In-Plane Rotation-Aware Monocular Depth Estimation Using SLAM

Yuki Saito(✉), Ryo Hachiuma(✉), Masahiro Yamaguchi(✉),
and Hideo Saito(✉)

Department of Information and Computer Science, Keio University, Yokohama, Japan
{yusa19971015,ryo-hachiuma,yama-1467,hs}@keio.jp

Abstract. Estimating accurate depth from an RGB image in any environment is challenging task in computer vision. Recent learning based method using deep Convolutional Neural Networks (CNNs) have driven plausible appearance, but these conventional methods are not good at estimating scenes that have a pure rotation of camera, such as in-plane rolling. This movement imposes perturbations on learning-based methods because gravity direction is considered to be strong prior to CNN depth estimation (i.e., the top region of an image has a relatively large depth, whereas bottom region tends to have a small depth). To overcome this crucial weakness in depth estimation with CNN, we propose a simple but effective refining method that incorporates in-plane roll alignment using camera poses of monocular Simultaneous Localization and Mapping (SLAM). For the experiment, we used public datasets and also created our own dataset composed of mostly in-plane roll camera movements. Evaluation results on these datasets show the effectiveness of our approach.

Keywords: Monocular depth estimation · Simultaneous Localization and Mapping · Convolutional Neural Network

1 Introduction

Depth estimation from an RGB image, i.e., predicting the per-pixel distance to the camera, has many applications, such as Augmented Reality (AR) [1], autonomous driving [2], robot application [3], etc. Given a single image, recent efforts to estimate depths from a single image have yielded high-quality outputs by taking advantages of fully convolutional neural networks (CNNs) [4,5] and large amount of training data from indoor [6] and outdoor [7] scenes.

Monocular depth estimation using CNN implicitly assumes that the camera orientation along the roll direction (in-plane rotation[1]) is almost same in every input image. This is because a person generally takes a photograph with the vertical axis of the image parallel to the direction of gravity. According to this

[1] A rotary motion around an optical axis in camera coordinate system.

W. Ohyama and S. K. Jung (Eds.): IW-FCV 2020, CCIS 1212, pp. 305–317, 2020.
https://doi.org/10.1007/978-981-15-4818-5_23

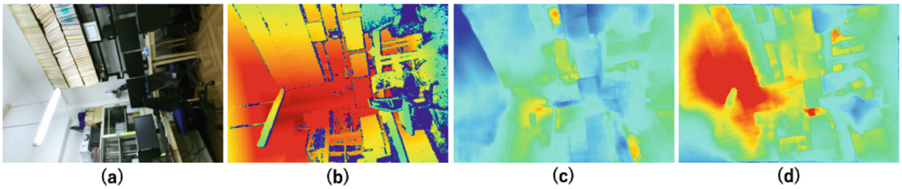

Fig. 1. CNN depth estimation in the in-plane rolled scene. (a) the input image, (b) the ground truth depth image. (c) the result using the conventional method, (d) the result using our proposed method. The pixels with large depth values are colored in red and the pixels with small depth values are colored in blue. (Color figure online)

implicit assumption, the orientation is a strong prior for inferring the depth information [8] in the monocular depth estimation using CNN. For example, the network implicitly learns that the lower side of the image is closer than the upper side of the image. This assumption is learned from the training dataset and is reasonable in its application to autonomous driving, because the position of the camera is fixed against the vehicle, and the camera itself does not move drastically. In contrast, when applied to AR or drone cameras, the user moves the smartphone/camera freely, and thus, this assumption collapses. Hence, when the camera rotates along in-plane direction, the accuracy of monocular depth estimation significantly drops. Figure 1(c) shows an failure example of monocular depth prediction. It is evident that the depth is not correctly estimated against the whole image because the input RGB image, Fig. 1(a), is rotated.

