Temporal Lidar Depth Completion

Pietari Kaskela
pkaskela@nvidia.com
Philipp Fischer
pfischer@nvidia.com
Timo Roman
troman@nvidia.com

Applied Deep Learning Research
NVIDIA

Abstract

Given the lidar measurements from an autonomous vehicle, we can project the points and generate a sparse depth image. Depth completion aims at increasing the resolution of such a depth image by infilling and interpolating the sparse depth values.

Like most existing approaches, we make use of camera images as guidance in very sparse or occluded regions. In addition, we propose a temporal algorithm that utilizes information from previous timesteps using recurrence. In this work, we show how a state-of-the-art method PENet can be modified to benefit from recurrency. Our algorithm achieves state-of-the-art results on the KITTI depth completion dataset while adding only less than one percent of additional overhead in terms of both neural network parameters and floating point operations. The accuracy is especially improved for faraway objects and regions containing a low amount of lidar depth samples. Even in regions without any ground truth (like sky and rooftops) we observe large improvements which are not captured by the existing evaluation metrics.

1 Introduction

The task of depth completion aims at recovering a dense depth map from a sparse depth map using additional inputs such as camera images as guidance (cf. Figure 2). The task is especially important in the context of autonomous vehicles (AVs), where sparse depth maps are produced by lidar sensors but dense depth maps are required by some employed perception algorithms. For example, the Velodyne HDL-64E lidar sensor used by the popular KITTI [6] dataset fills up only 6% of the depth values of a corresponding color image, when projected onto the image.

In addition to infilling and interpolating the depth values of the remaining 94% pixels, a proper depth completion solution needs to be able to deal with errors caused by the different mounting positions of the camera and lidar sensor, moving objects and the spinning movement of the lidar sensor itself. Figure 2 illustrates the inputs (color image, sparse depth) and the output (dense depth) of the depth completion task. Notice how there are occlusions (image regions with missing points) and overlaps (image regions with points from different depths) in the image, since the lidar and the camera have slightly different viewpoints.

Most state-of-the-art depth completion approaches rely on a U-Net [21] style backbone followed by a CSPN-based [2] refinement network [8, 14, 19]. For the closely related *depth*

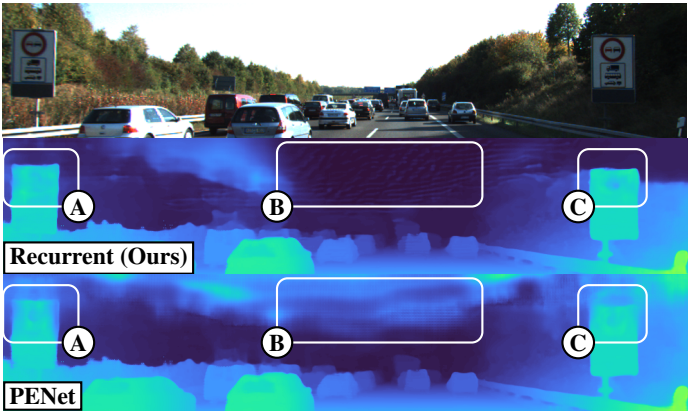

Figure 1: Our recurrent model achieves a new state-of-the-art result on the KITTI depth completion validation set. In addition, our model significantly improves upon regions which are not captured by the sparse ground truth or lidar input, as illustrated in regions (A), (B) and (C).

estimation task, in which only camera images are available, temporal techniques have been extensively studied [13, 20, 27, 31]. Prior work on temporal techniques in the depth completion setting can be found in [4, 9, 20], but these methods have not shown competitive accuracy with state-of-the-art non-temporal approaches.

In this paper, we propose a depth completion architecture to utilize temporal information for more accurate depth completion. Given any existing U-Net + refinement type approach, such as PENet [8], only minor modifications to the the number of input and output channels are required to implement our proposed temporal processing. We can further significantly improve the qualitative and quantitative results by utilizing pose information between timesteps to align the previous depth prediction with the next frame.

