Real-Time Temporally Consistent Depth Completion for VR-Teleoperated Robots

Chengfan Li[1] Automne Petitjean[2] Are Oelsner[1] Stefanie Tellex[1] James Tompkin[1]
[1]Brown University [2]ENS de Lyon

Abstract

VR-teleoperated robots provide immersive 3D experiences essential for executing complex remote tasks. However, rendering effects and 3D reconstruction quality significantly impact the precise robot-environment interaction. Due to hardware limitations, the depth data collected by the robot are challenged by sparsity, noise, and instability. To address this, we propose a real-time rendering and reconstruction system that enhances visual perception and interaction precision through dense depth perception and temporally consistent rendering in dynamic scenes. In particular, (1) We employ sensor-specific data augmentation that aligns with the robot's sensor input perceptual characteristics. (2) We achieve real-time depth completion using an algebraically-constrained, normalized CNN to propagate depth and confidence through multi-scale multi-modal fusion network regulated by a gradient matching loss. (3) A spatial-temporal geometry-aware filter is implemented to ensure temporally consistent point cloud reconstruction. Rendered in Unity, our system reconstructs 3D point clouds from robot camera feeds and uses consumer-grade VR hardware to remotely control a Boston Dynamics Spot robot. To our best knowledge, this is the first system for VR-teleoperated robots that concurrently addresses data sparsity and dynamic scene stability, achieving real-time rendering speeds of 40 FPS. Our code, demo videos and trained models are available at the project page.

1. Introduction

Robot teleoperation, i.e., the remote control of a robot by a human operator, is essential for performing dangerous and complex robotic tasks, especially in fields such as industrial automation [1], deep space exploration [31], and nuclear facility maintenance [12]. Unlike complex and expensive fully autonomous systems, robot teleoperation based on human-robot collaboration provides a more flexible and reliable solution that can enhance operational accuracy while retaining the flexibility of human decision-making [26]. The rapid development of Virtual reality (VR) has provided a breakthrough in robot teleoperation [7, 35, 39]. Unlike traditional 2D interfaces, the immersive 3D experience provided by VR can greatly enhance the operator's spatial perception, thereby improving the user experience and reducing cognitive load [36].

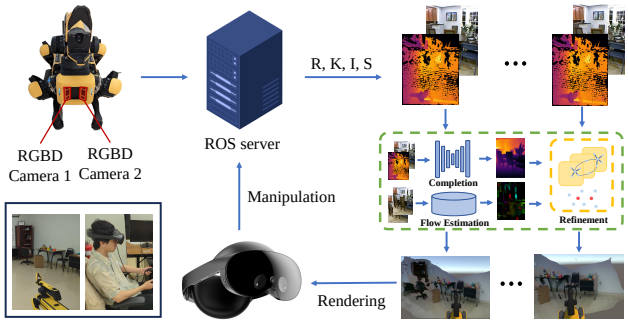

Figure 1. **Overview.** The Spot robot streams RGB images and sparse depth data to the ROS server. We first complete depth maps and estimate optical flow. Then the pose and optical flow are used to refine completed depth maps for temporal consistency. Finally, the reconstructed temporally consistent point clouds are rendered to VR through Unity, enabling real-time VR teleoperation.

Despite significant progress in VR-teleoperated robots, 3D scene reconstruction and rendering still face huge challenges due to the limitations of robot hardware and the complexity of constructing immersive environments in VR. This results in low reconstruction quality and a poor user experience.

Due to the hardware limitations of robots, the depth data from the robot sensors for 3D reconstruction faces problems such as sparsity, noise, and instability [22, 25]. Although offline depth completion [37, 38, 42, 49] has made significant progress, models on various datasets continue to increase the complexity of the architecture in order to improve accuracy, sacrificing inference speed and real-time performance. Therefore, although these models have high accuracy and good generalization capabilities, they cannot meet the real-time reconstruction and 3D scene rendering of VR-teleoperated robots. On the other hand, although some models can provide real-time depth completion [9, 16, 19, 40], a common challenge faced by these models is their lack of generalization ability. Specifically, they perform poorly under certain perceptual characteristics, especially when faced with noisy and specially distributed robot depth data. As a

result, their depth completion is not always reliable.

