 0.999 \). We set the batch size to 8, but 4 of them are obtained by flipping the other 4. Our default data augmentations are resize using a random scaling factor of 0.75 to 1.5, random cropping using size 640 x 192, random gamma, brightness and color noise. We trained 50 epochs in the first stage with an initialization learning rate of \( 1 \times 10^{-4} \), which is halved at 30 and 40 epochs, respectively. We trained 3 epochs with the initialization learning rate of \( 2.5 \times 10^{-5} \) in the second training stage, in which we disabled resize cropping and used full resolution (1280 x 384). We following the stage 2 in [12] train 10 epochs for our stage 3, replacing mirror loss with our self-distillation loss, and reducing the batch size to 4 and initialization learning rate to \( 2 \times 10^{-5} \), which is halved at 5 epoch. We set the minimum and maximum disparity to \( d_{min} = 2, d_{max} = 300 \) and the minimum and maximum camera heights to \( h_{min} = 1, h_{max} = 2 \). The hyper-parameters of our loss function are set to \( \lambda_1 = 0.1, \lambda_2 = 0.04 \).
Fig. 6. Qualitative results on the KITTI dataset. From left to right are the input and the visualization results of FALNet [12], EPCDepth [11], ours, respectively. Our network predicts smooth depth for the ground while preserving sharp edges of objects. It also has advantages for thin structures and occluded areas.

B. Datasets

We mainly conducted our experiments on the widely used KITTI [34] dataset, which captured sequential stereo images and sparse point clouds using stereo cameras and LiDAR mounted on a moving car. We followed the splits used in [8]. During stereo training, we adopt the Eigen training split [15], which contains 22,600 training and 888 validation stereo pairs. As for monocular training, we use the Zhou split [35], which removes static frames and contains 19,905 training and 2,212 validation stereo pairs. We evaluate our models on the 697 Eigen raw test images [15] and 652 Eigen improved test images [29] using the metrics proposed in [34], where Eigen improved test set [29] is obtained by warping the ground truth depth of 11 adjacent frames into one frame and filtering by SGM.

During both Eigen evaluations, all ground truth and predictions are clipped to within 80m and cropped by Eigen crop, and evaluated using the following metrics.

1) Abs Rel = \frac{1}{n} \sum_{i=1}^{n} \frac{|D_i - \hat{D}_i|}{D_i} \\
2) Sq Rel = \frac{1}{n} \sum_{i=1}^{n} \frac{|D_i - \hat{D}_i|^2}{D_i^2} \\
3) RMSE = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{|D_i - \hat{D}_i|^2}{D_i^2}\right)^{1/2} \\
4) RMSE \ log = \frac{1}{n} \sum_{i=1}^{n} \left(\left|\log(D_i) - \log(\hat{D}_i)\right|^2\right)^{1/2} \\
5) A_t = \frac{1}{n} \{\hat{D}_i|i \leq n, \max(D_i, \hat{D}_i) < 1.25^n\}

C. Benchmark Performance

We compare MDE performance with many previous state-of-the-art works on the raw and improved KITTI Eigen test set [15] and show results in TABLE I where S stands for stereo training using left and right views and a fixed baseline and MS stands for both stereo and monocular training adding front and rear frames of the left view without camera poses. ResNet-50-A represents U-Net [32] with DenseASPP module [33] using ResNet-50 [17] as the backbone. PP with checkmark ✓ refers to post-processing [35].

Quantitative results show that in stereo training, although we only use object relative size cue and does not use self-
TABLE II
COMPARISON OF DIFFERENT PLANE COMBINATIONS, WHERE LR MEANS LOW RESOLUTION 640 × 192, HR MEANS HIGH RESOLUTION 1280 × 384 AND RC WITH CHECKMARK MEANS USE RESIZE CROPPING.

| Stage1 Resolution | Stage1 RC | Stage2 Resolution | Stage2 RC | LR performance | HR performance |
|-------------------|-----------|-------------------|-----------|----------------|----------------|
|                   |           |                   |           | Abs Rel | Sq Rel | Rmse | A1 | Abs Rel | Sq Rel | Rmse | A1 |
| HR                | HR        |                   |           | 0.103   | 0.748 | 4.631 | 0.835 | 0.461   | 5.342 | 8.675 | 0.414 |
| LR                | LR        |                   |           | 0.100   | 0.708 | 4.588 | 0.882 | 0.090   | 0.616 | 4.188 | 0.898 |
| LR                | HR        |                   |           | 0.102   | 0.729 | 4.777 | 0.874 | 0.087   | 0.588 | 4.125 | 0.905 |

distortion loss to solve the occlusion problem in the first stage, benefiting from our plane prediction method and loss function, the performance is already better than previous methods with self-distillation loss or occlusion mask. After training in stage 2 and stage 3, the performance of our method is further improved and is the best among all methods. Thanks to the self-distillation loss combined with post-processing, our final performance without post-processing is very close to which with post-processing.