Applying these improvements to the open-source PENet architecture allows us to achieve state-of-the-art results on the KITTI depth completion validation set, with a neglible increase in both model parameters and required floating point operations. In addition, our model significantly improves the depth completion accuracy in regions not captured by the sparse depth ground truth, as shown in Figure 1. In summary, our contributions are as follows:

(i) We present general modifications to PENet [8] - a popular depth completion model - that allow the network to utilize temporal information. These modifications can be applied to other similar architectures.

(ii) We present an effective way of using pose information to warp the previous depth prediction to align with the current timestep and thereby significantly improve the accuracy.

(iii) Our depth completion solution achieves state-of-the-art accuracy on the KITTI depth completion dataset with a negligible increase in both neural network parameters and floating point operations required.

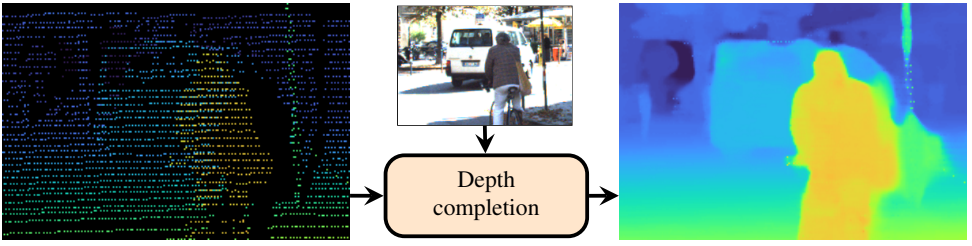

Figure 2: A cropped color image (middle), with the corresponding projected sparse lidar sample (left) and a depth predicted by a depth completion model (right). Notice how the projected sparse lidar samples for the cyclist and the van overlap because of the different viewpoints of the sensors. The camera is mounted to the right of the lidar sensor.

2 Related Work

We rely on the KITTI depth completion dataset by Uhrig et al. [25], a popular dataset for depth completion in the AV setting, which is an extension to the KITTI raw dataset [6]. The raw KITTI dataset contains driving sessions recorded with an extensive sensor suite, of which relevant to this work are the RGB cameras, the lidar sensor, the global positioning system (GPS) sensor and the inertial measurement unit (IMU) sensor. In addition, the KITTI depth completion dataset contains 93k sparse depth maps and the corresponding generated semi-dense ground truth depth maps, raw point clouds and RGB images. Uhrig et al. [25] also present a sparse convolutional neural network architecture for depth completion based on the sparse lidar depth input only.

Modern approaches focus on image-guided depth completion, in which the sparse depth is projected to the viewpoint of a corresponding color image. Most state-of-the-art solutions [8, 14, 19] rely on a U-Net-like backbone network to extract a coarse depth estimate and fused color-depth features, and then feed both to a CSPN-based [2] refinement network. Recently, transformer-based solutions [32] have also reached competitive accuracies in the KITTI depth completion challenge.

Previous work [4, 9, 20] has shown that temporal information can be used to improve depth completion solutions, but those have not achieved better accuracy compared to state-of-the-art non-temporal models. Patil et al. [20] built a joint framework for depth estimation and depth completion based on the ConvLSTM architecture [24]. Huang [9] explored integrating ConvGRU [1] and three-dimensional convolutional layers into existing depth completion architectures. Eskandar et al. [4] aim to not only complete sparse depth inputs using variational recurrent neural networks, but to also predict future dense depth maps.

Temporal approaches are much more popular in the image-only depth estimation setting, where the ill-posed nature of estimating 3-D depth from 2-D images alone is alleviated by exploiting the temporal correlations. Structure from Motion (SfM) [26] techniques have been studied for decades. Effective ways of utilizing the temporal information in modern approaches include implementing a recurrent architecture such as in [13, 20, 27, 31] or utilizing the sequences to calculate consistency losses between frames [7, 27, 33].

We chose to build upon the PENet [8] model, as it is open-source and shares a similar structure with many other state-of-the-art depth completion solutions. The architecture consists of a backbone with two U-Nets and a dilated and accelerated convolutional spatial propagation network (DA-CSPN++) refinement network. The backbone takes as input the

sparse lidar depth and the RGB image. The outputs of the backbone are a coarse depth estimate and internal features from the U-Nets. The coarse depth is then refined by the refinement module using an iterative procedure that propagates the nearby depth values based on the internal U-Net features and the sparse lidar depth.