Since the robot and its surrounding environment are dynamic, maintaining the consistency and stability of the point cloud in both spatiality and temporality between frames is critical for high-quality visual rendering results and seamless transition of 3D-reconstructed scenes. Most existing methods focus on consistent video depth estimation which render a consistent and dense 3D point cloud directly from RGB video [13, 15, 18, 21]. However, since there is no sparse point cloud as input, they cannot generate metric depth. Recurrent architectures have also made great progress in enforcing temporal consistency [6, 28, 46]. However, due to the quadratic scaling of the sequence input for the image and the complexity of the architecture itself, their inference speed is slow, which limits their application scenarios.

To address these challenges, our paper introduces a real-time depth completion and point cloud reconstruction system for VR-teleoperated robots. Compared to previous methods, our pipeline enables temporally consistent reconstruction and rendering in real-time. The system adapts Spot's perceptual characteristics by applying a Spot-collected sensor mask to generate training input for the model. It achieves accurate depth completion using a lightweight RGB-guided multi-scale multi-modal fusion network. It also ensures temporally consistent point cloud reconstruction between frames by using a spatial-temporal geometric awareness filter based on the pretrained NeuFlow-v2 optical flow estimation model [47]. As shown in Fig. 1, the Quest Pro consumer-grade VR hardware and the Spot mobile robot from Boston Dynamics [8] teleoperated by the GHOST [2] also provide an ideal testing platform for our system.

2. Related Work

Depth Completion LiDAR [10] and SfM [32] generate sparse depth data, prompting extensive research on depth completion. Traditional approaches, such as IP-Basic [16], combine image processing techniques to interpolate and refine the sparse depth maps. And in recent years, we have witnessed a rapid progresses in learning-based approaches.

Spatial propagation models use local information to complete depth. Nconv [9] conducts normalized-CNNs to jointly propagate confidence and depth. CSPN [4] utilizes fixed kernel sizes for simultaneous pixel updates, while CSPN++ [5] enhances this by integrating adaptive kernel sizes. DSPN [22] and NLSPN [27] predict non-local neighborhood by deformable convolutions and affinities. TVPD [44] propagate affinitive neighbor in TPV spaces. BPNet [37] incorporates content-dependent bilateral filter at the earliest stage. These propagation-based methods are suitable to be used as preprocessing for sparse depth [9, 37] or as refinement mechanism [5, 27, 44]. Building on the unguided Nconv [9], the first stage of our network propagates sparse depth to construct a light-weight architecture and accelerate convergence.

And in the fusion stage, the common way is to directly concatenate the depth map and RGB image [24]. RigNet [41] propose repetitive hourglass network to extract and fuse features. RigNet++ [43] further apply semantic information to guide the fusion. Deeplidar [29] integrates surface normal and ACMNet [48] uses symmetric gated fusion strategy. FseNet [3] applies continuous convolution on 3D points to fuse 3D geometry. Our approach utilize multi-scale depth features from the previous resolution layer.

Although many models [37, 38, 49] are highly accurate, these complex models cannot be used for real-time 3D scene reconstruction and VR teleoperated robots due to their slow inference speed. Some models [19, 40] with real-time inference speeds, however, face the problem of poor generalization ability. We propose the sensor-specific data augmentation to train lightweight but high-accurate models specifically for robot sensors.