In order to overcome this crucial weakness of monocular depth prediction using CNN, we propose using the Simultaneous Localization and Mapping (SLAM) system to improve the accuracy when estimating monocular depth against a rolled image. Using the SLAM system, the 6 Degrees of Freedom (6DoF) camera trajectories are estimated from the RGB image sequences. As the camera pose consists of the translation and orientation of the camera with respect to the initial frame, we can extract the roll rotation angle from the camera pose. By using that angle, an RGB image can be transformed as if the camera was not rotated. Then, the transformed image is fed to the neural network to obtain the depth of the image. Finally, the predicted depth image is transformed inversely using the extracted angle. As a result, we can predict the accurate depth against the rolled image. Figure 1(d) shows the depth prediction result of our proposed method. It is evident that the structure of the scene can be more clearly inferred than the conventional depth prediction in Fig. 1(c). As this is the first work which tackles depth estimation against a roll-rotated image, we assume that the camera is not rotated in the initial frame. This is reasonable assumption because in most cases when pictures are taken, it is implicitly assumed that the direction of gravity is almost parallel to the vertical axis of the image.

In this paper, the sequences in which the camera is rotated in the roll direction are necessary for evaluating our proposed method. In the TUM RGB-D dataset [9], which is a public dataset for evaluating monocular depth estimation, there are three sequences in which the camera rotates along the roll-pitch-yaw direction. However, three sequences are not sufficient for evaluating the robustness of the method. Therefore, we recorded another dataset by ourselves. In this dataset, there are six sequences which are recorded at different indoor locations and the camera is rotated drastically in each sequence. In the experiments, we demonstrate the performance of our system on sequences from the TUM RGB-D dataset as well as the self-created dataset. As a result, our method significantly improved the depth estimation accuracy in two evaluation metrics on our dataset from the baseline method.

Although our method is simple and requires not an single RGB image but RGB image sequences, it significantly improves the depth estimation accuracy of the roll-rotated image using only RGB information. Our method needs neither any additional sensors like inertial measurement unit (IMU) [10] nor any cost for re-training the CNN network. Furthermore, as our method does not rely on a particular backbone of the depth estimation network or SLAM system and is not computationally high, it can be easily integrated into a real-time monocular dense reconstruction system using a depth prediction network, such as CNN-SLAM [11], DeepFusion [12], or CNN-MonoFusion [1].

There are three contributions in this paper. First, we first propose a method using the SLAM system for monocular depth estimation that is robust for in-plane rotation. Second, we created a dataset in which the sequences that were recorded by the camera were rotating in a roll direction. Third, our proposed method outperformed the baseline in two evaluation metrics on our dataset.

2 Related Work

2.1 In-Plane Rotation-Aware Prediction

While there are no previous work that regard on CNN depth estimation for overcoming the dependency of in-plane rotation, Toyoda et al. [13] have tried to reduce the dependency of in-plane rotation by selecting the most consistent pose from images rolled at various angles of wild motion video in Deep Neural Network (DNN) based pose estimation. There are rare scenes such that subjects are upside down in the real world and these rare data were not learned generally in datasets. Hence, to save the cost of training data again, they calculated roll angles using the output joint position probability. However, in terms of depth estimation, the confidence of the depth are not outputted typically. Furthermore, rotating images by every quantized roll angle is computationally expensive, which is a crucial problem for real-time applications like SLAM. Therefore, we use SLAM tracking to calculate the scenes' in-plane rotation angles, which is more precise and computationally cheap.

Kurz and Benhimane have proposed gravity-aware AR [14], in which the gravity direction measured with an IMU improves the accuracy of the camera pose estimation. This is related to our method because gravity direction is used to improve the accuracy of vision-based 3D sensing. Different from them, we propose an approach without relying on highly functional sensors. Our system can work only from RGB images.

2.2 SLAM with Monocular Depth Estimation

Enormous monocular SLAM or Visual Odometry approaches have been developed for motion estimation and are divided into feature-based methods [15,16] and direct methods [17–19]. However, they only provide sparse or semi-dense depth maps and cannot estimate camera poses accurately by pure rotational motions even in high-textured scenes.

In order to reconstruct dense 3D maps and improve the trajectory using these points, combining SLAM with CNN depth estimation have been proposed, which produced higher benchmark scores rather than conventional monocular systems. One of the most accurate SLAM/CNN network combination is CNN-SLAM [11] where CNN's learned depth maps are fused into direct SLAM framework. Though CNN-SLAM guarantees the strong estimation of accurate trajectory and dense maps, it does have CNN's inaccuracy against rotated inputs, because depth maps are produced by KeyFrames which are not created unless the camera translates over a certain distance. In addition, CNN-MonoFusion [1] evaluates 3D reconstruction models whose scenes have pure-rotational motion, such as TUM RGB-D dataset's *rpy* sequences [9]. There is no quantitative evaluations about CNN depth accuracy.