3 Method

To add recurrence to the PENet [8] architecture, we chose to introduce additional input channels for the previous depth and hidden history as well as the corresponding output channel for the hidden history. If pose information between timesteps is available, we propose utilizing it to project the previous depth prediction to match with the current timestep. A general diagram of the whole model is illustrated in Figure 3.

3.1 Architecture

Since consecutive depth maps have a large correlation, we chose to base the recurrence of the model on the dense depth prediction of the previous timestep. We modify the PENet architecture consisting of a double U-Net backbone and a DA-CSPN++ refinement network such that the output depth of the DA-CSPN++ module is concatenated as an input channel to both U-Nets in the next timestep. This will allow the network to accumulate depth samples over time and to estimate depth in regions which contain a low amount of depth samples much better. Of course, the previous predicted dense depth is rather inaccurate in many scenarios, such as in settings with lots of moving objects or when the vehicle itself is moving fast. Solutions for alleviating the problems related to the movement of the vehicle itself are discussed in Section 3.2.

To allow the network to more freely learn to pass information between timesteps, a single-channel hidden history is also added to the architecture. This is implemented as an additional output channel from the last layer of the second U-Net and concatenated to the input of both of the U-Nets in the next timestep. This way, the hidden history contains latent information from the deeper and spatially smaller layers, but can also pass high-resolution information from earlier layers. The values of the hidden history are clamped between -1 and 1 to stabilize training.

3.2 Warping

Naively feeding the previous dense depth prediction as input to the next timestep is not optimal, due to the misalignment caused by the movement of the vehicle itself and moving objects in the environment. It is possible for the network to learn to correct this misalignment, but we found that effective utilization of previous depth values requires correcting as much of the misalignment as possible before the neural network. The problem of predicting the three-dimensional movement of dynamic objects in the environment is called scene flow estimation, but integrating such capability is out of the scope of this paper. However, the misalignment between timesteps caused by the movement of the vehicle itself can be corrected when the egomotion of the vehicle is available (i.e. its relative translation and rotation between the timesteps).

We propose applying classic reprojection to the predicted depth to transform the depth to the viewpoint of the vehicle in the next timestep. The warping requires the relative pose

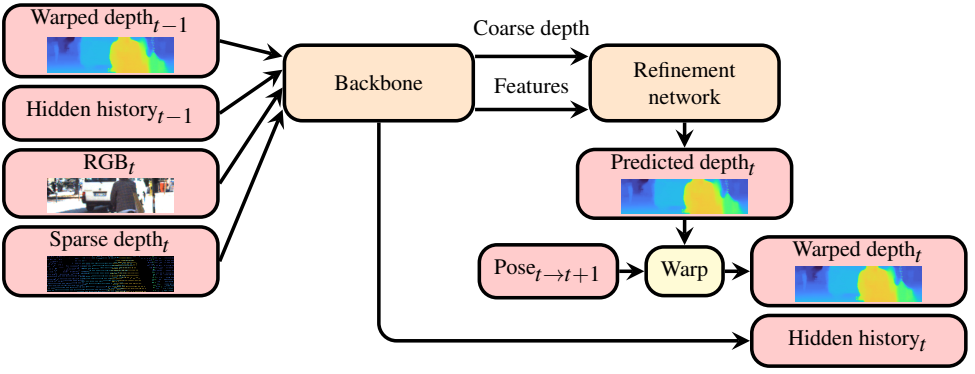

Figure 3: A general diagram of our recurrent depth completion model. The warped depth and hidden history from the previous timestep $t-1$ are fed as input to the backbone in the current timestep t.

matrix $P_{(t-1)\to t}$ describing the translation and rotation between the timesteps and the camera intrinsics matrix K used for projecting the point-cloud to the image-plane and back. Possible overlaps caused by the reprojection are solved by choosing the minimum depth value at each pixel. Warping can be implemented in an efficient and differentiable way by utilizing the scatter-function available in modern deep learning frameworks. This work considered warping the previous depth only. One possible extension to this would be to study the impact of warping the hidden history in addition. This is left for further study.