Consistent Video Depth Maintaining the consistency and stability of the reconstructed scene is critical for VR-based systems. Optical flow are widely used to ensure consistency, both in refining video [17], and in generating video depth estimation [15, 20, 23]. Li *et al.* [20] introduces optical flow-based alignment and occlusion handling to enforce temporally consistency. CVD [23] take advantage of optical flow and structure-from-motion. RobustCVD [15] apply geometry-aware depth filtering and spatially-varying depth deformation model to refine the estimated depth video. Recurrence models such as convolutional LSTMs [6, 28, 46] have also made great progress due to their ability to capture spatial and temporal dependencies simultaneously, but they have a more complex architecture and have longer inference time. Shao *et al.* [33] reformulate the task into a conditional generation problem and use Diffusion model to generate consistent video depth. Khan *et al.* [14] maintain a dynamicly-updated global point cloud to encourages consistency. Due to the noisy and unstable nature of the robot sensor and our need for real-time application, we denoise and refine the completed video depth using spaial-temporal geometry-aware filter based on real-time optical flow estimation [47].

3. Method

3.1. Overview

As shown in Fig. 1, the Spot robot streams RGB images and sparse depth data to the ROS server. Unity is then used to render the temporally consistent point cloud reconstructed by the pipeline shown in Fig. 2 into VR in real-time. Using our GHOST [2] system, the user can seamlessly navigate

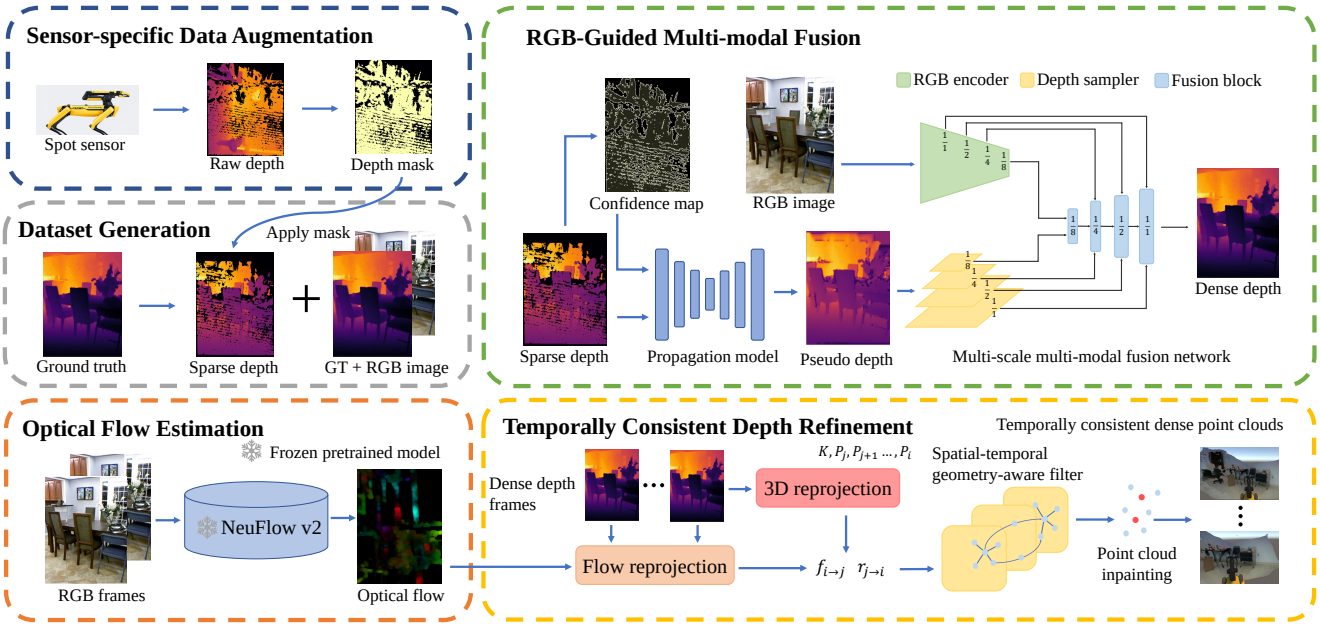

Figure 2. **Real-time consistent depth completion pipeline.** (a) Sensor-specific data augmentation applies robot-collected sensor masks to generate training inputs. (b) A lightweight, algebraically-constrained CNN performs efficient depth and confidence propagation via multi-scale RGB-guided fusion. (c) Optical flow between frames are generated from pretrained real-time optical flow estimation model. (d) A spatial-temporal geometry-aware filter ensures consistent point cloud reconstruction, enhancing dynamic scene rendering quality.

and manipulate the robot within a 3D virtual environment.