The results in the TABLE II also show that the performance of MS training is a little worse than that of S. It shows that when the stereo training is better, the moving objects caused by the timing inconsistency and poses predicted by networks affect the performance.

Fig. 6 shows the qualitative results between previous state-of-the-art methods and ours. The results show that our method can predict continuous depth for the ground while preserving sharp object edges. Our depth maps fit better with object edges, have fewer occlusion artifacts, and are less likely to ignore thin structures.

D. Ablation Studies

TABLE III
COMPARISON OF DIFFERENT PHOTOMETRIC LOSS,

| #VP | #GP | NPE | Abs Rel | Sq Rel | Rmse | A1 |
|-----|-----|-----|---------|--------|------|----|
| 49  |     |     | 0.092   | 0.601  | 4.143 | 0.895 |
| 49  |     | ✓   | 0.090   | 0.590  | 4.147 | 0.899 |
| 65  |     |     | 0.090   | 0.615  | 4.166 | 0.900 |
| 49  | 14  |     | 0.094   | 0.593  | 4.141 | 0.898 |
| 49  | 14  | ✓   | 0.090   | 0.604  | 4.151 | 0.898 |

1) Plane Prediction: We have tried various combinations of plane predictions and shows the results in TABLE III. As mentioned in [13], NPE can improve the performance of networks using only vertical planes by solving the problem of camera distortion. As we continue to add 14 vertical planes, the performance no longer improves. When we add 14 ground planes without NPE, the performance decreases significantly due to the ambiguity of the ground inclination. However, this decrease can also be significantly improved with the help of NPE. The visualization in Fig. 7 also shows that NPE can help the prediction of the ground, increasing the area and weight of the detected ground. These results suggest that in addition to solving the problem of camera distortion, taking the resize cropping parameter as the network input is also important for solving the ambiguity of the ground inclination.

The comparison results show that although we have no advantage in quantitative metrics, our method can segment the ground without supervision and predict continuous depth for it, which is important in some tasks, especially autonomous driving.

TABLE IV
COMPARISON OF DIFFERENT PHOTOMETRIC LOSS, WHERE MLL DENOTES OUR MIXTURE LAPLACE LOSS. MMP STANDS FOR MEAN MAXIMUM PROBABILITY.

| Loss   | Abs Rel | Sq Rel | Rmse | A1 | MMP |
|--------|---------|--------|------|----|-----|
| L1     | 0.095   | 0.606  | 4.293 | 0.893 | 0.543 |
| MLL    | 0.090   | 0.604  | 4.151 | 0.898 | 0.611 |

2) Mixture Laplace Loss: The quantitative results in TABLE IV show that the performance of our mixture Laplace loss is better than the baseline L1 loss, indicating that it is better to calculate the error of each warped plane separately than to calculate the error of a mixed image. Since the maximum probability reflects the degree of confidence of the network on the target value of depth classification, if the maximum probability is larger, it means that the network is more predicting or optimized with a certain objective. Otherwise, it means that the network uses an ambiguous combination of different values to get a certain objective. Therefore we propose to use the Mean Maximum Probability (MMP) to verify the effect of our loss. The results show that the MMP of our MML is higher than that of the L1 loss, indicating that MML is more inclined to use a single certain value as the optimization objective and the prediction result.