For the KITTI depth completion dataset, we calculate the pose matrices from the GPS and IMU sensor data using the approach detailed in the original publication [6]. The pose matrices can also be estimated using visual odometry [5] solutions such as COLMAP [22].

3.3 Implementation details

The models were trained on a machine with either 8 NVIDIA A100 GPUs, each with 40GB memory or 8 NVIDIA V100 GPUs, each with 32GB memory. The models were trained for up to 60 epochs using a cosine decay learning rate scheduler [15] with a linear warmup of 2 epochs and a batch size of 4. The base learning rate was 10^{-3}. All of the training runs used the AdamW optimizer [16] with a weight decay of 10^{-6} and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

We train on the KITTI depth completion training set, which contains 86k images and provide results on the KITTI depth completion validation set which contains 4k images. The temporal models are trained on uniformly sampled sequences of 32 contiguous frames, while the validation uses sequences of length 128. For training, sequences with less than 32 frames are dropped, but validation is done on all frames of the validation set.

The training images are first bottom-center cropped to 352 by 1216 pixels to remove most of the sky, which does not contain any lidar samples, and then randomly cropped to the final training size of 192 by 608 pixels. The augmentations used during training include flipping the image and lidar inputs horizontally for 40% of the sequences, color-jitter augmentation and dropping uniformly random-sized rectangles from the sparse lidar depth for 15% of the frames. The image input is scaled to the interval [0, 1] and the depth input and ground truth values are divided by 100. The loss is the mean square error calculated against the valid

ground truth pixels.

The networks are trained using truncated backpropagation through time (TBPTT) [28], which is parametrized by the weight update interval k_1 and the backpropagation length k_2. In Section 4 we provide an ablation on these parameters and show that temporally-aware training is required for optimal accuracy.

Our baseline non-temporal model and training procedure is derived from PENet [8]. In addition to the previously mentioned changes related to training, we also removed the Batch Normalization [11] operations that were applied after the mask and kernel confidence generating convolutions in the DA-CSPN++ refinement network.

4 Experiments

In this section we study the effects of the proposed architectural changes detailed in Section 3 and compare our results to current state-of-the-art depth completion solutions. Following the KITTI depth completion benchmark, we report the following four metrics that compare the predicted depth $\hat{D}$ and the ground truth depth D: root mean square error (RMSE), mean absolute error (MAE), root mean square error of the inverse depth (iRMSE) and mean absolute error of the inverse depth (iMAE).

4.1 Ablation study

Table 1 compares our proposed architectural changes to a non-temporal baseline derived from PENet. We trained the baseline four times to gauge the variance between training runs. In the *Previous Depth* -section of the table we introduce recurrence based on the unmodified depth output from the previous timestep and observe an absolute improvement of about 10 in the RMSE metric. The rather little gain is expected as the unwarped depth provides a useful guide to the network only in very limited static scenarios.

Configuration	RMSE ↓ (mm)	MAE ↓ (mm)	iRMSE ↓ (1/km)	iMAE ↓ (1/km)
Baseline	773.9±3.2	218.0±0.8	2.34±0.04	0.97±0.01
Previous Depth				
+ TBPTT(1, 1)	762.4 (-11.5)	215.1	2.24	0.94
+ TBPTT(2, 2)	772.0 (-1.9)	216.9	2.29	0.96
+ TBPTT(1, 2) + Hidden Hist.	758.5 (-15.4)	214.2	2.23	0.95
Warped Previous Depth				
+ TBPTT(1, 1)	728.7 (-45.2)	204.9	2.20	0.94
+ TBPTT(2, 2)	737.4 (-36.5)	209.3	2.31	0.97
+ TBPTT(1, 2) + Hidden Hist.	**720.8 (-53.1)**	203.5	2.25	0.94

Table 1: Metrics for the full KITTI depth completion validation set for the recurrent experiments. The metrics are reported from the checkpoint of the network with the lowest RMSE metric. For the baseline experiment, mean and variance from four training runs are reported.