As shown in Fig. 2, an RGB-guided multi-modal fusion network trained by sensor-specific data augmentation is used to generate dense depth $\tilde{D}$ from sparse depth S and RGB image I for each frame. A temporally consistent point cloud is then reconstructed from frames of completed depth based on optical flow estimation and the geometry-aware filter.

3.2. VR-teleoperation

Our GHOST [2] system collects data from Spot's sensors for reconstruction pipeline and provides a flexible navigation and manipulation of a Boston Dynamics Spot robot. For data collection, it uses a ROS-based pipeline to compresses depth data to overcome bandwidth limitations, as well as utilizes time synchronization to ensure data alignment. For robot control, the user can drive the Spot and control the robotic arm from flexible perspective in VR and get real-time feedback.

3.3. Sensor-Specific Data Augmentation

When there are mismatches between training data and the specific perceptual characteristics of robot sensors, the model's performance will degrade. Since the sparse depth maps in public datasets are not collected by the Spot sensor, and it's hard to obtain the dense depth from the Spot sensor directly to build our own dataset, we first collect a set of sparse depth maps $\mathcal{M}$ from our Spot as shown in the upper left of Fig. 2. Then for a ground truth dense depth map D, we select a Spot-collected sensor mask from $\mathcal{M}$ to generate the sparse training input S for our model as:

$$S = D \circ \mathbb{I}(m > 0.0), \quad m \sim \text{Uniform}(\mathcal{M}). \tag{1}$$

Together with the RGB image I corresponding to D in the original data set, (S, D, I) forms a pair of training data. Our experiments in Section 4 demonstrate that this sensor-specific augmentation significantly improves depth completion accuracy and visual quality on the Spot's sensor.

3.4. RGB-guided Multi-modal Fusion

Unguided normalized CNNs Building on the architecture presented in Nconv [9], the algebraically-constrained normalized convolution layers propagate depth $\tilde{S}^{l-1}$ and confidence C^{l-1} for the $(l-1)$th resolution layer to the lth layer as:

$$C_p^l = \frac{\sum_{q \in \mathcal{N}(p)} C_q^{l-1} \Gamma(W_q^l)}{\sum_{q \in \mathcal{N}(p)} \Gamma(W_q^l) + \epsilon}, \tag{2}$$

$$\tilde{S}_p^l = \frac{\sum_{q \in \mathcal{N}(p)} \tilde{S}_q^{l-1} C_q^{l-1} \Gamma(W_q^l)}{\sum_{q \in \mathcal{N}(p)} \tilde{S}_q^{l-1} \Gamma(W_q^l) + \epsilon} + b^l, \tag{3}$$

where $\mathcal{N}(p)$ refers to neighborhood centered around the pixel p, W is the weights generated by our model, Γ is the

activation function, ϵ is use to prevent divide-by-zero problems and b is the bias. Moreover, the initial value of C and $\tilde{S}$ can be writtern as:

$$C^0 = \mathbb{I}(m > 0.0), \tag{4}$$

$$\tilde{S}^0 = S, \tag{5}$$

where S is the sparse depth input.

By generating pseudo depth maps $\tilde{S}$ prior to multi-modal fusion, the propagation model addresses the challenges posed by sparse data, improving fusion quality and accelerating the convergence of the fusion network. Moreover, the number of parameters required for the unguided normalized CNN layer is less compared to other methods, thus ensuring the efficiency of the network and meeting the real-time requirement of the teleoperated-robot.

