Journal of Soft Computing and Artificial Intelligence, 6(2) (2025) 20-29 https://doi.org/10.55195/jscai.1748633

JSCAI

Journal of Soft Computing and Artificial Intelligence

ISSN (Online) 2717-8226

https://dergipark.org.tr/en/pub/jscai

Research Article

DepthP+P: Metric Accurate Monocular Depth Estimation using Planar and Parallax

Sadra Safadoust[1] **and Fatma Güney**[1*]

[1]*Department of Computer Engineering and KUIS AI Center, Koç University, Istanbul, Turkey*

[*]*Corresponding author*

Article Info

Keywords:
1. Monocular depth estimation
2. Metric depth estimation
3. Plane and parallax

Received: *23.06.2025*
Accepted: *11.11.2025*
Available online: *15.12.2025*

Abstract

Current self-supervised monocular depth estimation methods are mostly based on estimating a rigid-body motion representing camera motion. These methods suffer from the well-known scale ambiguity problem in their predictions. We propose DepthP+P, a method that learns to estimate outputs in metric scale by following the traditional planar parallax paradigm. We first align the two frames using a common ground plane which removes the effect of the rotation component in the camera motion. With two neural networks, we predict the depth and the camera translation, which is easier to predict alone compared to predicting it together with rotation. By assuming a known camera height, we can then calculate the induced 2D image motion of a 3D point and use it for reconstructing the target image in a self-supervised monocular approach. We perform experiments on the KITTI driving dataset and show that the planar parallax approach, which only needs to predict camera translation, can be a metrically accurate alternative to the current methods that rely on estimating 6DoF camera motion. Our method predicts monocular depth directly in metric scale, achieving an Abs Rel error of 0.134 on the KITTI dataset with the improved ground-truth annotations, without requiring any post-hoc scaling. Using additional stereo supervision, the error is further reduced to 0.084, demonstrating the effectiveness of the planar parallax paradigm for metrically accurate depth estimation.

1. Introduction

Understanding the 3D structure of a scene is fairly easy for human beings. We can easily reason about our surroundings and decompose them into different objects. Having this ability is crucial for autonomous vehicles to be able to drive in different environments. Training deep networks for estimating depth has proven successful in

Email address: fguney@ku.edu.tr
Cite as: S. Safadoust, F. Güney, *DepthP+P: Metric Accurate Monocular Depth Estimation using Planar and Parallax*, Journal of Soft Computing and Artificial Intelligence, 6(2) (2025), 20-29.

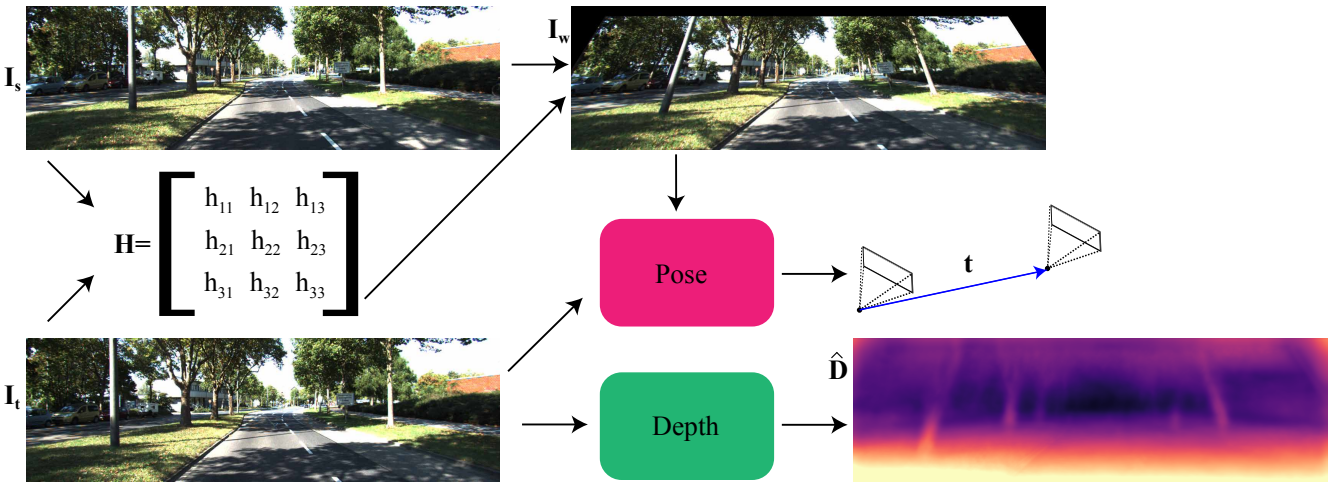

Figure 1.1: Overview of our Approach. Using the source image $\mathbf{I}_s$ and the target image $\mathbf{I}_t$, we first calculate the homography $\mathbf{H}$ that aligns the road plane across these two images. We then warp $\mathbf{I}_s$ according to $\mathbf{H}$ and obtain the aligned image $\mathbf{I}_w$. The aligned image $\mathbf{I}_w$ and the target image $\mathbf{I}_t$ are input to the pose network which estimates the camera translation $\mathbf{t}$ only. The depth network takes the $\mathbf{I}_t$ and produces a metric accurate depth map $\hat{\mathbf{D}}$.

computer vision research. However, many such methods are supervised and require ground truth depth which is costly to achieve. Another line of work uses a stereo setup that must be carefully calibrated. Both of these approaches cannot use the vast amount of unlabeled videos that are easily available for training. On the other hand, self-supervised monocular depth estimation methods that do not rely on stereo supervision do not suffer from these limitations and, in practice, have been closing the gap with their supervised or stereo counterparts.

