Monocular Depth Estimation for Tilted Images via Gravity Rectifier

Yuki Saito, Hideo Saito, Vincent Frémont

Monocular Depth Estimation for Tilted Images via Gravity Rectifier

Yuki Saito[1], Hideo Saito[1], and Vincent Frémont[2]
[1]*Faculty of Science and Technology, Keio University, Yokohama, Kanagawa, Japan*
[2]*CNRS, LS2N, Nantes Universite, Ecole Central de Nantes, UMR 6004, F-44000 Nantes, France*
{*yusa19971015, hs*}*@keio.jp, vincent.fremont@ec-nantes.fr*

Keywords: monocular depth estimation, tilted images, gravity prediction, convolutional neural network

Abstract: Monocular depth estimation is a challenging task in computer vision. Although many approaches using Convolutional neural networks (CNNs) have been proposed, most of them are trained on large-scale datasets mainly composed of gravity-aligned images. Therefore, conventional approaches fail to predict reliable depth for tilted images containing large pitch and roll camera rotations. To tackle this problem, we propose a novel refining method based on the distribution of gravity directions in the training sets. We designed a gravity rectifier that is learned to transform the gravity direction of a tilted image into a rectified one that matches the gravity-aligned training data distribution. For the evaluation, we employed public datasets and also created our own dataset composed of large pitch and roll camera movements. Our experiments showed that our approach successfully rectified the camera rotation and outperformed our baselines, which achieved 29% improvement in *abs_rel* over the vanilla model. Additionally, our method had competitive accuracy comparable to state-of-the-art monocular depth prediction approaches considering camera rotation.

1 INTRODUCTION

Monocular depth estimation, i.e., predicting a dense depth map from a single RGB image, is an essential task that is widely employed in many robotics and autonomous system tasks, such as ego-motion estimation (Tateno et al., 2017; Czarnowski et al., 2020), robot navigation systems (Yang et al., 2019; Marcu et al., 2018; Zhang et al., 2019), and augmented reality (Wang et al., 2018; Luo et al., 2020). Recently, depth prediction based on convolutional neural networks (CNNs) has demonstrated successful performance on many benchmark scores and predicted plausible depth appearance (Eigen et al., 2014; Laina et al., 2016; Fu et al., 2018; Godard et al., 2019).

Depth prediction approaches with CNN are generally trained with large-scale image datasets, which contain millions of RGB-D image pairs for various indoor and outdoor scenes (Dai et al., 2017; Silberman et al., 2012; Sturm et al., 2012; Geiger et al., 2013). However, these images are mainly captured under certain camera motions, which leads to biased camera pose distributions in the training set.

As a result, conventional depth prediction approaches fail to estimate reliable depth maps on images captured under uncommon camera poses, such as tilted inputs, which include large roll and pitch rotations (Saito et al., 2020; Zhao et al., 2021). For illustration, Figure 1 shows the distribution of camera rotation along roll directions in the ScanNet dataset (Dai et al., 2017). The distribution in the train-

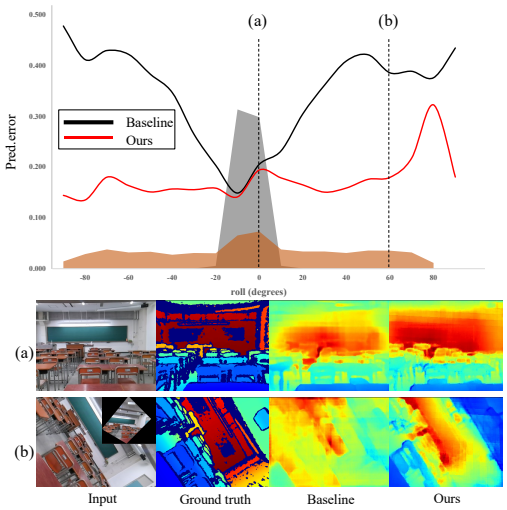

Figure 1: Distribution bias of camera rotation between the training set and test set. The horizontal axis shows the roll rotation angle of the camera (degree), and the vertical axis shows absolute relative error. The training set distribution is shown in gray, and the test set distribution is shown in orange.(a) shows the predicted depth in the upright scene, and (b) shows the predicted depth in tilted scenes.

ing set is gathered around 0°, and tilted scenes with large roll rotation are not equally represented in deep neural networks. Therefore, compared with gravity-aligned scenes (a), the conventional depth prediction approach (Baseline) was unable to predict a reliable depth map on tilted scenes (b), and its prediction error significantly increased. This performance degradation causes a crucial problem in applications for which body-/robot-mounted cameras are employed and tilted images are easily captured under unexpected control, e.g., mobile AR (Luo et al., 2020) and UAV (Marcu et al., 2018; Zhang et al., 2019).

To tackle this problem, several works have recently been proposed for monocular depth estimation by introducing camera pose as prior information. Saito et al. (Saito et al., 2020) and Sartipi et al. (Sartipi et al., 2020) proposed a method to rectify the in-plane rotation of images with Visual-SLAM (Mur-Artal and Tardós, 2017), which enabled more accurate depth prediction. Also, Zhao et al. (Zhao et al., 2021) proposed a method to incorporate 2D maps encoded from camera poses with RGB input as prior knowledge of depth prediction. However, these methods heavily relied on offline pose estimation systems or external sensors, which required high computational costs or additional equipment, like an inertial measurement unit (IMU).

In this paper, we addressed this challenging task for depth estimation with tilted inputs leveraging only RGB information. We hypothesized that gravity direction is an indicator of the global orientation of the scene, which is a strong prior to depth estimation with CNNs (Saito et al., 2020; Mi et al., 2022; Do et al., 2020). We considered that gravity alignment between the training set and the test set can minimize the domain gap between the test set composed of tilted images and the training set composed of upright images.

To this end, we proposed a gravity rectifier network to learn transformation that warps tilted input to an upright image so that its gravity direction can be matched to the dominant direction, where most of the gravity vectors in the training set are densely distributed.

Different from previous approaches, our method does not need highly functional sensors like IMUs or other back-end systems for camera pose prediction. Furthermore, as our method does not rely on a particular backbone of depth estimation network and is computationally efficient, it can be easily integrated into a real-time application for scene understanding (Jiang et al., 2018; Chen et al., 2019) or scene reconstruction (Wang et al., 2018; Laidlow et al., 2019; Tateno et al., 2017).