Multi-scale multi-modal fusion network After propagating sparse data to generate pseudo depth maps, we incorporate RGB data from the Spot's camera to enhance edge, corner and texture information. We adopted a general structure and fusion strategy similar to some existing architectures [29, 37, 40]. As shown in Fig. 2, an encoder consisting of a simple residual block [11] extracts features of multiple scales from the RGB image. These RGB features $\phi(I^\ell)$ are then concatenated with corresponding downsampled pseudo depth maps $\tilde{S}^\ell$ for the ℓth resolution layer using an early fusion strategy and fed into the multi-modal fusion decoder. We adopt a multi-scale approach, where each intermediate fusion layer receives $\tilde{S}^\ell$ and $\phi(I^\ell)$ of current resolution, and additionally obtains the depth result $\tilde{S}^{\ell-1}$ of the previous layer, thereby integrating coarse-to-fine information. This fusion process results in the final high-resolution estimated depth map $\tilde{D}$, improving depth completion accuracy.

3.5. Temporally Consistent Depth Refinement

The robot and its surrounding environment are dynamic, leading to noisy and unstable point clouds. Frame-by-frame completion and reconstruction of unstable depth data will produce jittery scene in VR rendering. As shown in Fig. 2, a spatial-temporal geometric awareness filter is implemented to ensure consistent reconstruction.

Flow Reprojection Our refinement pipeline begins with the utilization of the NeuFlow-v2[47] model to estimate optical flow between frame i and frame j. For RGB images I_i and I_j, the flow reprojection from frame i to frame j can be obtained as:

$$f_{i\to j}(p) = \mathbb{V}_{j\to i}(\mathbb{F}_{NeuFlow}(I_i, I_j), p), \tag{6}$$

where $\mathbb{F}_{NeuFlow}$ represents the pre-trained NeuFlow-v2[47] model that return a flow vector from I_i to I_j and $\mathbb{V}_{j\to i}$ is the map from 2D depth index p in the frame j to the 2D depth index of the corresponding point in frame i base on optical flow.

3D Reprojection Then the flow reprojection can be used to project the corresponding point of p in frame i to the camera coordinate of frame j:

$$T_{i\to j}(p) = R_j^\top (R_i K^{-1} \tilde{D}_i \circ f_{i\to j}(p) + t_i - t_j), \tag{7}$$

where R is the rotation matrix, K is the camera intrinsic matrix, t is the translation vector and $\tilde{D}$ is the completed depth map. R and t can be directly calculated from the pose matrix P based on

$$P_i = [R_i|t_i]. \tag{8}$$

To complement the 2D motion information from optical flow, the built-in kinematics module in ROS is used to obtain real-time pose estimations.

Temporally Consistent Point Cloud Denoising We implement a spatial-temporal geometry-aware filter similar to CVD [23] and RobustCVD [15] as the point cloud refinement to generate consistent point clouds. To denoise and refine the point cloud, we calculate the reprojection weights of p and its neighbors $\mathcal{N}(p)$ for each geometrically corresponding pixel in the past n frames. And use 3D reprojection of them to calculate the refined camera coordinate position $\tilde{c}_i(p)$ for frame i:

$$\tilde{c}_i(p) = \sum_{q\in\mathcal{N}(p)} \sum_{j=i-n}^{i} T_{j\to i}(q) w_{rp}(c_i(p), T_{j\to i}(q)), \tag{9}$$

where w_{rp} is the reprojection weight. Previously, CVD [23] and RobustCVD [15] used disparity loss and ratio loss, respectively, to calculate the reprojection weight. However, they only consider the spatial weight and ignore the temporal weight. In our method, we take advantage of spatial-temporal information to formulate our reprojection weight:

$$w_{rp}(v_i, v_j) = w_s(v_i, v_j) w_t(i-j), \tag{10}$$

where the spatial weight w_s is the ratio weight with custom parameter σ_s which can be written as:

$$w_s(u, v) = \exp\left(-\sigma_s\left(\frac{\max(u_z, v_z)}{\min(u_z, v_z)} - 1\right)\right), \tag{11}$$

the reason to use ratio weight is because of the unstable pose estimation and the bias of the disparity loss. And the new temporal weight can be written as:

$$w_t(\tilde{t}) = \exp\left(-\frac{(\eta\tilde{t})^2}{2\sigma_t^2}\right), \tag{12}$$

where $\tilde{t}$ is frame index difference, η is time between two frames and σ_t is custom parameter.