In the *Warped Previous Depth* -section of Table 1 we introduce warping of the output depth between the timesteps, which results in significant improvement in all metrics. Combining the warped depth experiment with a single-channel hidden history and a temporally-

aware training algorithm allows us to reach a RMSE of 720.8 on the full KITTI depth completion validation set.

As previously mentioned, we use truncated backpropagation through time as a training algorithm to explore the tradeoffs between the temporal capabilities of the model, the final accuracy and training time. The training algorithm TBPTT(k_1,k_2) is parametrized by the weight update interval k_1 and the backpropagation length k_2. For faster training, we experimented on increasing the weight update interval parameter k_1 of TBPTT from one to two, but the value of one resulted in higher accuracy consistently in all of our experiments. To induce more temporal utilization at the cost of training time, we tried different values of the backpropagation length k_2. Backpropagation length of two is necessary for the network to be able to utilize the hidden history and based on our experiments also sufficient for realizing most of the temporal benefits.

4.2 Comparison to state-of-the-art

In Table 2 we compare our results to other state-of-the-art solutions on the KITTI depth completion validation set. On the KITTI depth completion validation set our results significantly improve upon the previous published state-of-the-art solution SemAttNet [19]. Note that we are unable to provide results on the test set for our model as the test set is not published in a temporal format which is needed by our method. Unfortunately, the test set does not consist of consecutive frames.

The baseline PENet (containing a double U-Net backbone and the DA-CSPN++ refinement network) has about 130 million parameters and 404 GFLOPs. While significantly improving the results, our best recurrent model increases the parameter count by less than a hundredth of a percent and the FLOPs by 0.35%. The overhead is small, since we only change the outermost layers of the neural networks. The additional depth warping overhead is negligible as well.

Model	RMSE ↓ (mm)	MAE ↓ (mm)	iRMSE ↓ (1/km)	iMAE ↓ (1/km)
PENet [8]	757.2	209.0	2.22	0.92
RigNet [30]	752.1	205.2	3.22	0.93
DySPN [14]	739.4	**191.4**	-	-
SemAttNet [19]	738.1	204.5	2.01	0.89
Recurrent (Ours)	**722.2**	204.0	2.30	0.96

Table 2: Metrics for the KITTI depth completion selected validation set with 1000 frames for several state-of-the-art methods. Our recurrent model outperforms all of the published models on the KITTI depth completion validation set. DySPN is the only method that trains on the L1-loss and thus has a significantly lower MAE.

No additional training data was used by us except for the poses provided by the original KITTI dataset, compared to the other methods in Table 2. Since the existing training set contains sequences, our recurrent method can use their order. This information (poses and frame sequences) is also available in a real-world automotive setting, which we are targeting.

4.3 Analysis

As the semi-dense ground truth depth of the KITTI depth completion dataset still fills only 16% of the pixels of the corresponding color image, standard metrics such as RMSE and MAE fail to capture differences between models in regions that contain a low amount of ground truth samples.

Improvements in such regions are illustrated in Figure 4, where the increased detail is especially clear on street signs, buildings and trees. These regions contain few sparse depth lidar samples in the current timestep and thus the predicted depth is much more detailed when the model is allowed to warp and accumulate depth samples temporally.