For the experiments, we employed not only ScanNet (Dai et al., 2017) and NYUv2 (Silberman et al., 2012) datasets but also recorded a new dataset with a large roll or pitch camera movement for various indoor scenes. We evaluated our approach with three baseline methods, including data augmentation baseline, and verified that our gravity rectifier significantly improved depth prediction accuracy qualitatively and quantitatively. Moreover, we compared our method with state-of-the-art approaches with camera pose priors and achieved highly competitive accuracy with only RGB information.

In summary, our contributions are as follows: (1) We proposed a gravity rectifier that enables better performance by synthesizing gravity-aligned images for monocular depth estimation leveraging only RGB information without any external systems or sensors. (2) We created a new dataset including large camera rotation along with roll and pitch directions under various indoor scenes. (3) Our proposed method outperformed our baselines (which achieved 38% improvement in *abs_rel* over the vanilla model) and had comparable accuracy compared with state-of-the-art approaches.

2 RELATED WORK

2.1 Monocular Depth Estimation

Inferring depth from a single RGB image is an ill-posed problem as 3D points from multiple depth planes are projected to the same pixel on the image plane. Conventional approaches originally relied on stereo vision (Agarwal et al., 2009; Sinz et al., 2004) or different shading conditions (Zhang et al., 1999; Suwajanakorn et al., 2015). Recently, CNN-based depth prediction trained on large-scale datasets demonstrated promising results and enabled the production of reasonable depth maps (Eigen et al., 2014; Laina et al., 2016; Fu et al., 2018).

Nevertheless, they still have difficulties obtaining accurate depth under extreme circumstances. One of the main issues is the pure-rotation of the camera at inference time. Previous work (Dijk and Croon, 2019) analyzed CNN-based depth prediction, which performed poorly on images captured under unusual camera poses not included in the training data. To address this problem, we propose a novel depth prediction technique by refining camera poses to fill the gap between camera pose distribution in the training set and test set.

2.2 Gravity Estimation

Predicting gravity direction, i.e., estimating global scene orientation, is a fundamental task in computer vision. Conventional works (Lee and Yoon, 2015; Mirzaei and Roumeliotis, 2011) have leveraged visual cues such as vanishing points in indoor scenes to estimate gravity without external sensors like IMUs. Recently, learning-based approaches with deep regression models have been proposed by employing rich geometric representations extracted from an RGB image (Olmschenk et al., 2017; Xian et al., 2019). However, they have relied on the sophisticated network architecture of CNN or non-linear geometric optimization, which would be difficult for online gravity estimation.

Furthermore, the idea of predicting gravity from Visual SLAM has been proposed (Saito et al., 2020; Sartipi et al., 2020; Fei et al., 2019), which has enabled more accurate monocular depth estimation or surface normal estimation. Nevertheless, these methods require highly functional sensors like IMUs or offline pose estimation backbones.

Unlike previous methods that rely on external sensors or offline gravity estimation, we propose here a gravity rectifier network that directly regresses the gravity direction and can be trained with a depth estimation network in an end-to-end manner. Inspired by the spatial transformer network (Jaderberg et al., 2015), our network transforms a tilted image with homography warping induced from 3D rotation parameterized by gravity direction, improving the prediction accuracy of a depth map.

2.3 Rotation-Aware Prediction

Conventional CNN models fail in dense prediction tasks on images captured in uncommon camera poses, like tilted inputs. This is mainly caused by distribution bias with the training set and test set, e.g., training examples might be collected with minimal roll and pitch rotations, but the testing environment where users can control body-/robot-mounted cameras freely might capture images containing large roll and pitch rotations.

To overcome this issue, Saito et al. (Saito et al., 2020) and Sartipi et al. (Sartipi et al., 2020) proposed rectifying roll rotation of tilted images with camera poses from Visual-SLAM. Also, Zhao et al. (Zhao et al., 2021) incorporated encoded camera poses from IMUs into the depth prediction network directly. Nevertheless, they heavily relied on offline pose estimation systems (SLAM) or external sensors (IMUs), resulting in high computational costs or additional equipment.

Moreover, Do et al. (Do et al., 2020) proposed a new refinement method for tilted images in single-view surface normal prediction. They transformed the tilted images to rectified ones so that their surface normal distributions could be matched to those of the gravity-aligned images in the training data.

In this paper, we explore the benefit of spatial transformation to align tilted images to upright ones. Our proposed gravity rectifier, which can be trained in an end-to-end fashion only employed RGB information and corrected roll and pitch rotations with homography warping.

3 PROPOSED METHOD

Figure 2 shows the overview of our proposed networks. First, we input tilted images into a gravity estimator and predict gravity directions. Second, input images are warped with a gravity rectifier so that the estimated gravity directions is matched to the dominant direction of the gravity in the training set. Third, the rectified images are input into the depth prediction network. Finally, we re-warp the output depth map to the inverse direction so that the predicted depth map has the same resolution of the original image.

3.1 Gravity Rectifier

Given a tilted image I, we compute the gravity direction via a gravity predictor network formulated as a regression problem to produce gravity-aligned images $\overline{I}$ through a gravity rectifier.

The Gravity prediction network takes as input $\mathbf{q} \in \mathcal{R}(I)$ and outputs its gravity direction $\mathbf{g} \in \mathbb{R}^3$. Then, the gravity-aligned image is expressed as Eq.1

$$\overline{I}(\mathbf{q}) = I(\mathbf{W_g}(\mathbf{q})) \tag{1}$$

where $\mathbf{W_g}$ is the gravity rectifier that warps a tilted image I to the rectified image $\overline{I}$. Suppose now that $\mathbf{K}$ is the camera intrinsic matrix so the gravity rectifier is expressed as a homography induced by the camera rotation like Eq.2.

$$\mathbf{W_g} = \mathbf{K}\mathbf{R_g}\mathbf{K^{-1}} \tag{2}$$

Here, camera rotation $\mathbf{R_g}$ maps gravity direction $\mathbf{g}$ to dominant direction $\mathbf{a} \in \mathbb{R}^3$, written as Eq.3.

$$\mathbf{R_g} = \mathbf{I_3} + [\mathbf{g} \times \mathbf{a}]_\times + [\mathbf{g} \times \mathbf{a}]_\times^2 \times \frac{\mathbf{1}}{\mathbf{1} + \mathbf{a^T g}} \tag{3}$$

$\mathbf{I_3}$ is 3×3 the identity matrix, and $[\mathbf{g} \times \mathbf{a}]_\times$ is a skew-symmetric matrix of $\mathbf{g} \times \mathbf{a}$. Here, we define the