In contrast to these conventional approaches, we directly face the weakness of learning-based depth estimation and measure our refinement's efficiency from both qualitative and quantitative perspectives.

2.3 Learning Based Rotation Prediction

The learning-based approaches to predict camera rotation from a single RGB image have been proposed. Fischer et al. [20] constructed a network that directly regresses the orientation angle of an image. Moreover, Olmschenk et al. [21] proposed a method that estimates pitch and roll angles of the camera by using CNN from only an RGB image. However, these methods can only estimate a rough orientation angle which is insufficient in accuracy and have no geometric constraints (i.e., CNN outputs statistical likelihood values only based on learning data).

In another perspective, Xian et al. [22] estimated 2DoF camera orientation using both local and global scene representations extracted from an RGB image. Nevertheless, they verify the effectiveness of their method only with small roll angles, such as ±20° or ±50°.

Our method can estimate geometric aligned orientation angle without strong dependency on the scene environment. The roll angle is extracted accurately from the camera trajectory in our method by using ORB-SLAM [16], which is known to be able to estimate highly reliable camera poses. Furthermore, our approach can handle large roll angles.

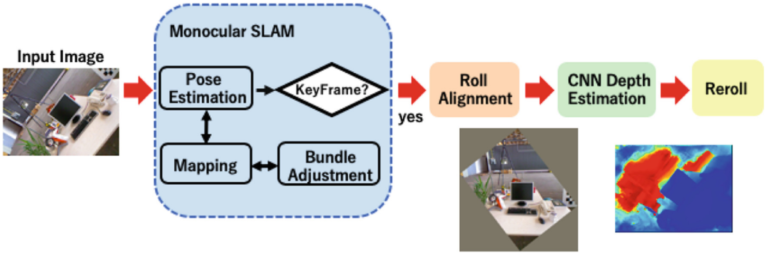

Fig. 2. Our framework overview.

3 Method

The overview of our method is shown in Fig. 2. First, We use monocular SLAM to obtain the camera poses of input RGB images. Second, we calculate transformation function $F(\theta)$ which transform the KeyFrame images against roll direction using SLAM camera poses. This transformation aims to set so that gravity vectors extracted from images are aligned to true gravity direction in the parallel scene. Third, CNN estimates the depth images of the input transformed images. At last, we reroll the CNN outputs in the reverse direction so that the final results have the same in-plane roll angles as the initial input images.

3.1 Camera Pose Estimation

To obtain accurate camera pose act as one of the most important roll in our system. Monocular SLAM or Visual Odometry can estimate precise camera poses based on multi-view geometry, but they suffer from pure rotational movement, including in-plane rotation. To keep tracking accuracy in such a difficult environment, we chose the feature-based RGB version of ORB-SLAM2 [23], which has state of the art accuracy in terms of pose estimation. Compared to other direct approaches [17,18], this system has relatively low computational costs and is easy to combine with.

3.2 CNN Depth Estimation

We employ the same network architecture as CNN-MonoFusion [1]. This network is based on a Resnet50 model [24] following the work of Laina et al. [4]. Atrous convolution and up-projection layers are applied to broaden the field of view and prevent pooling loss. Also, a multi-scale skip-concat is introduced to unit high-level and low-level features. This network adopts *AdaBerhu* loss, which incorporates normalized depth to train the network using various indoor-scene datasets with different focal lengths. To obtain an absolute scale for images taken by the camera, whose intrinsic parameter is different from one at the training time, the output depth is converted to the SLAM scale, in the same manner as CNN-SLAM [11], using the following scaling:

$$D_{test} = \frac{f_{test}}{f_{tr}} D_{CNN}, \tag{1}$$

where D_{CNN} denotes the depth value predicted by the network, f_{test} is the focal length of the camera used for the SLAM, and f_{tr} is a reference focal length used at the training time.