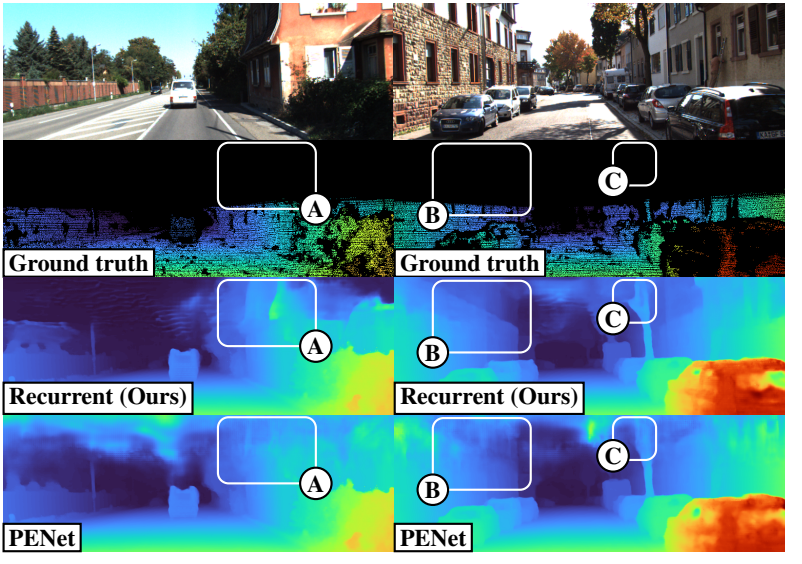

Figure 4: RGB color input and ground truth (top) with recurrent model output (middle) compared to PENet model output (bottom). Notice how the additional temporal information helps especially in regions (A), (B) and (C), where the current timestep sparse lidar input has very few samples.

Figure 5 illustrates the average difference of errors for 8 by 8 pixel blocks calculated over the KITTI depth completion validation set. The top image illustrates the pixels where our recurrent model has on average lower error than PENet and vice versa for the bottom image. Notice how our recurrent model is more accurate on mid- and long-range depth completion and in regions containing less lidar samples such as the upper parts of the image.

Region (A) of Figure 5 contains an interesting phenomenon caused by left-hand traffic, as the vehicles moving to the opposite direction are difficult for both models to estimate. In this scenario, the warped depth is incorrect as warping cannot account for the movement of the other vehicles. We further analyzed this region by calculating the validation metrics only in this region and only for those pixels that are segmented as vehicles by a segmentation network based on [12]. While the accuracy improvement on vehicles moving to the opposite direction is not as substantial as the average improvement (approx. 50 RMSE), it is still noteworthy at approximately 25 RMSE.

Notice that the improvement in accuracy of our model near the edges of the Figure 5 is below the average improvement. This is especially visible in region (B) where the errors

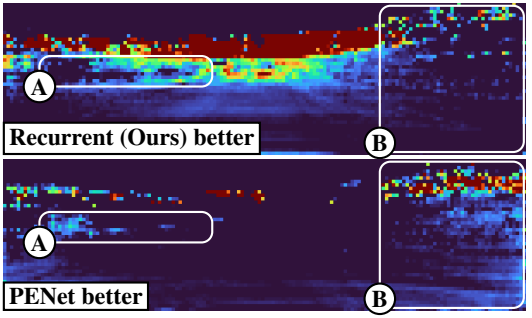

Figure 5: Difference of average errors for 8 by 8 pixel blocks calculated over the KITTI depth completion validation set. Illustrated using the Turbo colormap [17], where red denotes high values and dark blue low values. Our recurrent model is generally better at mid- to long-range prediction, while matching the accuracy of PENet at short ranges. Regions (A) and (B) highlight the limitations of solutions based on warping, caused by moving objects and small compounding errors.

in the intrinsics matrix K and the errors in the measurements between the lidar sensor and right camera are compounded. The warping procedure is especially sensitive to errors in the intrinsics and pose matrices when warping depth samples near the edges, as even small errors can change the location of a warped depth sample by several pixels. Cvišić et al. [3] have proven that the default calibration parameters of the KITTI dataset are not optimal and that better calibration parameters can significantly increase the accuracy of visual odometry algorithms.

The error of our recurrent model decreases quickly during the first frames of the accumulation. While in the first frame our method is on average 100 RMSE worse than the PENet baseline, the error is already 50 RMSE better starting from the third frame. Figure 6 illustrates the average RMSE per frame compared to PENet. Note that in a real-world setting, the first frames can most likely be discarded. For the evaluation, we include all frames (also the first ones).

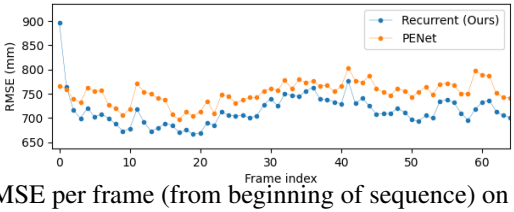

Figure 6: Average RMSE per frame (from beginning of sequence) on the KITTI depth completion validation set, reported with our recurrent model and the PENet.