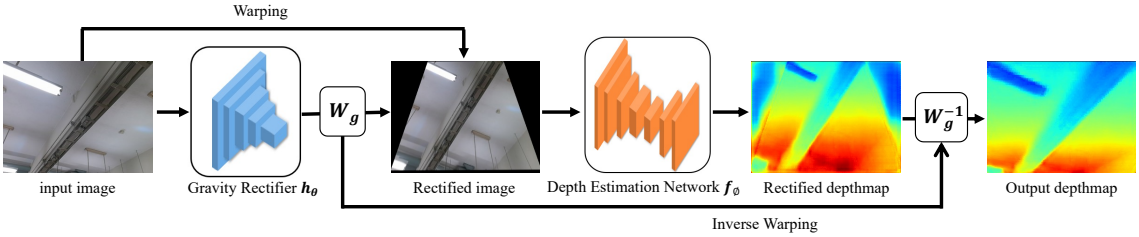

Figure 2: The overview of our proposed method. The gravity rectifier h_θ is learned to predict gravity direction $\mathbf{g}$ in the tilted image. This allows us to warp the tilted image to the rectified image whose gravity direction matches the gravity distribution of the training data. The gravity estimation network f_ϕ is used to predict the rectified depth maps and warp back to the tilted depth maps.

dominant direction $\mathbf{a} = [\mathbf{0}, \mathbf{1}, \mathbf{0}]^\mathbf{T}$ as a unit vector along the vertical axis of the camera, where the distribution of ground truth gravity directions in the training set is most densely distributed.

3.2 Network Architecture

We summarized the network architecture of both gravity rectifier h_θ and depth estimation network f_ϕ in Figure 3.

The gravity rectifier network h_θ predicts the gravity direction $\mathbf{g} \in \mathbb{R}^\mathbf{3}$ from tilted image $\mathbf{q} \in \mathcal{R}(I)$ as in Eq.4.

$$h_\theta(\mathbf{q}; I) = \mathbf{g}^\mathbf{T} \tag{4}$$

The architecture is built upon Resnet-18 (He et al., 2016). The last fully connected layer and softmax function, which was part of the original architecture, are replaced with our novel multilayer perceptron (MLP). The MLP is composed of two fully connected layers (128, 3 output channels each) and rectified linear units (ReLU) activation functions, yielding an output of 3×1 gravity vector.

Then, the depth estimation network f_ϕ takes gravity-aligned image $\mathbf{q} \in \mathcal{R}(\overline{I})$ as input and predicts its corresponding depth map. Our final depth $d_\mathbf{q} \in \mathbb{R}$ is obtained by applying an inverse warping of the gravity rectifier to the predicted depth map, as in Eq.5.

$$d_\mathbf{q} = \mathbf{v}^\mathbf{T}\mathbf{R_g}^\mathbf{T}\mathbf{f}_\phi(\mathbf{W_g^{-1}}(\mathbf{q}); \overline{I_t})\mathbf{K}^{-1}\mathbf{q_h} \tag{5}$$

where $\mathbf{R_g}$ and $\mathbf{W_g}$ are defined in section 3.1, and $\mathbf{v} = [\mathbf{0}, \mathbf{0}, \mathbf{1}]^\mathbf{T}$. We employ U-Net style architecture (Ronneberger et al., 2015) based on Resnet-50 (He et al., 2016) for the encoder part. We replaced the last average pooling and fully connected layers of the original Resnet-50 architecture with a convolutional layer and Batch Normalization instead, yielding a feature map with 1024 output channels. This feature map is then fed to a decoder part composed of successive series of bi-linear upsampling and convolutional layers with their skip connections. The convolution layer in our decoder applies to the concatenation of the block after bi-linear upsampling and the block in the encoder with the same spatial size.

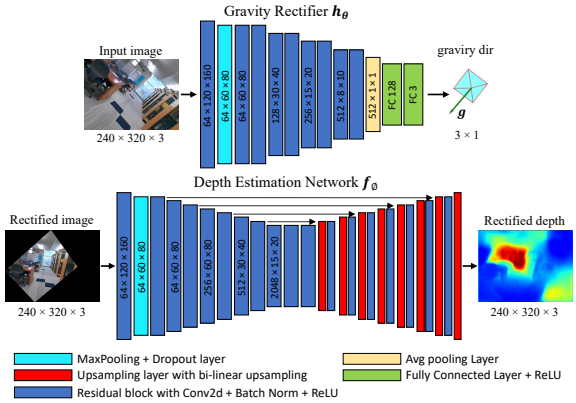

Figure 3: The architecture of our gravity rectifier and depth estimation network.

3.3 Loss Function

We learn the parameters of our total networks (ϕ and θ) by minimizing the following loss:

$$\mathcal{L}(\phi, \theta) = \sum_{\mathbf{q} \in \mathcal{R}(I)} \mathcal{L}_\phi(d_\mathbf{q}, \hat{d_\mathbf{q}}) + \lambda \mathcal{L}_\theta(\mathbf{g}^\mathbf{T}, \hat{\mathbf{g}}) \tag{6}$$

where $\hat{d_\mathbf{q}}$ is a ground truth depth map of tilted input I and $\hat{\mathbf{g}}$ is a ground truth gravity vector of I. λ is a scalar parameter balancing both the gravity prediction network and the depth prediction network.

For gravity predictor network loss $\mathcal{L}_\theta$, we employ truncated angular loss as proposed in (Do et al., 2020) to avoid a vanishing gradient around the small angular error as follows:

$$\mathcal{L}_\theta(\mathbf{g^T}, \hat{\mathbf{g}}) = \begin{cases} 0 & (1-\varepsilon \leq \mathbf{g^T}\hat{\mathbf{g}}) \\ \cos^{-1}(\mathbf{g^T}, \hat{\mathbf{g}}) & (0 \leq \mathbf{g^T}\hat{\mathbf{g}} < \mathbf{1}-\varepsilon) \\ \frac{\pi}{2} - \mathbf{g^T}\hat{\mathbf{g}} & (\mathbf{g^T}\hat{\mathbf{g}} < \mathbf{0}) \end{cases} \quad (7)$$

,where $\varepsilon = 10^{-6}$.