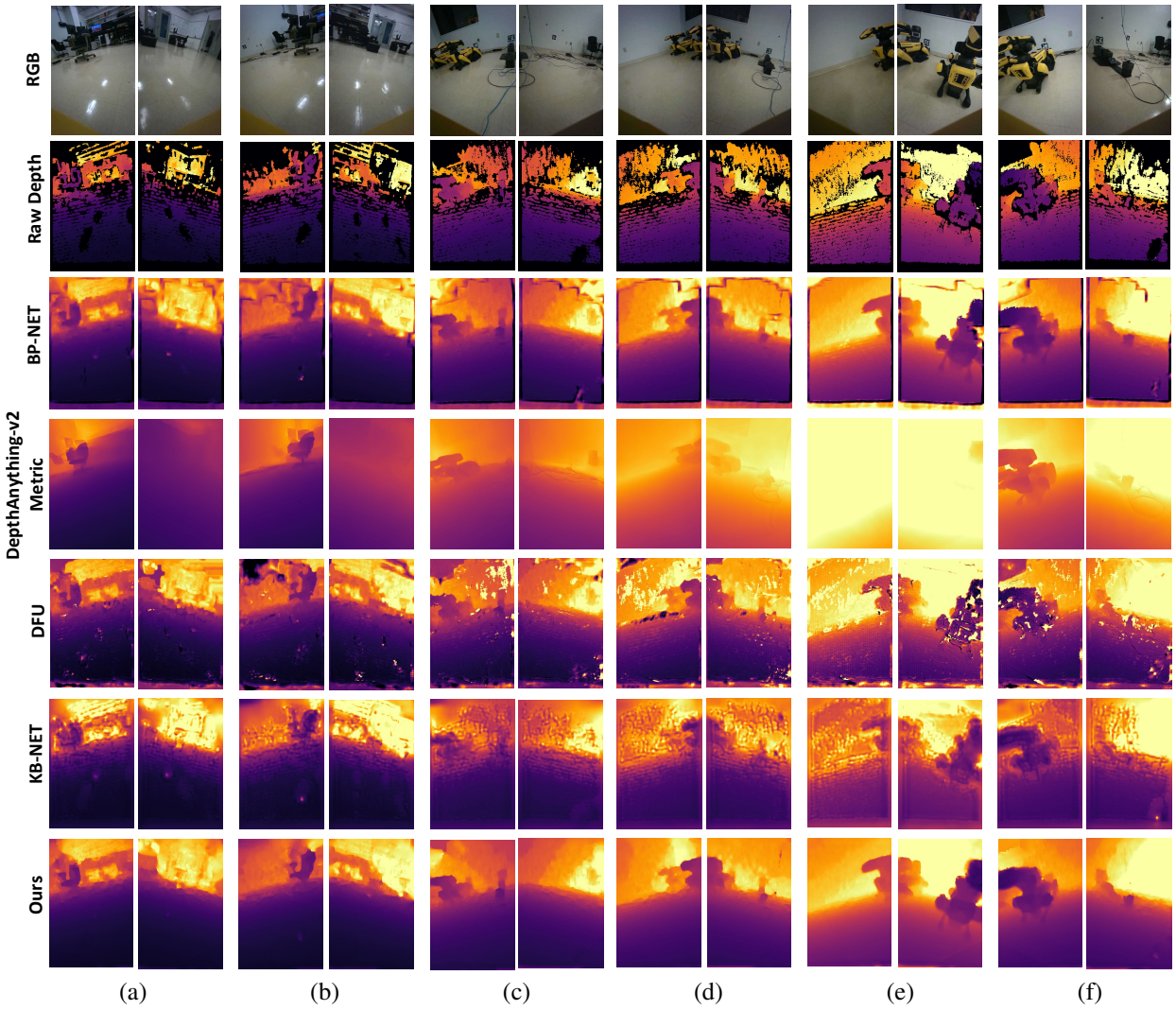

Figure 3. **Qualitative Comparison** with BP-NET [37], DepthAnything-v2-Metric [45], DFU [38] and KB-NET [40] on our data collected from the Spot robot. Our method is demonstrated in the last row.