5 Conclusion

Generating a dense depth map (100% image coverage) from a sparse depth map with only 6% coverage seems like a very difficult task. However, by utilizing additional information such as the RGB camera image, the task becomes feasible as previous work has shown.

In this work, we demonstrated a natural extension to this idea: By using a full sequence

of consecutive frames with both sparse lidar measurements and camera images as an input to a recurrent neural network, we are able to surpass the current state of the art results on the KITTI validation set. To this end, we evaluated different options in an ablation study and found that warping the previous predicted depth as an input to the next iteration works best.

In regions without ground truth, we expect the improvement to be even larger. However, this effect is not easily measurable. To verify this assumption, one could manually generate a small test set with dense depth by fitting 3D models into the scene. Another approach would be synthetic data generation.

For the offline use case, we believe accuracy could be further improved by making use of not only past but also future frames. This could be implemented using bidirectional recurrent neural networks [23] or via models that explicitly use multiple frames [11]. Another possible future line of work could be to investigate the impact of utilizing more than one former depth frame, as such information would be especially helpful in handling disoccluded regions.

Furthermore, we found our approach to benefit from accurate car ego motion (poses), so combining this approach with leading approaches from the KITTI Visual Odometry challenge [5], might improve the results further. Integrating a learned approach for pose estimation would also robustify our model against errors originating from the GPS and IMU sensors and allow us to apply our model to indoor depth completion datasets [18, 29].

Acknowledgments We thank Jaakko Lehtinen for supervising this project and Pekka Jänis for discussion and reviewing early drafts.

References

[1] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations, 2016.

[2] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network, 2018.

[3] Igor Cvišić, Ivan Marković, and Ivan Petrović. Recalibrating the kitti dataset camera setup for improved odometry accuracy. 2021. doi: 10.48550/ARXIV.2109.03462. URL https://arxiv.org/abs/2109.03462.

[4] George Eskandar, Alexander Braun, Martin Meinke, Karim Armanious, and Bin Yang. Slpc: a vrnn-based approach for stochastic lidar prediction and completion in autonomous driving, 2021.

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[7] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019.

[8] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Towards precise and efficient image guided depth completion. 2021.

[9] Xinan Huang. Exploring the effectiveness of naive spatio-temporal exploits for depth completion, 2020.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[11] Pietari Kaskela. Temporal Depth Completion for Autonomous Vehicle Lidar Depth Sensing. Master's thesis, Aalto University. School of Science, 2023. URL http://urn.fi/URN:NBN:fi:aalto-202303262586.

[12] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks, 2019. URL https://arxiv.org/abs/1901.02446.

[13] Aran C.S. Kumar, Suchendra M. Bhandarkar, and Mukta Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 396–3968, 2018. doi: 10.1109/CVPRW.2018.00066.

[14] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion, 2022.

[15] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[17] Anton Mikhailov. Turbo, an improved rainbow colormap for visualization, 2019. URL https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html.

[18] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[19] Danish Nazir, Alain Pagani, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Semattnet: Towards attention-based semantic aware guided depth completion. *IEEE Access*, pages 1–1, 2022. doi: 10.1109/ACCESS.2022.3214316.

[20] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video, 2020.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[22] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093.

[24] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.

[25] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns, 2017.

[26] S. Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 203(1153):405–426, 1979. ISSN 00804649. URL http://www.jstor.org/stable/77505.

[27] Rui Wang, Stephen M. Pizer, and Jan-Michael Frahm. Recurrent neural network for (un-)supervised learning of monocular videovisual odometry and depth, 2019.

[28] Ronald J. Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2(4):490–501, 1990. doi: 10.1162/neco.1990.2.4.490.

[29] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5 (2):1899–1906, 2020.

[30] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Baobei Xu, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. *CoRR*, abs/2107.13802, 2021. URL https://arxiv.org/abs/2107.13802.

[31] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation, 2019.

[32] Youmin Zhang, Matteo Poggi, Zheng Zhu, Guan Huang, Stefano Mattoccia, et al. Completionformer: Depth completion with convolutions and vision transformers.

[33] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video, 2017.