For the depth prediction network loss $\mathcal{L}_\phi$, we adopt the loss function proposed by (Alhashim and Wonka, 2018) composed of mean absolute error (MAE) and structural similarity (SSIM) (Wang et al., 2004) loss:

$$\mathcal{L}_\phi(d_\mathbf{q}, \hat{d_\mathbf{q}}) = \gamma \mathcal{L}_{MAE}(d_\mathbf{q}, \hat{d_\mathbf{q}}) + \mathcal{L}_{SSIM}(d_\mathbf{q}, \hat{d_\mathbf{q}}) \quad (8)$$

where

$$\mathcal{L}_{MAE}(d_\mathbf{q}, \hat{d_\mathbf{q}}) = \frac{1}{n}\sum_\mathbf{q} |d_q - \hat{d_q}| \quad (9)$$

$$\mathcal{L}_{SSIM}(d_\mathbf{q}, \hat{d_\mathbf{q}}) = \frac{1 - SSIM(d_\mathbf{q}, \hat{d_\mathbf{q}})}{2} \quad (10)$$

n is the total number of pixels of depth map $d_\mathbf{q}$ and γ is a scalar parameter for the MAE term which is set as $\gamma = 0.1$.

4 EXPERIMENTS

4.1 Evaluation Dataset

To evaluate our proposed method, we employed publicly available RGB-D datasets of ScanNet (Dai et al., 2017) and NYUv2 (Silberman et al., 2012) that are almost composed of upright scenes. However, these datasets are not sufficient for evaluating the robustness of our method since they are captured under limited camera motions containing minimal pitch and roll rotations. Therefore, we recorded a new dataset composed of tilted scenes at various indoor scenes with Kinect Azure.

ScanNet (Dai et al., 2017): an RGB-D video dataset containing a large variety of indoor scenes. We used the 20,942 images from their standard testing split. For the ground truth gravity vector, we calculated the ground plane's normal direction from semantic labels and its point cloud.

NYUv2 (Silberman et al., 2012): an RGB-D dataset captured with MS Kinect V1. We employed a labeled sequence for testing, which contains 654 image pairs. We employed accelerometer data in the dataset for the ground truth gravity vector.

OurDataset: We collected a new free-hand dataset with Kinect Azure that included 12 different scenes. Each RGB-D image pair was recorded in the resolution of 480×640 and ground truth gravity vector from IMU with 30 frames per second (FPS). Two types of scenes are collected. (i) Roll-rotated scenes: We applied strong roll rotation of the camera ranging from -90°to 90°. We captured 1,520 images composed of six sequences. (ii) Pitch-rotated scenes: We also captured images with pitch rotation from -45°to 45°. We captured 1,717 images composed of six sequences. In each scene, roll and pitch angles are uniformly distributed.

4.2 Network Training

We trained our model with a standard training/validation split of ScanNet (Dai et al., 2017): 189,916 images for training, 53,193 images for validation. We employed a batch size of 32 and optimized using Adam (Kingma and Ba, 2015) with a learning rate of 1.0×10^{-4}. The model converged after 40 epochs, which takes about 30 hours on a GeForce RTX 3090 GPU (24 GB of memory). All frames were resized into resolution 240×320. For our loss function of Eq.6, we used $\lambda = 0.01$. The weights of Resnet-18 and Resnet-50 (He et al., 2016) in both the gravity prediction network and depth prediction network were initialized with the pre-trained ImageNet (Deng et al., 2009). Our code, dataset are available on GitHub[1].

4.3 Evaluation Details

We evaluated our method with baselines quantitatively and qualitatively. We set our own baselines:

- **ResnetUnet**: We trained the depth prediction network described in Section 3.2 without any data augmentation or pose rectification.
- **ResnetUnet+AUG**: We trained the depth prediction network with data augmentation (AUG) by synthesizing tilted images with random camera rotation around the roll and pitch directions. We applied random roll rotation from -90°to 90°and pitch rotation from -45°to 45°. Note that this model does not take rectified images as input.
- **ResnetUnet+IMU**: We trained the depth prediction network with gravity-aligned images using ground truth gravity vectors. We employed ground plane normal direction as gravity vectors in ScanNet (Dai et al., 2017).
- **ResnetUnet+GR (Ours)**: We trained the depth prediction network with gravity rectifier (GR) described in Section 3.1.

[1] https://github.com/WeLoveKiraboshi/DeepTiltedDepthEstimation

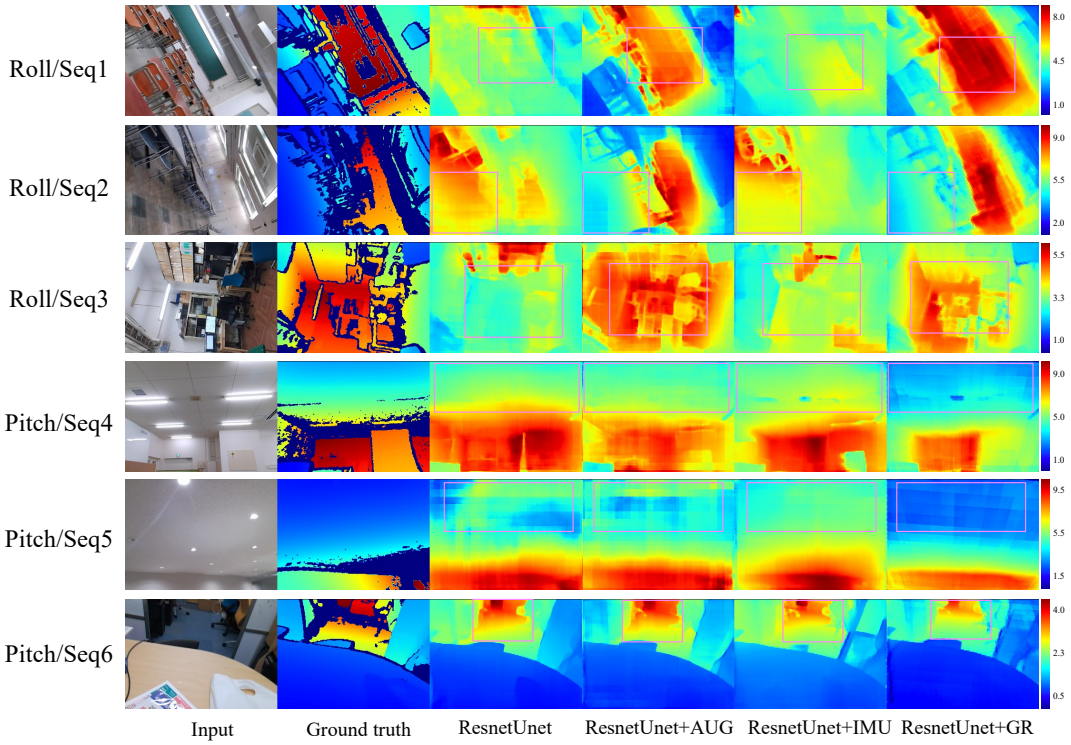

Figure 4: Qualitative results on our test dataset compared with our baseline methods: ResnetUnet, ResnetUnet+AUG, ResnetUnet+IMU. Note that the depth pixel colored in red shows that the depth is a large value, and the pixel colored in blue shows that the depth is a small value.