Current self-supervised monocular depth estimation approaches all follow the same basic idea proposed in [1]. They use a pose network to estimate the ego-motion between a source frame and the target frame and a depth network to estimate the depth of the target image. These estimations can then be used to sample pixels from the source image to synthesize the target frame. The difference between the target frame and the synthesized can be used as the source of supervision for training the networks. In this paper, we propose another approach to synthesize the target image. Our approach, **DepthP+P**, illustrated in Figure 1.1, uses the traditional planar parallax formulation [2, 3], which decomposes the motion into a planar homography and a residual parallax. Consider a plane in the scene and its motion represented by a homography from the source to the target image. By first warping the source image according to this homography, the motion of the plane is canceled. Then the residual image motion depends on two factors: (1) the deviations of the scene structure from the plane, i.e., the depth of points and their perpendicular distance to the plane and (2) only the translational motion of the camera. Autonomous driving is a perfect use case for this approach because there is typically a planar surface in front of the vehicle, i.e., the road. However, it is important to note that the plane in the planar parallax formulation does not necessarily have to be a real plane and can also be a virtual plane, but choosing the road as the planar surface makes it easier to implement in practice. Moreover, our approach does not rely on the availability of a plane to predict depth during inference.

In this approach, we first align the road plane between the source and target images. This is achieved by calculating the homography between the road regions in two frames and then warping the source frame according to the homography to obtain the aligned image. By doing so, the road regions in the aligned image and target image match. The residual motion between the aligned image and the target image can be explained as follows: We first estimate the depth of each pixel with a monocular depth network and back-project them into 3D. Then using a known camera height, we can calculate the perpendicular distance of each point to the road. In addition, we estimate the translation between the camera origins. Note that this is different from the typical monocular depth approach [1, 4] which needs to estimate both the rotation and translation components. Finally, the target image can be synthesized from the aligned image using

the calculated residual parallax as shown in Figure 3.1. The planar parallax approach for self-supervised monocular estimations has a number of advantages over the previous paradigm. Firstly, it is much easier to optimize because it removes the ambiguities associated with predicting rotational camera motion [5]. Secondly, it can produce metric accurate outputs. Previous monocular depth methods can estimate depth and motion up to a scale. Typically, during inference, ground truth depth data is used to scale the predicted depth values such that the median of the predicted depth is equal to that of ground truth depth [1]. Our approach is able to predict metric accurate depth without needing ground truth depth data by only assuming a known camera height.

2. Related Work

2.1. Self-Supervised Monocular Depth

View Synthesis: Garg et al. [6] were the first to propose a method that uses view synthesis as an objective for depth estimation from single images. Instead of using ground truth depth for supervision, they use a CNN to estimate a depth map for the left image in the stereo setup. The estimated depth map and the known camera baseline are then used to inverse warp the right image to reconstruct the left. The photometric error between the reconstructed image and the original left image is used to train their network. Monodepth [7] uses Spatial Transformer Networks (STNs) [8] to synthesize the images in a fully-differentiable way. Using only the left image in the stereo setup as its input to the CNN network, it estimates the disparities for both left and right images. Subsequently, it reconstructs both images using the other image's estimated disparities and also enforces left-right consistency between disparities. SfmLearner [1] generalizes view synthesis to temporally consecutive images by using another network to predict the relative pose between them. Since they do not use any stereo supervision, their predicted depth is defined up to an unknown scale factor. Therefore, during evaluation, they scale their estimated depth map such that its median is equal to the median of the ground truth depth values. Zhan et al. [9] use stereo sequences to perform view synthesis using temporally consecutive pairs as well as the left-right pairs, enabling them to benefit from both monocular and stereo supervision. In addition to image reconstruction, they also use feature reconstruction as supervision. Similarly, by going beyond pixel-wise reconstruction error, Mahjourian et al. [10] propose to use a 3D point cloud alignment loss to enforce the estimated point clouds and the camera pose to be consistent temporally. Wang et al. [11] use direct visual odometry in a differentiable manner to solve for ego-motion using the estimated depth.

In addition to depth and camera pose, several methods estimate optical flow for residual motion. After predicting the camera motion, GeoNet [12] estimates the remaining object motion using optical flow. In order to prevent the errors of camera pose or depth predictions from propagating to flow estimations, DF-Net [13] enforces consistency between optical flow and the flow induced by the depth and pose predictions. GLNet [14] uses epipolar constraint for optical flow, along with other geometric constraints, further improving the performance. EPC++ [15] proposes a holistic 3D motion parser that uses predicted depth, pose, and optical flow to estimate segmentation masks for dynamic objects and their motion as well as background motion. Ranjan et al. [16] jointly train networks for depth, pose, optical flow, and motion segmentation so that they can use geometric constraints on the static regions and generic optical flow on moving objects. MonoDepthSeg [17] proposes to jointly estimate depth, independently moving regions, and their motion with an efficient architecture.