3.6. Implementation Details

Loss Function We use two loss terms for each scale to train our model. The multi-scale loss is used to ensure that depth maps generated from each scale are fully supervised. The loss function can be written as:

$$\mathcal{L}_{train} = \sum_{s=0}^{S} \lambda_s(\alpha\mathcal{L}_{gd} + (1-\alpha)\mathcal{L}_{l1}), \tag{13}$$

where $\mathcal{L}_{gd}$ is the gradient matching loss , $\mathcal{L}_{l1}$ is the $l1$ reconstruction loss, S is the total number of scales, λ_s is the weight for each scale and α is customer parameter.

The gradient matching loss $\mathcal{L}_{gd}$ is proposed by MiDaS [30] that will tend to produce the same discontinuous edges as the ground truth, thus ensuring sharpness and discontinuities at the edges of the complemented depth map. The gradient matching loss can be written as:

$$\mathcal{L}_{gd} = \frac{1}{N}\sum_{i=1}^{N}\left(|\nabla_x H_i| + |\nabla_y H_i|\right), \tag{14}$$

where

$$H_i = D_i - \tilde{D}_i. \tag{15}$$

And we define the $l1$ reconstruction loss as

$$\mathcal{L}_{l1} = \frac{1}{N}\sum_{i=1}^{N}|H_i|. \tag{16}$$

4. Experiments

4.1. Experimental setup

Datasets We evaluate the effectiveness of our proposed method on two customized datasets. The first dataset was

Figure 4. **Temporally Consistent Point Cloud Denoising** The top row shows the final rendering results of the Spot moving through the dynamic scene, the middle row shows the result after frame averaging, and the last row shows the result after applying temporally consistent point cloud denoising. We can observe that the geometry of the chair is preserved when the Spot is moving.

used to qualitatively evaluate and compare different models. This dataset contains 3000 sets of consecutive frames captured from Boston Dynamics Spot Robot's [8] two front sensors, each frame contains the sparse depth map and the corresponding RGB image. Since the data collected on Spot does not contain ground truth depth, we only use these data for qualitative evaluation to compare the performance of different models on real robots.

To quantitatively evaluate the performance of the models. We collected some sparse depth maps from Spot and used sensor-specific data augmentation introduced in Section 3.3 to preprocess the NYUv2 [34] dataset to generate our second dataset for quantitative evaluation. The NYUv2 [34] dataset was chosen because the depth data distribution in it is similar to the environment of our Spot robot, and the use of sensor-specific data augmentation to simulate the effects of robotic sensor acquisitions aligns the perceptual characteristic between the two datasets.

Evaluation Metrics We adopt the following evaluation metrics in our quantitative evaluation dataset: root mean squared error (RMSE), mean absolute error (MAE), inverse reciprocal mean squared error (iRMSE), inverse reciprocal mean absolute error (iMAE), gradient matching error (GME), relative error (REL) and percent error ($\delta_{1.25}$).

4.2. Depth Completion

We first evaluate our depth completion model on our quantitative evaluation dataset. Tab. 1 lists the results of our model, compared with other state-of-the-art model in depth completion and depth estimation. Note that although DepthAnything-v2 [45] is a depth estimation model, here we use its metric depth estimation version for our experiments. Overall, our model secures the top position across all seven evaluation metrics among all the methods in our test dataset.