We also compared our approach with state-of-the-art methods of monocular depth estimation with camera pose priors.

- **Saito et al. (Saito et al., 2020)**: A training-free depth prediction approach for roll-rotated scenes with offline pose estimation from RGB-SLAM. For the depth prediction network, we input gravity-aligned images warped from tilted inputs with affine transformation. We employed the weight of ResnetUnet trained without any data augmentation and pose rectification.
- **Zhao et al.**(CPP) **(Zhao et al., 2021)**: A method to estimate depth from concatenated images of RGB input and a 2D map encoded from a ground truth camera pose. We calculated ground truth pitch angle, roll, angle, and camera height from the ground plane in ScanNet (Dai et al., 2017).
- **Zhao et al.**(CPP_{pred}) **(Zhao et al., 2021)**: A method to estimate depth with a 2D map encoded from a predicted camera pose of CNN. For the pose prediction network, we employed the same architecture of our gravity rectifier with Resnet-18 backbone (He et al., 2016). We initialized the network weight with ImageNet pre-trained (Deng et al., 2009). We also applied random augmentation for input: roll rotation ranging from -90°to 90°and pitch rotation ranging from -45°to 45°.
- **Zhao et al.**$(CPP+PDA)$ **(Zhao et al., 2021)**: A method to estimate depth with a CPP map and data augmentation by synthesizing tilted images with random camera rotation (PDA). We applied roll rotation ranging from -90°to 90°and pitch rotation ranging from -45°to 45°for augmentation.

4.4 Evaluation Metrics

We evaluated the accuracy of predicted depth maps with the standard four metrics used in prior works(Eigen et al., 2014; Alhashim and Wonka, 2018): (a) mean absolute relative error (*abs_rel*), (b) mean squared relative error (*sq_rel*), (c) root mean squared error (*rmse*), (d) threshold accuracy (δ_i) for which $max(\frac{d_{\mathbf{q}}}{\hat{d}_{\mathbf{q}}}, \frac{\hat{d}_{\mathbf{q}}}{d_{\mathbf{q}}}) < 1.25^i (i = 1,2,3)$, where $d_{\mathbf{q}}$ is the predicted depths and $\hat{d}_{\mathbf{q}}$ is the ground truth depths.

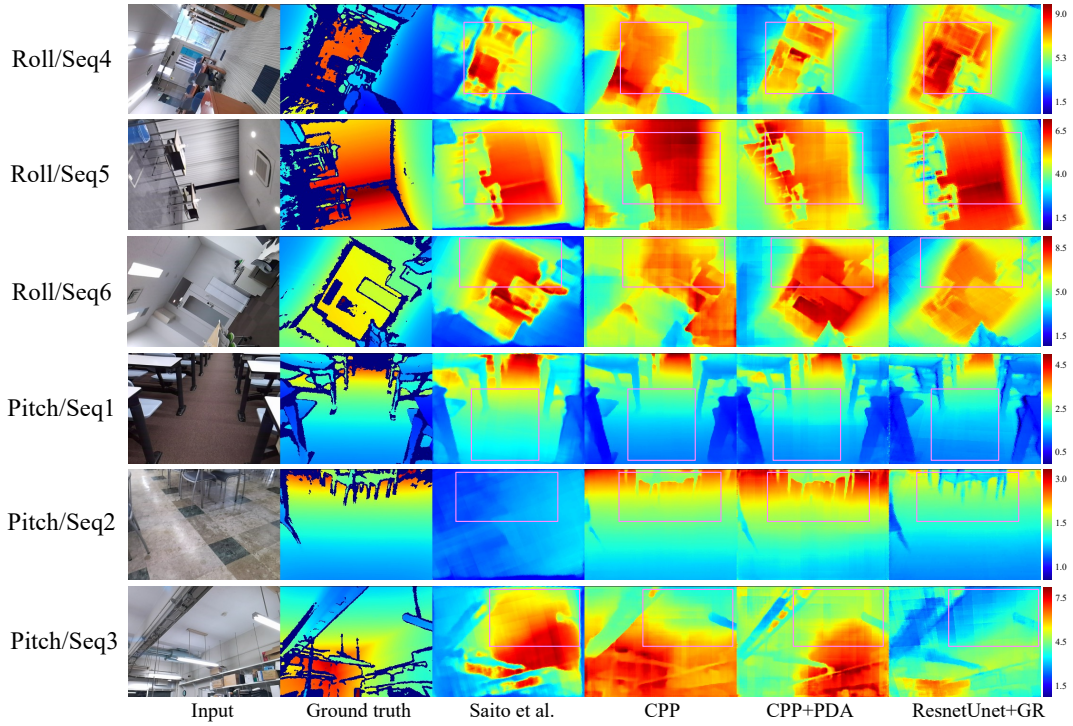

Figure 5: Qualitative results on our test dataset compared with state-of-the-art methods: Saito et al. (Saito et al., 2020) and Zhao et al. (Zhao et al., 2021). Note that the depth pixel colored in red shows that the depth is a large value, and the pixel colored in blue shows that the depth is a small value.

5 RESULTS

5.1 Qualitative Evaluation

Figure 4 shows the qualitative results of our predicted depth map tested on our test dataset compared with our own baselines (ResnetUnet, ResnetUnet+AUG, ResnetUnet+IMU). Although some of our baselines (ResnetUnet, ResnetUnet+IMU) failed to estimate a reasonable depth map, especially in roll-rotated scenes, our predicted depth map (ResnetUnet+GR) had a more plausible appearance to ground truth depth, as well as an augmented model (ResnetUnet+AUG). This performance degradation is caused by a domain gap between the training set and the test set, i.e., the training set is mainly composed of gravity-aligned images while the testing tilted images have large roll and pitch angles.