Some approaches keep the original framework with a depth and a pose network but improve the performance with better loss functions, improved network architectures, and innovative design choices. When estimating depth at multiple scales, Monodepth2 [4] proposes to first upsample the estimated low-scale depths to the input image size and then calculate the photometric loss at that scale. Monodepth2 also proposes to calculate the minimum of reprojection errors per pixel instead of averaging them when synthesizing the target image from multiple views to prevent blurry depth estimations. PackNet [18] changes the architecture of the depth network and uses 3D convolutions to learn to preserve spatial information using symmetrical 3D packing and unpacking blocks for predicting depth.

Scale Ambiguity: Self-supervised monocular depth estimation models suffer from the scale ambiguity problem,

and the depth and pose outputs of such models are in an unknown scale. The median scaling technique used by many previous methods does not actually solve this problem because it relies on ground truth depth data during inference which is not always easily available. Bian et al. [19] introduce a loss to minimize normalized differences of depth maps across the entire sequence. This makes the estimations globally scale-consistent. However, although this means that the predictions are at the same scale, that specific scale is still unknown, and the median scaling is still required during evaluation.

There are a number of monocular methods that can output depth estimations in absolute scale. Roussel et al. [20] use a network that was pre-trained with stereo pairs on a dataset and finetunes it on another dataset while maintaining the metric scale. Guizilini et al. [18] propose a version of their PackNet that uses ground truth camera velocity and the timestamps of images to enforce the estimations to be metrically accurate. Bartoccioni et al. [21] supervise their depth predictions with a sparse LiDAR. However, all of these approaches rely on ground truth data from extra sensors during training.

There are a few other methods that do not require additional supervision and only use the camera height to achieve depth estimations in metric units similar to the proposed method. DNet [22] estimates the ground plane during inference and, using the real height of the camera, recovers the scale of the predictions. However, it needs a ground plane to be visible during the test time. In other words, they do not train their depth outputs to be in absolute scale. Rather, they recover the scale of the estimations with another module during test time. Wagstaff and Kelly [23] train a network that learns the metric scale during training using camera height. They introduce a plane segmentation network and propose a three-staged training procedure for training the depth estimation model in metric scale. First, they train an unscaled depth network and then use it to train the plane segmentation network. Finally, they train a new metrically accurate depth network using the pre-trained plane segmentation network. Similar to [23], we also learn the metric scale during training, but we do not need a multi-stage process, nor do we rely on the existence of a ground plane during inference, differently from previous work [22].

2.2. Planar Parallax

The Planar Parallax paradigm, also called Plane + Parallax (P+P), has been used to understand the 3D structure of a scene from multiple images by decomposing the motion into a planar homography and a residual parallax. Sawhney [2] proposes a formulation for the residual parallax that uses depth and distance to the plane. Irani et al. [3] use this formulation to derive a rigidity constraint between pairs of points over multiple images. Irani et al. [24] derive trifocal constraints and use them to propose a simple method for new view synthesis. In a follow-up work [5], they extend the planar parallax method to more than two uncalibrated frames.

More recently, MR-Flow [25] uses P+P to refine the optical flow estimations with rigidity constraints. Chaney et al. [26] use P+P to estimate the height of points in the scene with event-based cameras. We propose a method to use the P+P formulation within the view synthesis framework for self-supervised monocular depth estimation.

3. Methodology

Despite the success of current self-supervised monocular depth estimation approaches, they suffer from scale ambiguity. i.e., the estimated depth values are in an unknown scale. Therefore, in order to evaluate and compare these methods, they are usually normalized using the median scaling approach [1]. Here, we propose an approach that predicts depth maps in metric scale without using any ground truth depth supervision.

3.1. DepthP+P

Our approach is based on the Planar Parallax decomposition which has been studied in detail before [2, 3]. We first introduce it here to establish our notation and then build our method to predict depth following that notation.

Notation: Let Π be a 3D plane and $\mathbf{H}$ be the homography aligning Π between the target image $\mathbf{I}_t$ and the source image $\mathbf{I}_s$. Let $\mathbf{p}$ and $\mathbf{p}'$ be the images of the 3D point $\mathbf{x} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]^T$ on the $\mathbf{I}_t$ and $\mathbf{I}_s$ respectively. As shown on the left in Figure 3.1, we can warp $\mathbf{p}'$ by the homography $\mathbf{H}$ and obtain the image point $\mathbf{p}_w$:

$$\mathbf{p}_w \sim \mathbf{H}\mathbf{p}' \tag{3.1}$$

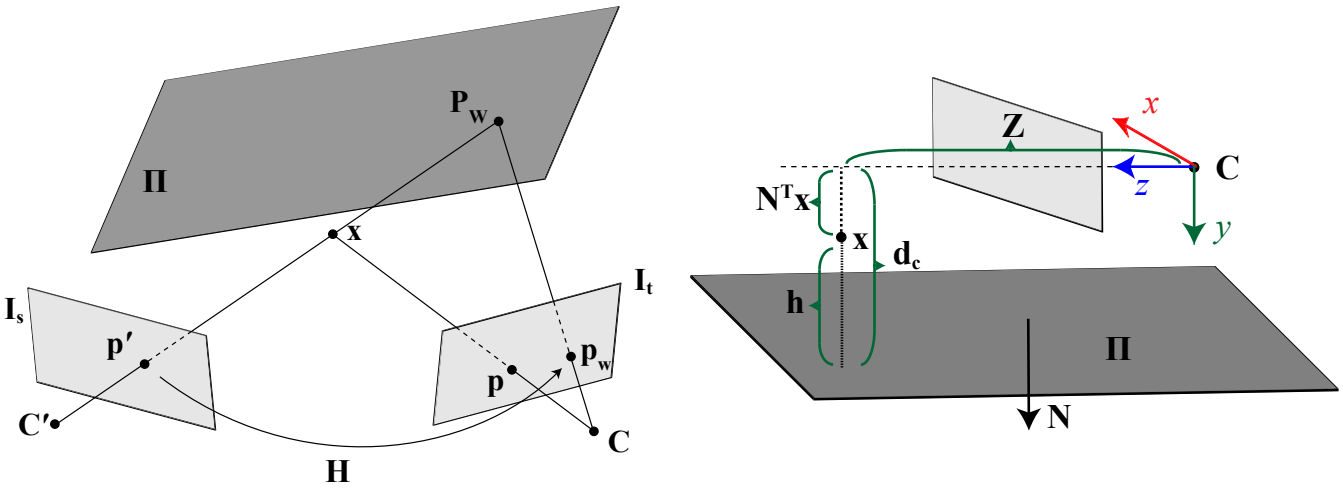

Figure 3.1: Visualization of the Planar Parallax. *Left:* The 3D point $\mathbf{x}$ is projected to points $\mathbf{p}$ and $\mathbf{p}'$ on the target image $\mathbf{I}_t$ and the source image $\mathbf{I}_s$ respectively. Using the homography $\mathbf{H}$ induced by the plane Π, the point $\mathbf{p}'$ will be transformed to point $\mathbf{p}_w$ on the target image. *Right:* Calculating $\mathbf{h}$, distance of $\mathbf{x}$ to the Π using the camera height $\mathbf{d}_c$ and the normal vector $\mathbf{N}$ of the plane. $\mathbf{C}$ and $\mathbf{C}'$ are the camera centers of $\mathbf{I}_t$ and $\mathbf{I}_s$ and $\mathbf{Z}$ is the depth of the point.

where we omit the conversion to the homogenous coordinates. Note that by warping the source image $\mathbf{I}_s$, we obtain the aligned image $\mathbf{I}_w$ such that the plane Π matches between them. The displacement between $\mathbf{p}_w$ and $\mathbf{p}$ can be computed as follows:

$$\mathbf{p}_w - \mathbf{p} = \frac{\gamma}{\mathbf{d}_c - \gamma \mathbf{t}_z}(\mathbf{t}_z \mathbf{p} - \mathbf{K}\mathbf{t}) \tag{3.2}$$

where $\mathbf{K}$ is the camera intrinsic, $\mathbf{t} = [\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z]^T$ is the translation vector between the $\mathbf{I}_t$ and $\mathbf{I}_s$, and $\mathbf{d}_c$ is the distance between the camera for the source view to the plane Π. The structure is represented by $\gamma = \frac{\mathbf{h}}{\mathbf{Z}}$ where $\mathbf{h}$ is the distance of $\mathbf{x}$ to Π. Note that when $\mathbf{x}$ lies on the plane Π, i.e., $\mathbf{h} = 0$, we will have $\mathbf{p}_w = \mathbf{p}$.

DepthP+P: Following the typical self-supervised monocular depth approach, our framework has two networks, one for estimating depth and another for estimating the translation between frames. Note that, unlike other methods, we do not need to estimate the rotation between the two views. Precisely, our pose network takes the target image $\mathbf{I}_t$ and aligned image $\mathbf{I}_w$ and outputs the translation vector $\mathbf{t}$. The depth network takes the target image $\mathbf{I}_t$ and outputs the depth map $\hat{\mathbf{D}}$ for $\mathbf{I}_t$. For every pixel $\mathbf{p} = [x, y]$, let $\hat{\mathbf{D}}(\mathbf{p})$ denote its estimated depth. We backproject $\mathbf{p}$ using the camera intrinsics and the estimated depth to obtain the corresponding 3D point $\hat{\mathbf{x}}$ in the camera coordinate system as follows:

$$\hat{\mathbf{x}} = \hat{\mathbf{D}}(\mathbf{p})\ \mathbf{K}^{-1}\ [x, y, 1]^T. \tag{3.3}$$