3.3 Roll Alignment

We transformed input RGB images before and after applying CNN part for the same absolute angles, so all CNN inputs are parallel to the scenes. As mentioned in Sect. 1, we assumed that the initial Frame of SLAM will be parallel to the scene (i.e. the gravity direction of the subjects in the image matches the vertical axis in the camera coordinates).

We estimated the transformation $F_t(\theta)$ against the input RGB image I_t to obtain the depth image D_t using the camera pose. We set D_{test} in the previous section as D_t, and t denotes the timestep in the sequence. Note that the transformation $F_t(\theta)$ is an affine transformation.

The current camera pose $\mathbf{T}_t^{cw} \in \mathbb{R}^{4\times4}$ can be estimated using ORB-SLAM2 [23]; the pose is composed of the camera rotation $\mathbf{R}_t^{cw} \in \mathbb{R}^{3\times3}$ and camera translation $\mathbf{s}_t^{cw} \in \mathbb{R}^3$, which are relative to the initial frame. Camera rotation $\mathbf{R}^{cw}$ can be divided into three rotation matrices about xyz axes, as seen in the Eq. 2;

$$\mathbf{R}^{cw} = \mathbf{R}^{cw}(\psi)\mathbf{R}^{cw}(\phi)\mathbf{R}^{cw}(\theta), \tag{2}$$

where ψ is the pitch, ϕ is the yaw, and θ is the roll rotation. Focusing on in-plane rolling, $\mathbf{R}^{cw}(\theta)$ is expressed as a 3×3 matrix, as shown in the Eq. 3;

$$\mathbf{R}^{cw}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

From this angle θ, we can obtain a 2×3 transformation matrix $F_t(\theta)$, as shown in the Eq. 4;

$$F_t(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & s_x \\ \sin\theta & \cos\theta & s_y \end{pmatrix}, \tag{4}$$

where s_x and s_y denotes translation vector to align the center of original image with the center of transformed image.

We applied the $F_t(\theta)$ affine transformation with bilinear interpolation so that CNN input would not lose its original pixels and have blank pixels which have no RGB data as few as possible. To prevent blank pixels from disturbing the CNN calculation, we altered their pixel values to zero in convolutional layer.

After getting CNN outputs, we reroll the depth image in the same manner in the inverse direction to obtain the same resolution depth images as the original KeyFrame RGB images.

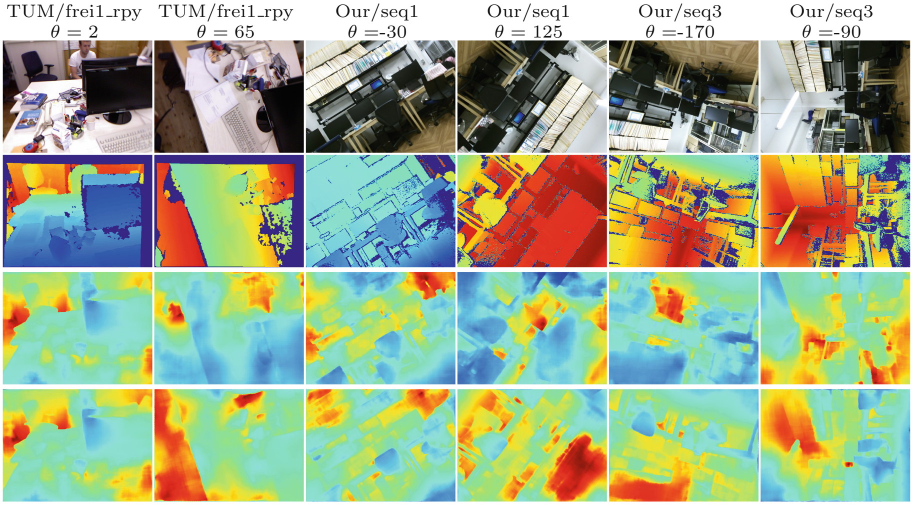

Fig. 3. Qualitative results on TUM RGB-D and our dataset. From top to bottom: the input image, the ground truth depth image, the predicted depth image by the baseline method, and the predicted depth image by our proposed method.