We also summarized the qualitative results compared with state-of-the-art methods with camera pose priors tested on our test dataset in Figure 5. Saito et al. (Saito et al., 2020) which rectify tilted inputs with RGB-SLAM, seemed to make erroneous predictions in large pitch-rotated scenes since they only considered the rectification of roll rotation with 2D affine transformation. On the other hand, Zhao et al. (CPP, CPP+PDA) (Zhao et al., 2021) seemed to predict a more reliable depth map in both roll- and pitch- rotated scenes due to its strong prior of ground truth camera poses. While these prior works completely relied on external sensors or systems like IMU and SLAM, our proposed method (ResnetUnet+GR) successfully produced visually improved results even though we only employed RGB information for prediction.

5.2 Quantitative Evaluation

Table 1 shows the quantitative results of our proposed method evaluated on gravity-aligned scenes from test sequences of ScanNet (Dai et al., 2017) and NYUv2 (Silberman et al., 2012) datasets. We observed that all networks performed excellently on unseen gravity-aligned frames in ScanNet, as the dataset contains sufficient scene diversity. Our proposed method (ResnetUnet+GR) slightly underperformed compared to Zhao et al. (*CPP*) due to the lack of ground truth camera pose information and suffering from a prediction on the feature-less part of the scene, e.g., floor

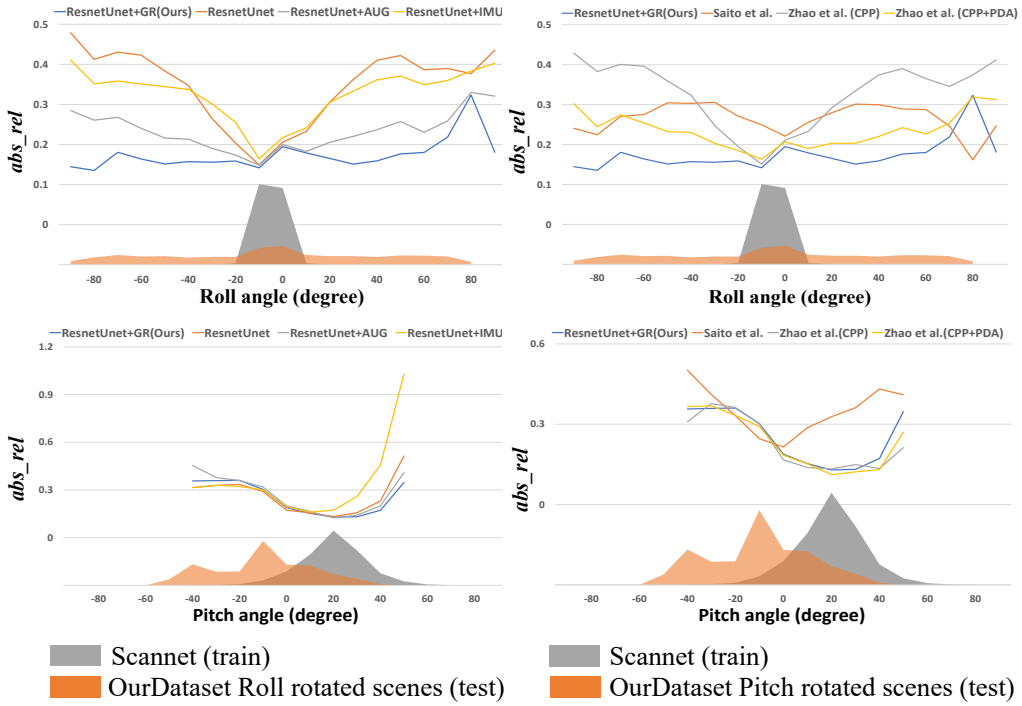

Figure 6: The correlation between the rotation angle of the camera pose and absolute relative error (*abs_rel*) was evaluated on our test dataset. The horizontal axis shows the ground truth camera rotation angle (roll/pitch), and the vertical axis shows the *abs_rel* value. Camera pose distribution in the training set is shown in gray, and the distribution in the test set is shown in orange.

Table 1: The quantitative results of our method with all baselines on gravity-aligned scenes from test sequences of ScanNet and NYUv2.

Method	ScanNet						NYUv2					
	abs_rel ↓	*sq_rel* ↓	*rmse* ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑	*abs_rel* ↓	*sq_rel* ↓	*rmse* ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
ResnetUnet	0.133	0.068	0.317	0.814	0.951	0.986	0.179	0.154	0.648	0.711	0.917	0.973
ResnetUnet+AUG	0.145	0.078	0.348	0.787	0.943	0.983	0.196	0.180	0.707	0.661	0.891	0.966
ResnetUnet+IMU	0.211	0.134	0.419	0.695	0.900	0.968	0.197	0.176	0.692	0.682	0.904	0.972
ResnetUnet+GR (Ours)	0.132	0.068	0.313	0.818	0.951	**0.987**	**0.171**	**0.147**	**0.619**	**0.734**	**0.920**	**0.976**
Zhao et al. (*CPP*)	**0.114**	**0.059**	**0.274**	**0.855**	**0.957**	0.986	0.205	0.178	0.686	0.663	0.900	0.971
Zhao et al. (CPP_{pred})	0.136	0.073	0.309	0.824	0.950	0.984	0.176	0.151	0.633	0.725	0.915	0.974
Zhao et al. ($CPP+PDA$)	0.135	0.073	0.330	0.811	0.947	0.983	0.199	0.186	0.709	0.665	0.897	0.969

and walls. Nevertheless, our method outperformed all baselines in NYUv2. Since the scenes in NYUv2 contained rich geometric features rather than ScanNet, our depth prediction network realized reasonable prediction.

For tilted scenes, we summarized the quantitative results of our proposed method evaluated on our test dataset in Table 2. Our proposed method shows excellent performances compared to our baselines on both roll- and pitch-rotated scenes with unseen large camera rotation, e.g., the percentage drop in *abs_rel* for ResnetUnet (by 29%), ResnetUnet+AUG (by 15%), ResnetUnet+IMU (by 33%). Our method also significantly outperformed other state-of-the-art methods (Saito et al., 2020; Zhao et al., 2021) on roll-rotated scenes, though these methods heavily relied on ground truth camera poses from IMU or offline camera pose estimation like SLAM. In pitch-rotated scenes, our proposed method achieved on-par performance with Zhao et al.(CPP, CPP+PDA) (Zhao et al., 2021). There are two main explanations for this fact: First, the distribution gap between the pitch rotation of the training set and the test set was less dissociated than for roll-rotated scenes. Second, Zhao et al. (Zhao et al., 2021) employed ground truth camera poses for their prediction while we only leveraged image frames with RGB information.