Therefore, as demonstrated on the right in Figure 3.1, we have the following:

$$\hat{\mathbf{h}} = \mathbf{d}_c - \mathbf{N}^T \hat{\mathbf{x}}, \qquad \hat{\gamma} = \frac{\hat{\mathbf{h}}}{\hat{\mathbf{D}}(\mathbf{p})} \tag{3.4}$$

where $\mathbf{N}$ is the normal vector of the plane Π, $\hat{\mathbf{h}}$ is the estimated distance of the point $\hat{\mathbf{x}}$ to the plane Π and $\hat{\gamma}$ is our estimate of the structure variable γ. As a result, we obtain all the parameters required to use Eq. (3.2) to reconstruct the target image $\mathbf{I}_t$ by warping the aligned image $\mathbf{I}_w$ resulting in $\hat{\mathbf{I}}_w$. In other words, for each pixel $\mathbf{p}$ on the $\mathbf{I}_t$, we calculate the $\mathbf{p}_w$ using (3.2) according to the depth and translation predicted by our two networks and then inverse warp $\mathbf{I}_w$ and obtain $\hat{\mathbf{I}}_w$ to reconstruct $\mathbf{I}_t$:

$$\mathbf{I}_t(\mathbf{p}) \approx \hat{\mathbf{I}}_w(\mathbf{p}) = \mathbf{I}_w(\mathbf{p}_w) \tag{3.5}$$

We minimize the difference between $\mathbf{I}_t$ and $\hat{\mathbf{I}}_w$ for supervision as explained in Section 3.2. In order to obtain the aligned images, we perform a pre-processing step on the dataset. We calculate a homography for every consecutive frame by using the road as the plane Π and warp the frames according to the calculated homographies. In other words, we calculate a homography $\mathbf{H}$ for every target image $\mathbf{I}_t$ and source image $\mathbf{I}_s$ pair, and then warp $\mathbf{I}_s$ according to $\mathbf{H}$ to obtain the warped source image $\mathbf{I}_w$. We explain the details of this pre-processing step in Section 4.1.

3.2. Self-Supervised Training Loss

In our approach, we define our photometric loss function as the linear combination of the L1 distance and the structural similarity (SSIM) [27] to minimize the difference between the target image $\mathbf{I}_t$ and the reconstructed image $\hat{\mathbf{I}}_w$. Our photometric loss is therefore defined as follows:

$$\mathscr{L}_{\text{photo}}(\mathbf{p}) = (1-\alpha)\,|\mathbf{I}_t(\mathbf{p}) - \hat{\mathbf{I}}_w(\mathbf{p})| + \frac{\alpha}{2}\left(1 - \text{SSIM}(\mathbf{I}_t, \hat{\mathbf{I}}_w)(\mathbf{p})\right) \quad (3.6)$$

where we set $\alpha = 0.85$. Note that for every target image we consider two aligned images. One from warping the previous frame, and one from warping the next frame. We use the per-pixel minimum reprojection error introduced in [4] and calculate the minimum of the $\mathscr{L}_{photo}$ for each pixel across the previous and next aligned images. We also define $\mathscr{L}_{smooth}$ as an edge-aware smoothness loss over the mean-normalized inverse depth estimates [11] to encourage the depth predictions to be locally smooth. Our total loss function is a combination of $\mathscr{L}_{smooth}$ and $\mathscr{L}_{photo}$ averaged over all N pixels:

$$\mathscr{L} = \frac{1}{N}\sum_{\mathbf{p}} \lambda\ \mathscr{L}_{smooth}(\mathbf{p}) + \min_{w}(\mathscr{L}_{photo}(\mathbf{p})) \quad (3.7)$$

where $\min_w$ calculates the minimum over the previous and next aligned frames and λ is a hyperparameter controlling the effect of the loss terms.

3.3. Network Architecture

Our depth network is based on the U-Net architecture [28]. We use a ResNet [29] pre-trained on the ImageNet [30] as the encoder for our depth network, and for the decoder we use the architecture similar to the one used by [4]. The difference is that we directly estimate depth by multiplying the output of the last sigmoid layer by 250, which is the maximum depth value that can be predicted, instead of estimating disparity as in [4]. The depth network takes as input a single target image $\mathbf{I}_t$ and outputs the per-pixel depth estimates. Note that the output of our depth decoder is in metric scale.

In DepthP+P, our second network takes $\mathbf{I}_w$ and $\mathbf{I}_t$ and outputs only the translation vector between the views. The network is similar to the pose network proposed in [4], except that the output is a 3-element vector representing the translation and is metric scale in our case.

4. Experiments

4.1. Dataset

KITTI: We use the Eigen split [31] of the KITTI dataset [32, 33] to train and evaluate our model. We use all of the images in the split for which we could accurately estimate the homography for aligning the road between consecutive images as explained in the next paragraph. This results in 45000 training and 1769 validation samples. We evaluate our model on the 697 test images in the split using the original ground truth provided by LiDAR. We also report results using the improved ground truth for 652 test images provided by Uhrig et al [34]. They use a stereo-reconstruction method to remove the outliers in LiDAR points and increase the ground truth density by accumulating laser scans which result in high-quality ground truth data. The camera height in this dataset is $\mathbf{d}_c = 1.65$ and we assume that the road is completely horizontal, i.e., $\mathbf{N} = [0,1,0]^T$.