4 Experiment

4.1 Experiment Detail

In the experiments, we compared our proposed method to the baseline method that directly inputs the RGB image to the depth prediction network, which does not apply roll rectification using SLAM. Note that the depth prediction network is the same as the proposed method and the baseline method. The difference is that the proposed method transformed the input image. We evaluated our method from both qualitative and quantitative perspectives. This evaluation was carried out on a desktop PC with an Intel Core i7-7700 CPU at 3.60 GHz and a Nvidia GTX 1080Ti GPU.

For the depth prediction network, we employed the trained model from CNN-MonoFusion [1] which was publicly available[2]. The model is trained with both the NEAIR dataset [1] and the NYU Depth V2 dataset [6]. This network predicts the depth image with a resolution of half size of the input image, so that the depth image is rescaled as the same resolution of the input image.

4.2 Dataset

To evaluate our proposed method, we use three *rpy* sequences of a TUM RGB-D dataset [9], which is widely known and has a pure rotational camera movement captured by a Kinect V1 sensor. *rpy* is an abbreviation that the camera moves

[2] https://github.com/NetEaseAI-CVLab/CNN-MonoFusion.

along in a *roll-pitch-yaw* direction. However, only using this dataset is not sufficient for evaluating our method for the following reasons: (1) this dataset has a lot of ground truth depth images in which several pixels' values are zero because some scenes contain objects over 4.0 m away, which is the max depth range of Kinect V1 and (2) the number of frames at each in-plane roll angle is not uniformly distributed (e.g., the number of frames around 90° is much less than that around 0°).

Therefore, we recorded our own dataset using a Kinect V2 sensor. This dataset is composed of six sequences with in-plane rotation ranging from $-180 < \theta < 180$ with a uniform distribution. Each sequence is around 30 or 80 s, and the overall dataset contains $7,704$ pairs of RGB images and aligned depth images with a resolution of 640×480 pixels. All sequences were recorded in indoor environments and we assumed that the max depth was 4.5 m within which the depth can be obtained accurately.

Table 1. Average errors in our dataset (left) and TUM RGB-D dataset (right)

	Abs_Rel ↓		RMSE ↓	
	Ours	Baseline	Ours	Baseline
seq1	**0.2372**	0.3260	**0.7865**	1.0565
seq2	**0.5161**	0.5929	**0.6557**	0.7321
seq3	**0.2745**	0.3425	**0.9406**	1.2068
seq4	**0.3590**	0.4225	**1.2818**	1.4601
seq5	**0.3687**	0.3930	1.9395	**1.9262**
seq6	**0.2614**	0.3339	**0.8769**	1.1095
ave	**0.3169**	0.3818	**1.0961**	1.2784

	Abs_Rel ↓		RMSE ↓	
	Ours	Baseline	Ours	Baseline
frei1_rpy	2.2482	**2.2380**	**1.0036**	1.0217
frei2_rpy	1.7147	**1.6586**	0.9486	**0.9285**
frei3_rpy	0.9190	**0.8557**	0.9062	**0.8672**
ave	1.6211	**1.5708**	0.9480	**0.9297**

5 Result

5.1 Qualitative Evaluation

Figure 3 shows the qualitative results on TUM RGB-D dataset and our dataset. When comparing our proposed method to the baseline, it is evident that the predicted depth is improved drastically as the structure of the scene can be predicted correctly. In the results of seq1 (in our dataset), our proposed method predicted that the ceiling was farther than the desk. However, the baseline method predicted that the ceiling has almost the same depth as the chair or desk.

In addition, Fig. 8 shows the crucial failure of conventional approach without in-plane roll refinement in a scene from the TUM frei2_rpy sequence. In Fig. 8(b), points of the teddy bear are located in front of the PC and the points of the wall, which should be far away from the desk and the PC, are sticking out. In contrast, Fig. 8(c) estimates that the points of teddy bear are in the back side of

PC and the points of wall are extended in the far side from the camera position like the ground truth model shown in Fig. 8(a).