We also summarized the relationship between camera rotation angle and the error rate on roll- and pitch- rotated scenes from our dataset in Figure 6. As can be seen in the left column of Figure 6, the errors around where the training set distribution is densely

Table 2: The quantitative results of our method with all baselines on our test dataset on roll- and pitch-rotated scenes.

Method	Roll-rotated scenes								
	rgb	imu	SLAM	*abs_rel* ↓	*sq_rel* ↓	*rmse* ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
ResnetUnet	✓			0.317	0.976	2.266	0.476	0.584	0.661
ResnetUnet+AUG	✓			0.216	0.476	1.563	0.551	0.731	0.871
ResnetUnet+IMU	✓			0.299	0.881	2.219	0.481	0.582	0.673
ResnetUnet+GR (Ours)	✓			**0.166**	**0.286**	**1.101**	**0.698**	**0.885**	**0.955**
Saito et al.	✓		✓	0.262	0.622	1.663	0.568	0.567	0.567
Zhao et al. (*CPP*)	✓	✓		0.299	0.877	2.136	0.472	0.602	0.688
Zhao et al. (CPP_{pred})	✓			0.290	0.856	2.098	0.507	0.617	0.696
Zhao et al. ($CPP+PDA$)	✓	✓		0.219	0.424	1.495	0.551	0.748	0.883
Method	Pitch-rotated scenes								
	rgb	imu	SLAM	*abs_rel* ↓	*sq_rel* ↓	*rmse* ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
ResnetUnet	✓			0.217	0.364	1.265	0.568	0.770	0.905
ResnetUnet+AUG	✓			0.230	0.398	1.308	0.561	0.743	0.872
ResnetUnet+IMU	✓			0.267	0.430	1.360	0.502	0.719	0.885
ResnetUnet+GR (Ours)	✓			0.213	0.334	1.195	0.571	**0.783**	**0.934**
Saito et al.	✓		✓	0.320	0.616	1.580	0.431	0.432	0.433
Zhao et al. (*CPP*)	✓	✓		0.202	0.346	1.222	**0.594**	0.780	0.903
Zhao et al. (CPP_{pred})	✓			0.232	0.379	1.271	0.551	0.722	0.899
Zhao et al. ($CPP+PDA$)	✓	✓		**0.200**	**0.330**	**1.165**	0.582	0.770	0.907

populated do not show any difference between our method vs. the baselines. However, in larger rotation angles, even though the errors of the baselines increased, the error of our proposed method does not depend on rotation angles, which shows the effectiveness of our proposed method.

5.3 Network Efficiency

We finally compared our proposed method (ResnetUnet+GR) with other baselines in terms of the number of parameters, actual memory consumption, number of floating operations (FLOPS), and inference time as summarized in Table 3. Although our proposed method performed 1.3x larger memory consumption with the vanilla model (ResnetUnet), due to the additional network parameter of gravity rectifier, we achieved 66.8 FPS for our prediction, which is highly sufficient for real-time applications like real robot navigation systems. Since we did not employ offline pose prediction systems like SLAM in the back-end, we successfully realized our speed-up (1.25x faster than Saito et al. (Saito et al., 2020)).

5.4 Application to SLAM

To complement our results, we demonstrated the effectiveness of our proposed depth prediction within the SLAM applications. We integrated our depth prediction into a CNN-MonoFusion (Wang et al., 2018), which reconstructed dense 3D maps by integrating monocular depth estimation with CNN into conventional camera-tracking systems like ORB-SLAM2 (Mur-Artal and Tardós, 2017).

Figure 7 shows the reconstruction result of our proposed method and other baseline methods tested on our dataset with roll- and pitch- rotated scenes. As it can be seen, our proposed model (ResnetUnet+GR) successfully yielded more accurate reconstruction of the scene, compared to the vanilla model which failed to reconstruct reasonable scene geometry (e.g., there is some misalignment in parts of the floor in Roll/seq4). The improved accuracy of depth prediction with our gravity rectifier is not only obvious in Figure 4, but also in real-time applications like SLAM. We figured out our proposed method provide a more robust system for AR and robotics applications where users manipulate the device freely and can cause significant camera orientation.

6 CONCLUSION

In this paper, we proposed a gravity rectifier, a novel rectification approach to improve the accuracy of monocular depth estimation for tilted images, leveraging only RGB information. Our gravity rectifier is learned to transform a tilted image into a gravity-aligned image and can be trained jointly with the depth estimation network in an end-to-end fashion. To show the effectiveness of our method, we evaluated our method both qualitatively and quantitatively using our own dataset with large roll and pitch camera rotations. The results showed that our approach significantly outperformed baselines, including data augmentation, and has competitive accuracy

Table 3: Network efficiency of our proposed method with all baselines in terms of the number of parameters, memory consumption, FLOPS, and inference time (FPS with batch size 1). We employed our test dataset including both roll- and pitch- rotated scenes.

Network	Backbone	Params	Memory (MB)	FLOPS(GB)	FPS
ResnetUnet+GR (Ours)	Resnet-50+Resnet-18	47.5M	190.1	39.5	66.8
ResnetUnet	Resnet-50	36.3M	145.1	36.7	94.9
Saito et al.	Resnet-50	36.3M	145.1	36.7	53.5
Zhao et al. (*CPP*)	Resnet-50	36.3M	145.2	36.8	73.9
Zhao et al. (CPP_{pred})	Resnet-50+Resnet-18	47.5M	190.1	39.6	63.3

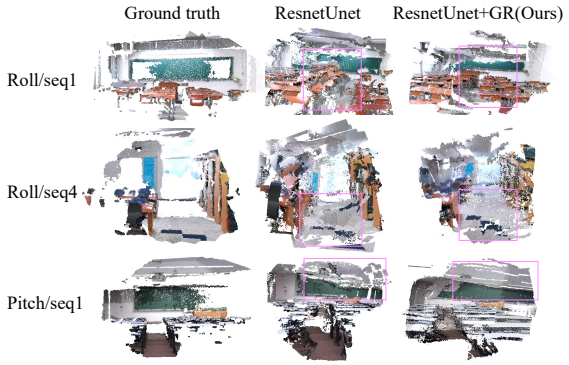

Figure 7: Dense reconstruction result of CNN-MonoFusion (Wang et al., 2018) evaluated on our test dataset. From left to right: result with ground truth depth, predicted depth from ResnetUnet, predicted depth from ResnetUnet+GR (Ours).

as well as state-of-the-art methods with external sensor or offline pose estimation systems.