Pre-processing the dataset for DepthP+P: In order to use our P+P approach, we need to calculate the homography between the consecutive frames and warp the source frame according to the estimated homographies. Since we work on the driving scenarios on KITTI, we choose the "road" as our plane Π which is visible in most of the frames. For calculating the homography, we need to find a set of (at least 4) corresponding pairs of road pixels between a source view $\mathbf{I}_s$ and the target view $\mathbf{I}_t$, i.e., two consecutive images. For this purpose, we use the optical flow between $\mathbf{I}_s$ and $\mathbf{I}_t$ using [35] to find the corresponding pixels. We then use [36] to select only the pixels that belong to the semantic class "road". Using the corresponding pairs of road pixels, we estimate the homography $\mathbf{H}$ using OpenCV's RANSAC-based robust method. We do this to find the homography $\mathbf{H}$ for all of the consecutive pairs of frames on KITTI. Note that for any consecutive pair of frames $\mathbf{I}_1, \mathbf{I}_2$ and the homography $\mathbf{H}$ between them, we use $\mathbf{H}$ to warp $\mathbf{I}_1$ towards $\mathbf{I}_2$ and also use $\mathbf{H}^{-1}$ to warp $\mathbf{I}_2$ towards $\mathbf{I}_1$.

4.2. Depth Estimation Results

In Table 4.1, we report the depth estimation results of our method on the KITTI Eigen split using both the

Table 4.1: Quantitative Results for Monocular Training on KITTI. This table compares our proposed approach, **DepthP+P**, to previous approaches on the KITTI dataset that were trained only with monocular supervision. The **scale** column specifies whether the method can estimate depth in metric scale. We provide results with the original and improved ground truth. We show the results for the input resolution 640×192. The best method in each column is shown in bold and the second best is underlined.

	Method	Scale	Lower Better				Higher Better		
			Abs Rel	Sq Rel	RMSE	RMSE_{log}	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Original Ground Truth	Zhou et al. [1]	✗	0.183	1.595	6.709	0.270	0.734	0.902	0.959
	Yang et al. [37]	✗	0.182	1.481	6.501	0.267	0.725	0.906	0.963
	Mahjourian et al. [10]	✗	0.163	1.240	6.220	0.250	0.762	0.916	0.968
	Yin et al. [12]	✗	0.149	1.060	5.567	2.226	0.796	0.935	0.975
	Wang et al. [11]	✗	0.151	1.257	5.583	0.228	0.810	0.936	0.974
	Zou et al. [13]	✗	0.150	1.124	5.507	0.223	0.806	0.933	0.973
	Yang et al. [38]	✗	0.162	1.352	6.276	0.252	-	-	-
	Ranjan et al. [16]	✗	0.148	1.149	5.464	0.226	0.815	0.935	0.973
	Luo et al. [15]	✗	0.141	1.029	5.350	0.216	0.816	0.941	0.976
	Chen et al. [14]	✗	0.135	1.070	5.230	0.210	0.841	0.948	0.980
	Godard et al. [4]	✗	**0.110**	0.831	4.642	**0.187**	**0.883**	**0.962**	**0.982**
	Guizilini et al. [18]	✗	0.111	**0.785**	**4.601**	0.189	0.878	0.960	**0.982**
	Safadoust et al. [17]	✗	**0.110**	0.792	4.700	0.189	0.881	0.960	**0.982**
	Xue et al. [22]	✓	0.118	0.925	4.918	0.199	0.862	0.953	0.979
	Wagstaff and Kelly [23]	✓	0.123	0.996	5.253	0.213	0.840	0.947	0.978
	DepthP+P (Ours)	✓	0.152	1.322	6.185	0.239	0.781	0.920	0.970
Improved GT [34]	Zhou et al. [1]	✗	0.176	1.532	6.129	0.244	0.758	0.921	0.971
	Mahjourian et al. [10]	✗	0.134	0.983	5.501	0.203	0.827	0.944	0.981
	Yin et al. [12]	✗	0.132	0.994	5.240	0.193	0.833	0.953	0.985
	Wang et al. [11]	✗	0.126	0.866	4.932	0.185	0.851	0.958	0.986
	Ranjan et al. [16]	✗	0.123	0.881	4.834	0.181	0.860	0.959	0.985
	Luo et al. [15]	✗	0.120	0.789	4.755	0.177	0.856	0.961	0.987
	Godard et al. [4]	✗	0.085	0.468	3.672	0.128	0.921	0.985	0.995
	Safadoust et al. [17]	✗	0.085	0.458	3.779	0.131	0.919	0.985	**0.996**
	Guizilini et al. [18]	✗	**0.078**	**0.420**	**3.485**	**0.121**	**0.931**	**0.986**	**0.996**
	DepthP+P (Ours)	✓	0.134	1.042	5.566	0.199	0.820	0.946	0.983

original and the improved ground truth. To the best of our knowledge, this is the first time that a deep learning model has been trained with view synthesis through the planar parallax paradigm (Eq. (3.2)). All of the previous methods are trained based on estimating the pose whereas our method introduces a novel approach. We can see that our method achieves significantly better results than the initial models by predicting the pose and depth. After the initial proposal of SfMLearner by Zhou et al. [1], several improvements have been proposed to improve its performance. Therefore, we believe that similar improvements can follow our model as future work to make it perform better than our initial proposal as well as the other state-of-the-art models that are trained to estimate the full pose.