As we mentioned in Sect. 1, objects which have relatively large depth, such as the ceiling and wall, tend to occupy the upper side, and those which have small depth, such as the floor, occupy the lower side of an image. Therefore, the CNN, which was trained on almost parallel scenes, hallucinates this natural perspective assumption regardless of input's camera pose. Our simple contrivance that incorporates camera poses can directly prevent this harmful characteristic.

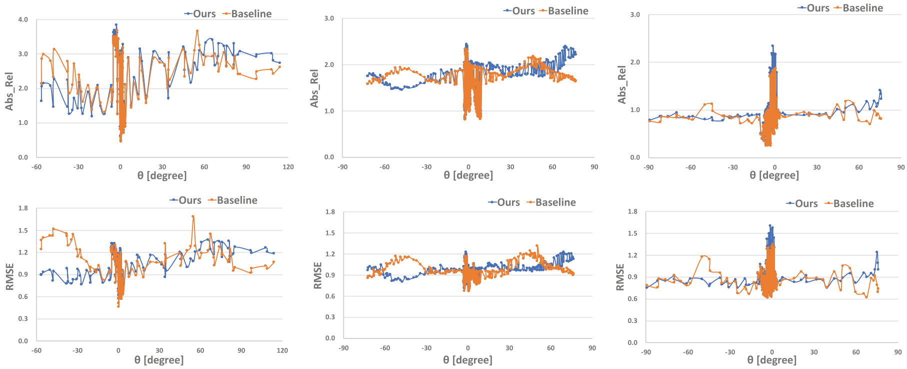

Fig. 4. The correlation between roll angle of frame pose and absolute relative error evaluated in the TUM dataset by Abs_Rel (top) and RMSE (bottom).

5.2 Quantitative Evaluation

We employ absolute relative error (Abs_Rel) and root mean squared error (RMSE) as the evaluation metrics, which means that the lower value is better. Table 1 left shows the results of the self-created dataset. From the Table 1, our proposed method outperformed the baseline method for both the Abs_Rel and the RMSE evaluation metrics.

Figure 5 shows the Abs_Rel of each sequence in our dataset. The horizontal axis shows the estimated roll angle and the vertical axis shows the Abs_Rel value. The errors around $\theta = 0$ do not show any difference between the error of the proposed method vs the baseline method. However, at the range $-45 < \theta < 45$, the baseline method outperformed ours, because the pixels, which are filled with zero disturb the performance slightly, as mentioned in Sect. 3.3. For larger θ, even though the error of the baseline method increased, the error of our proposed method does not depend on θ value. This shows the effectiveness of our proposed method.

Table 1 right shows the results of the TUM RGB-D dataset. In this dataset, our method does not outperform the baseline method. The Abs_Rel and RMSE of each sequence is summarized in Fig. 4. However, from the predicted result of the TUM frei1_rpy in Fig. 3, our proposed method correctly predicts the depth of the floor which is located in the farther side rather than a table. Also, the results in Fig. 7, which shows the average errors by 10° rolling angles in the TUM dataset and our dataset, indicates our methods are not so far behind from the baseline method.

There are two main explanations for the fact that our proposed method qualitatively predicted correct depth but quantitatively did not outperform. First, scenes in frei2_rpy and frei3_rpy have ground truth depth data whose values are 0 in walls and ceiling because of Kinect V1's limitation of depth range. These pixels were not considered in the evaluation, so our proposed method could not show the effectiveness even though our method outperformed qualitatively. Second, and the main reason, is that the scenes in frei2_rpy and frei3_rpy have few KeyFrames in the relatively large angles, such as over ±60°, while around 0°, they have dozens of KeyFrames (Fig. 6).

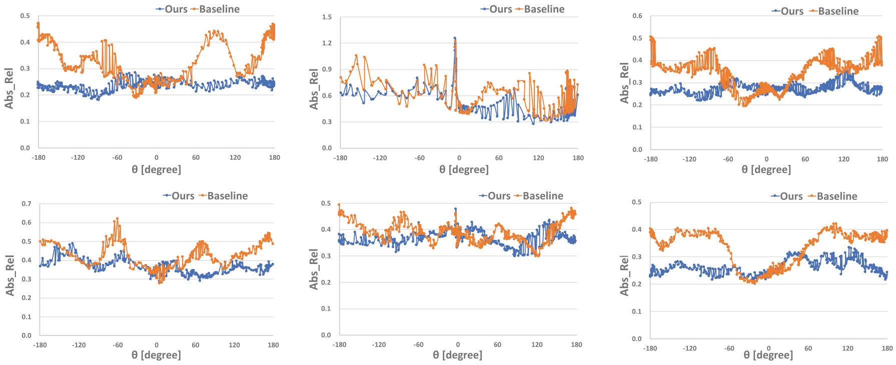

Fig. 5. The correlation between roll angle of frame pose and Abs_Rel evaluated in our dataset.