Acknowledgement

This work was conducted on JEMARO (Japan-Europe Master on Advanced Robotics), supported by IUEP EU-Japan, MEXT and Erasmus+ programme.

REFERENCES

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2009). Building rome in a day. In *IEEE International Conference on Computer Vision*, pages 72–79. IEEE.

Alhashim, I. and Wonka, P. (2018). High quality monocular depth estimation via transfer learning. In *arXiv preprint arXiv:1812.11941*.

Chen, P.-Y., Liu, A. H., Liu, Y.-C., and Wang, Y.-C. F. (2019). Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2624–2632. IEEE.

Czarnowski, J., Laidlow, T., Clark, R., and Davison, A. J. (2020). Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728.

Dai, A., X. Chang, A., Savva, M., Halber, M., A. Funkhouser, T., and Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 5828–5839. IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

Dijk, T. V. and Croon, G. D. (2019). How do neural networks see depth in single images? In *IEEE International Conference on Computer Vision*, pages 2183–2191. IEEE.

Do, T., Vuong, K., Roumeliotis, S. I., and Park, H. S. (2020). Surface normal estimation of tilted images via spatial rectifier. In *European Conference on Computer Vision*, pages 265–280. Springer.

Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, volume 27, pages 2366–2374. Neural information processing systems foundation.

Fei, X., Wong, A., and Soatto, S. (2019). Geo-supervised visual depth prediction. *IEEE Robotics and Automation Letters*, 4(2):1661–1668.

Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D. (2018). Deep ordinal regression network for monocular depth estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2002–2011. IEEE.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.

Godard, C., Mac Aodha, O., Firman, M., and Brostow, G. J. (2019). Digging into self-supervised monocular depth estimation. In *IEEE International Conference on Computer Vision*, pages 3828–3838. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015).

Spatial transformer networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 2017–2025. Neural information processing systems foundation.

Jiang, H., Larsson, G., Shakhnarovich, M. M. G., and Learned-Miller, E. (2018). Self-supervised relative depth learning for urban scene understanding. In *European Conference on Computer Vision*, pages 19–35. IEEE.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *IEEE International Conference on Learning Representations*, pages 1–15. IEEE.

Laidlow, T., Czarnowski, J., and Leutenegger, S. (2019). Deepfusion: Real-time dense 3d reconstruction for monocular slam using single-view depth and gradient predictions. In *IEEE International Conference on Robotics and Automation*, pages 4068–4074. IEEE.

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., and Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. In *IEEE International Conference on 3D Vision*, pages 239–248. IEEE.

Lee, J.-K. and Yoon, K.-J. (2015). Real-time joint estimation of camera orientation and vanishing points. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1866–1874. IEEE.

Luo, X., Huang, J.-B., Szeliski, R., Matzen, K., and Kopf, J. (2020). Consistent video depth estimation. *ACM Transactions on Graphics*, 39(4):156–167.

Marcu, A., Costea, D., Licaret, V., Pîrvu, M., Slusanschi, E., and Leordeanu, M. (2018). Safeuav: Learning to estimate depth and safe landing areas for uavs from synthetic data. In *European Conference on Computer Vision Workshops*, pages 43–58. IEEE.

Mi, L., Wang, H., Tian, Y., He, H., and Shavit, N. N. (2022). Training-free uncertainty estimation for dense regression: Sensitivity as a surrogate. In *AAAI Conference on Artificial Intelligence*. AAAI Press.

Mirzaei, F. M. and Roumeliotis, S. I. (2011). Optimal estimation of vanishing points in a manhattan world. In *IEEE International Conference on Computer Vision*, pages 2454–2461. IEEE.

Mur-Artal, R. and Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Olmschenk, G., Tang, H., and Zhu, Z. (2017). Pitch and roll camera orientation from a single 2d image using convolutional neural networks. In *IEEE Canadian Conference on Computer and Robot Vision*, pages 261–268. IEEE.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, volume 9351, pages 234–241. Springer.

Saito, Y., Hachiuma, R., Yamaguchi, M., and Saito, H. (2020). In-plane rotation-aware monocular depth estimation using slam. In *International Workshop on Frontiers of Computer Vision*, pages 305–317. Springer.

Sartipi, K., Do, T., Ke, T., Vuong, K., and Roumeliotis, S. I. (2020). Deep depth estimation from visual-inertial slam. In *IEEE International Conference on Intelligent Robots and Systems*, pages 10038–10045. IEEE.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer.

Sinz, F. H., Candela, J. Q., Bakır, G. H., Rasmussen, C. E., and Franz, M. O. (2004). Learning depth from stereo. In *Joint Pattern Recognition Symposium*, pages 245–252. Springer.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). A benchmark for the evaluation of rgb-d slam systems. In *IEEE International Conference on Intelligent Robot Systems*, pages 573–580. IEEE.

Suwajanakorn, S., Hernandez, C., and Seitz, S. M. (2015). Depth from focus with your mobile phone. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3497–3506. IEEE.

Tateno, K., Tombari, F., Laina, I., and Navab, N. (2017). Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 6243–6252. IEEE.

Wang, J., Liu, H., Cong, L., Xiahou, Z., and Wang, L. (2018). Cnn-monofusion: Online monocular dense reconstruction using learned depth from single view. In *IEEE International Symposium on Mixed and Augmented Reality Adjunct*, pages 57–62. IEEE.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transaction on Image Processing*, 13(4):600–612.

Xian, W., Li, Z., Fisher, M., Eisenmann, J., Shechtman, E., and Snavely, N. (2019). Uprightnet: Geometry-aware camera orientation estimation from single images. In *IEEE International Conference on Computer Vision*, pages 9974–9983. IEEE.

Yang, X., Chen, J., Dang, Y., Luo, H., Tang, Y., Liao, C., Chen, P., and Cheng, K.-T. (2019). Fast depth prediction and obstacle avoidance on a monocular drone using probabilistic convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, 22(1):156–167.

Zhang, R., Tsai, P.-S., Cryer, J. E., and Shah, M. (1999). Shape-from-shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706.

Zhang, Z., Xiong, M., and Xiong, H. (2019). Monocular depth estimation for uav obstacle avoidance. In *IEEE International Conference on Cloud Computing and Internet of Things*, pages 43–47. IEEE.

Zhao, Y., Kong, S., and Fowlkes, C. (2021). Camera pose matters: Improving depth prediction by mitigating pose distribution bias. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 15759–15768. IEEE.