Note that [22] is not trained to estimate metrically accurate depth. Instead, its depth network outputs depth in an unknown scale, and then during inference, it needs a ground plane to be visible on the image to recover the scale of the network. When the ground plane is not visible on the image, [22] fails completely as shown in Figure 4.1. As can be seen in this figure, this image from the KITTI dataset does not have a ground plane, and [22] cannot recover the scale and produces completely wrong estimates. While our method needs a ground plane during training, it does not rely on the availability of the ground plane during inference, therefore it can still perform well. For reference, the absolute relative (Abs Rel) error of [22] on Figure 4.1 is 1.178, while our model achieves a 0.252 error. [23] achieves better results by using a pre-trained plane segmentation network in addition to the depth network, while our approach can achieve

Table 4.2: Quantitative Results on KITTI with Additional Stereo Supervision. We compare DepthP+P to previous approaches that use additional stereo supervision on KITTI. Stereo supervision significantly improves the results of DepthP+P model. By using ResNet50, our model performs on par with Monodepth2 [4].

	Method	Lower Better				Higher Better		
		Abs Rel	Sq Rel	RMSE	$RMSE_{log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Original GT	Li et al. [39]	0.183	1.730	6.570	0.268	-	-	-
	Zhan et al. [9]	0.135	1.132	5.585	0.229	0.820	0.933	0.971
	Luo et al. [15]	0.128	0.935	5.011	0.209	0.831	0.945	**0.979**
	Godard et al. [4]	**0.106**	**0.818**	**4.750**	**0.196**	**0.874**	**0.957**	**0.979**
	DepthP+P (ResNet18)	0.110	0.907	4.888	0.199	0.867	0.954	**0.979**
	DepthP+P (ResNet50)	**0.106**	0.900	4.828	0.198	0.871	0.954	**0.979**
Improved GT	Zhan et al. [9]	0.130	1.520	5.184	0.205	0.859	0.955	0.981
	Luo et al. [15]	0.123	0.754	4.453	0.172	0.863	0.964	0.989
	Godard et al. [4]	**0.080**	**0.466**	**3.681**	**0.127**	**0.926**	**0.985**	**0.995**
	DepthP+P (ResNet18)	0.088	0.572	3.905	0.138	0.911	0.981	0.994
	DepthP+P (ResNet50)	0.084	0.543	3.784	0.134	0.916	0.982	**0.995**

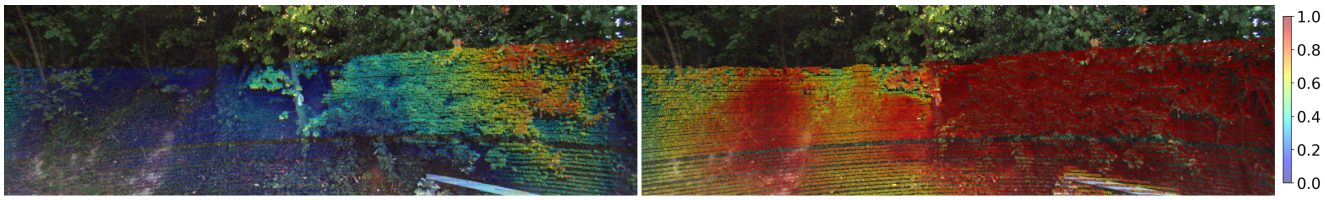

Figure 4.1: Qualitative Comparison. We compare the absolute relative error of our depth estimation method (**left**) with DNet [22] (**right**) on an image from the KITTI dataset without a ground plane. The colorbar on the right shows the values of the absolute relative error metric. We cap the max error at the value of 1.0 for visualization. We can see that [22] completely fails to estimate the metric depth due to the wrong scale recovery because there is no ground plane in the image, while our model does not have this issue and can still perform well. The absolute relative error for [22] is 1.178 while it is 0.252 for our method.

comparable results without a separate segmentation network.

DepthP+P can also be trained with additional stereo supervision. In the proposed approach, we obtain monocular supervision from the P+P paradigm. In addition, using the known camera baseline and the estimated depth, we can warp the other image in the stereo setup to the input image for additional supervision signal. In Table 4.2, we report the performances of the methods that also use stereo supervision for training. Using stereo supervision significantly improves the performance of our DepthP+P model, outperforming all methods except for Monodepth2 [4]. We show that by using a ResNet50 backbone instead of ResNet18, DepthP+P can obtain comparable results to Monodepth2 [4].

5. Conclusion and Future Work

In this paper, we presented a new approach to self-supervised monocular depth estimation following the traditional planar parallax paradigm. We showed that our approach is able to produce metrically accurate depth estimates by using a known camera height. Unlike previous methods that rely on estimating the full rigid-body motion of the camera, our method only needs to estimate the camera translation. We discussed the advantage of our method compared to the other scale-aware depth prediction methods. We see our approach as a first step to unlocking the potential of the plane and parallax for efficient and metric-accurate depth estimation. An exciting future direction can focus on detecting moving foreground objects by checking the violations in the plane and parallax constraints [3].