3) Training Strategy: We compare different training strategies in TABLE II and Fig. 7. To ensure that performance changes are not the effect of longer training times, we retain the stage2 training for the first three experiments in TABLE II with the same settings as the stage1. The results show that the networks without resize cropping only works at the training resolution, and only networks trained with resize cropping can perform well in both low and high resolutions, which indicates that resize cropping forces the network to learn the relationship between the relative size of objects and disparity. Compared with training on HR all the time, our strategy of training on LR and finetuning on HR saves a lot of training time and performs better with two cues. Compared with training on LR with RC all the time, our method learn the relationship between disparity and object vertical image position at HR in the second stage, which improves the performance at HR while...
We propose a novel plane-based self-supervised depth estimation network. It can automatically segment and predict continuous depth for the ground of outdoor datasets while preserving sharp boundaries for other objects. We discuss improvements suitable for our PlaneDepth from a complete pipeline of data augmentation, plane prediction, loss function and training strategy. We are the first to propose a data augmentation that allows the network to learn the relative size cue of objects during monocular training. We fit this data augmentation to our predefined plane. We then improved the training strategy so the network could learn both cues. For our depth classification method, we propose a loss function that fits the depth distribution to deal with the non-trivial problem. These improvements tightly fit our PlaneDepth and can also inspire other works independently.

VI. Acknowledgements

This work was supported by National Key R&D Program of China (2018AAA0100704), NSFC #61932020, #62172279, Science and Technology Commission of Shanghai Municipality (Grant No. 20ZR1436000), Program of Shanghai Academic Research Leader, and “Shuguang Program” supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission.

References

[1] S. Gur and L. Wolf, “Single image depth estimation trained via depth from defocus cues,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7683–7692.
[2] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, “Single view stereo matching,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 155–163.
[3] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2002–2011.
[4] Z. Li, X. Wang, X. Liu, and J. Jiang, “Binsformer: Revisiting adaptive bins for monocular depth estimation,” arXiv preprint arXiv:2204.09087, 2022.
[5] S. F. Bhat, I. Alhashim, and P. Wonka, “Adabins: Depth estimation using adaptive bins,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4009–4018.
[6] Z. Zhou and Q. Dong, “Learning occlusion-aware coarse-to-fine depth map for self-supervised monocular depth estimation,” arXiv preprint arXiv:2203.10925, 2022.
[7] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 270–279.
[8] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “ Digging into self-supervised monocular depth estimation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3828–3838.
[9] J. Watson, M. Firman, G. J. Brostow, and D. Tumukhambetov, “Self-supervised monocular depth hints,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2162–2171.
[10] C. Shu, K. Yu, Z. Duan, and K. Yang, “Feature-metric loss for self-supervised learning of depth and egomotion,” in European Conference on Computer Vision, Springer, 2020, pp. 572–588.
[11] R. Peng, R. Wang, Y. Lai, L. Tang, and Y. Cai, “Excavating the potential capacity of self-supervised monocular depth estimation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15.560–15.569.
[12] J. L. Gonzalez-Bello and M. Kim, “Forget about the lidar: Self-supervised depth estimators with med probability volumes,” Advances in Neural Information Processing Systems, vol. 33, pp. 12.626–12.637, 2020.
[13] J. L. Gonzalez and M. Kim, “Plade-net: Towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6851–6860.
[14] T. v. Dijk and G. d. Croon, “How do neural networks see depth in single images?” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2183–2191.
[15] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” Advances in neural information processing systems, vol. 27, 2014.
[16] J. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in 2016 Fourth international conference on 3D vision (3DV). IEEE, 2016, pp. 239–248.
[17] 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.
[18] R. Tucker and N. Snavely, “Single-view view synthesis with multiview images,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 551–560.
[19] J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G. H. Lee, “Mine: Towards continuous depth mpi with nerf for novel view synthesis,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12,578–12,588.

[20] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4104–4113.

[21] A. Pilzer, S. Lathuiliere, N. Sebe, and E. Ricci, “Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9768–9777.

[22] Z. Yu, L. Jin, and S. Gao, “Pnet: Patch-match and plane-regularization for unsupervised indoor depth estimation,” in European Conference on Computer Vision. Springer, 2020, pp. 206–222.

[23] S. Tulsiani, R. Tucker, and N. Snavely, “Layer-structured 3d scene inference via view synthesis,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 302–317.

[24] T. Khakhulin, D. Korzhenkov, P. Solovev, G. Sterkin, A.-T. Ardelean, and V. Lempitsky, “Stereo magnification with multi-layer images,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8687–8696.

[25] F. Tosi, Y. Liao, C. Schmitt, and A. Geiger, “Smd-nets: Stereo mixture density networks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8942–8952.

[26] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in European conference on computer vision. Springer, 2016, pp. 694–711.

[27] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[29] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

[30] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, “Denseaspp for semantic segmentation in street scenes,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3684–3692.

[31] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1851–1858.