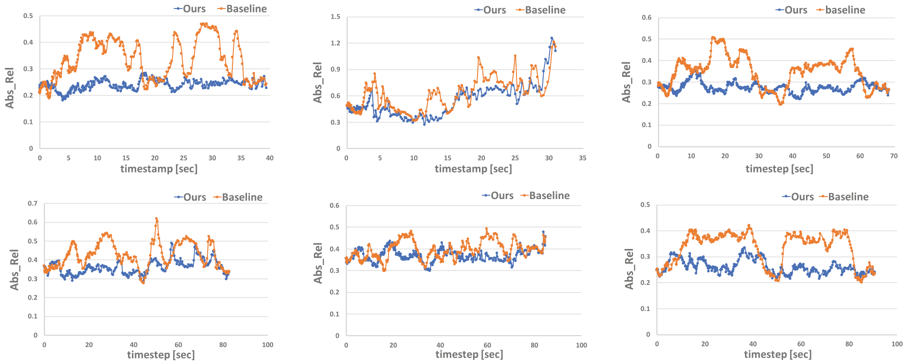

Fig. 6. Time series graph of Abs_Rel evaluated in our six sequences.

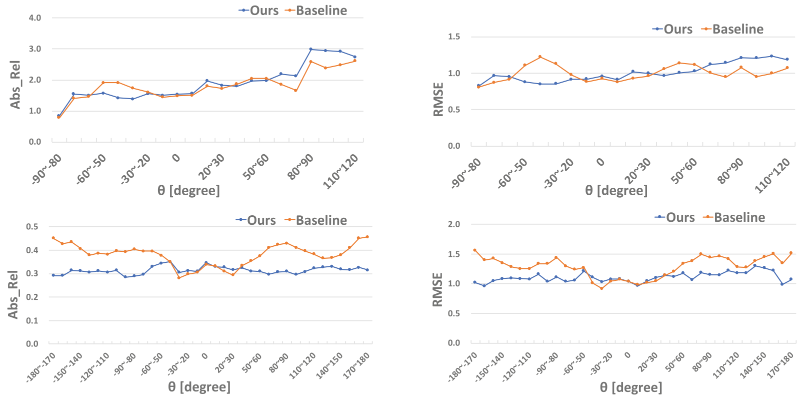

Fig. 7. The correlation between roll angle of frame and Abs_Rel or RMSE over all sequences in TUM and our datasets. Data is divided into bins by 10° and represented as average errors in the bins.

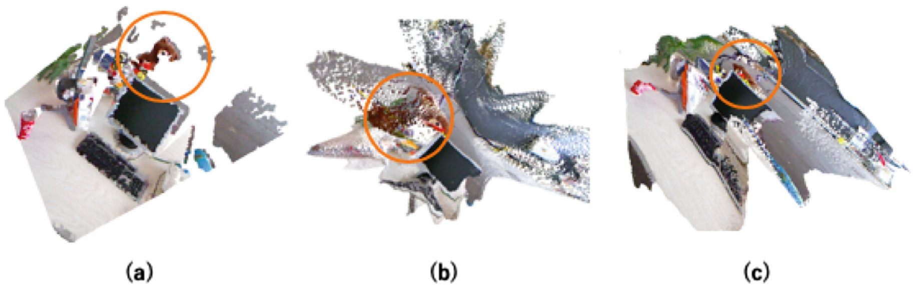

Fig. 8. Result of reprojected points using CNN predicted depth in a scene from TUM frei2_rpy. (a) result by ground truth depth, (b) result by depth of baseline approach, (c) result by depth of our proposed approach. The area circled with an orange marker indicates the location of the teddy bear in the scene. (Color figure online)

6 Conclusion

In this paper, we proposed a simple but effective method to estimate accurate depth for roll-rotated images by using the camera poses directly, which is extracted from the monocular SLAM system. Our method rotates the in-plane rolled image as if it was not roll-rotated before inputting to CNN and re-roll predicted depth image inversely after CNN. Although our approach is simple, we showed the effectiveness of this contrivance by evaluating both qualitatively and quantitatively in public and our own dataset for the in-plane rolling motion of the camera. Our system does not rely on the specific CNN architecture and can run only from RGB images.