In addition, we showed that this approach performs even better than many other methods that cannot estimate depth in unknown scale. We believe that similar to the improvements to the original SfmLearner method, e.g., handling the motion of moving objects, our method can benefit from such additions to achieve even better results.

Article Information

Funding Statement: This project has received funding from KUIS AI Center and TÜBİTAK (118C256) grant.

Author Contributions: *Sadra Safadoust:* Conceptualization, Methodology, Data Curation, Software, Writing - Review & Editing
Fatma Güney: Conceptualization, Validation, Resources, Writing - Original Draft, Supervision, Project Administration, Funding Acquisition

Artificial Intelligence Statement: No AI tools were used in the writing or editing of this manuscript.

Conflict of Interest Disclosure: The authors declare that they have no conflict of interest.

Plagiarism Statement: This manuscript has been checked for plagiarism using appropriate similarity detection tools and complies with the ethical standards of JSCAI.

References

[1] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, pp. 1851–1858, 2017.

[2] Sawhney, "3d geometry from planar parallax," in *CVPR*, pp. 929–934, 1994.

[3] M. Irani and P. Anandan, "Parallax geometry of pairs of points for 3d scene analysis," in *ECCV* (B. Buxton and R. Cipolla, eds.), (Berlin, Heidelberg), pp. 17–30, Springer Berlin Heidelberg, 1996.

[4] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*, 2019.

[5] M. Irani, P. Anandan, and M. Cohen, "Direct recovery of planar-parallax from multiple frames," *PAMI*, vol. 24, no. 11, pp. 1528–1534, 2002.

[6] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *ECCV*, pp. 740–756, 2016.

[7] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, pp. 270–279, 2017.

[8] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *NeurIPS*, pp. 2017–2025, 2015.

[9] 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 *CVPR*, pp. 340–349, 2018.

[10] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *CVPR*, pp. 5667–5675, 2018.

[11] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *CVPR*, pp. 2022–2030, 2018.

[12] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *CVPR*, pp. 1983–1992, 2018.

[13] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *ECCV*, pp. 36–53, 2018.

[14] Y. Chen, C. Schmid, and C. Sminchisescu, "Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera," in *ICCV*, pp. 7063–7072, 2019.

[15] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *PAMI*, vol. 42, no. 10, pp. 2624–2641, 2019.

[16] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *CVPR*, pp. 12240–12249, 2019.

[17] S. Safadoust and F. Güney, "Self-supervised monocular scene decomposition and depth estimation," in *International Conference on 3D Vision (3DV)*, pp. 627–636, 2021.

[18] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *CVPR*, 2020.

[19] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *NeurIPS*, pp. 35–45, 2019.

[20] T. Roussel, L. V. Eycken, and T. Tuytelaars, "Monocular depth estimation in new environments with absolute scale," in *IROS*, pp. 1735–1741, 2019.

[21] F. Bartoccioni, E. Zablocki, P. Perez, M. Cord, and K. Alahari, "Lidartouch: Monocular metric depth estimation with a few-beam lidar," *arXiv.org*, vol. 2109.03569, 2021.

[22] F. Xue, G. Zhuo, Z. Huang, W. Fu, Z. Wu, and M. H. Ang, "Toward hierarchical self-supervised monocular absolute depth estimation for autonomous driving applications," in *IROS*, pp. 2330–2337, IEEE, 2020.

[23] B. Wagstaff and J. Kelly, "Self-supervised scale recovery for monocular depth and egomotion estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2620–2627, IEEE, 2021.

[24] M. Irani, P. Anandan, and D. Weinshall, "From reference frames to reference planes: Multi-view parallax geometry and applications," in *ECCV* (H. Burkhardt and B. Neumann, eds.), (Berlin, Heidelberg), pp. 829–845, Springer Berlin Heidelberg, 1998.

[25] J. Wulff, L. Sevilla-Lara, and M. J. Black, "Optical flow in mostly rigid scenes," in *CVPR*, pp. 4671–4680, 2017.

[26] K. Chaney, A. Z. Zhu, and K. Daniilidis, "Learning event-based height from plane and parallax," in *IROS*, pp. 3690–3696, 2019.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *TIP*, vol. 13, no. 4, pp. 600–612, 2004.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *MICCAI*, pp. 234–241, 2015.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770–778, 2016.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "ImageNet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[31] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NeurIPS*, pp. 2366–2374, 2014.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *IJRR*, 2013.

[33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[34] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *3DV*, 2017.

[35] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *ECCV*, pp. 402–419, Springer, 2020.

[36] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *CVPR*, 2019.

[37] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia, "Unsupervised learning of geometry from videos with edge-aware depth-normal consistency," in *AAAI*, 2018.

[38] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "LEGO: Learning edge with geometry all at once by watching videos," in *CVPR*, pp. 225–234, 2018.

[39] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *ICRA*, pp. 7286–7291, IEEE, 2018.