We resolved the CNN's drawback that learning-based approach cannot handle the roll-rotated image because of the implicit disposition of learning data and the limitation of data argumentation. In the future work, we are going to get rid of the assumption of a parallel scene in an initial frame and find a more accurate gravity vector in a 2D image. Also, We are going to compare our approach with the trained model with random roll augmentation.

Acknowledgement. This work was partially supported by the Japan Science and Technology Agency (JST) under grant JPMJMI19B2 and JPMJCR1683.

References

1. Wang, J., Liu, H., Cong, L., Xiahou, Z., Wang, L.: CNN-MonoFusion: online monocular dense reconstruction using learned depth from single view. In: IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), pp. 57–62. IEEE, Munich (2018)
2. Wang, Y., Chao, W., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.: Pseudo-LiDAR from visual depth estimation: bridging the gap in 3D object detection for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 8445–8453. IEEE (2019)
3. Marcu, A., Costea, D., Licăreţ, V., Pîrvu, M., Sluşanschi, E., Leordeanu, M.: SafeUAV: learning to estimate depth and safe landing areas for UAVs from synthetic data. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11130, pp. 43–58. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11012-3_4
4. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: International Conference on 3D Vision (3DV), pp. 11–20. IEEE (2016)
5. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2002–2011. IEEE (2018)
6. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54
7. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant CNNs. In: International Conference on 3D Vision (3DV), pp. 11–20. IEEE (2017)
8. Mi, L., Wang, H., Tian, Y., Shavit, N.: Training-free uncertainty estimation for neural networks. arXiv preprint arXiv:1910.04858 (2019)
9. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE International Conference on Intelligent Robot Systems, pp. 573–580. IEEE (2012)
10. Grisettiyz, G., Stachniss, C., Burgard, W.: Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In: IEEE International Conference on Robotics and Automation, pp. 2432–2437. IEEE (2005)
11. Tateno, K., Tombari, F., Laina, I., Navab, N.: CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6243–6252. IEEE (2017)
12. Laidlow, T., Czarnowski, J., Leutenegger, S.: DeepFusion: real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient predictions. In: International Conference on Robotics and Automation, pp. 4068–4074. IEEE (2019)

13. Toyoda, K., Kono, M., Rekimoto, J.: Post-data augmentation to improve deep pose estimation of extreme and wild motions. arXiv preprint arXiv:1902.04250 (2019)
14. Kurz, D., Benhimane, S.: Gravity-aware handheld augmented reality. In: IEEE International Symposium on Mixed and Augmented Reality, pp. 111–120. IEEE (2011)
15. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 225–234. IEEE (2007)
16. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
17. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
18. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2017)
19. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: International Conference on Robotics and Automation, pp. 15–22. IEEE (2014)
20. Fischer, P., Dosovitskiy, A., Brox, T.: Image orientation estimation with convolutional networks. In: Gall, J., Gehler, P., Leibe, B. (eds.) GCPR 2015. LNCS, vol. 9358, pp. 368–378. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24947-6_30
21. Olmschenk, G., Tang, H., Zhu, Z.: Pitch and roll camera orientation from a single 2D image using convolutional neural networks. In: 2017 14th Conference on Computer and Robot Vision, pp. 261–268. IEEE (2015)
22. Xian, W., Li, Z., Fisher, M., Eisenmann, J., Shechtman, E., Snavely, N.: UprightNet: geometry-aware camera orientation estimation from single images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9974–9983. IEEE (2019)
23. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. **33**(5), 1255–1262 (2015)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. IEEE (2016)