Fig. 3 shows the qualitative comparisons with other methods on real Spot data. Compared to other models, since we have preprocessed the training set of our model using sensor-specific data augmentation, our model is more adapted to this specific perceptual characteristics of the Spot sensor and thus can produce higher quality results on real robot data. Specifically, compared to other depth completion methods, our model is more suitable for filling inhomogeneous depth gaps. Whereas other models produce discontinuous depth results in the middle of large gaps. Compared to depth estimation models, though, depth estimation models produce smoother depth maps and higher quality details. However, they do not estimate depth very accurately and will scale the scene, which we will continue to explore in Sec. 4.3.

4.3. Rendered Point Cloud

Since the VR operator is consistently immersed in a 3D environment, it is clear that the quantitative and qualita-

	RMSE↓(mm)	MAE↓(mm)	iRMSE↓(1/km)	iMAE↓(1/km)	GME↓	REL↓	$\delta_{1.25}$ ↑(%)
BP-Net [37]	190.444	137.998	81.315	46.685	0.251	0.054	92.148
KB-Net [40]	378.377	372.879	87.512	62.124	0.802	0.127	80.958
DFU [38]	171.746	201.614	81.612	36.383	0.573	0.084	91.914
DepthAnything-v2(Metric) [45]	484.033	615.561	101.050	79.988	0.227	0.228	67.291
Ours	**119.816**	**69.730**	**31.558**	**10.178**	**0.160**	**0.024**	**97.548**

Table 1. **Quantitative Comparison Results** We examine the quantitative performance of five models on our second dataset. The performance is measured by RMSE↓(mm), MAE↓(mm), iRMSE↓(1/km), iMAE↓(1/km), GME↓, REL↓ and $\delta_{1.25}$ ↑(%).

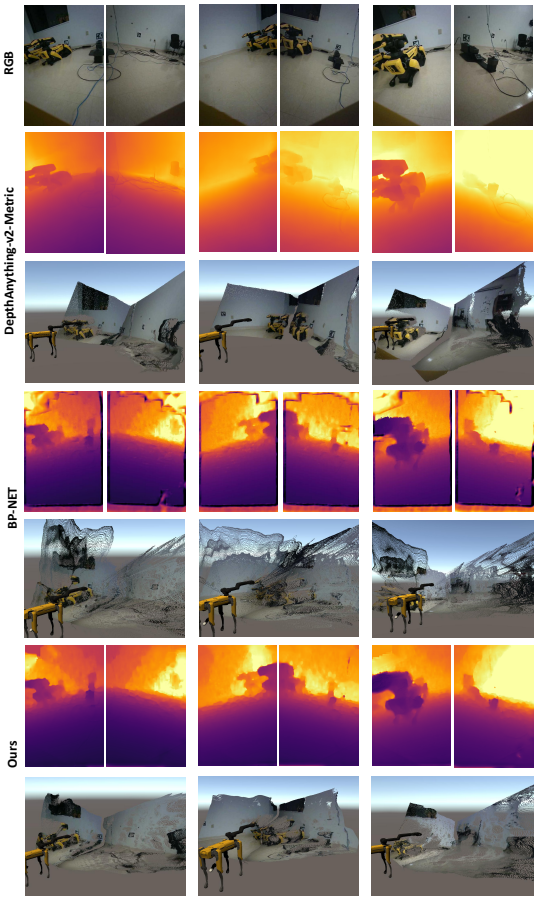

Figure 5. **Qualitative Comparison of Reconstruction Results** The top row shows the input RGB images, followed by depth maps and 3D reconstructions from DepthAnything-v2-Metric [45], BP-NET [37], and Ours.

tive evaluations of the depth map alone do not provide the most realistic representation of the robot operator's visual experience. To better evaluate the user's visual experience, we reconstructed and rendered several keyframes from BP-NET [37], DepthAnything-v2 [45], and our own model within Unity, as demonstrated in Fig. 5.

Although the depth maps produced by DepthAnything-v2[45] appear smooth and rich in detail, the corresponding VR renderings suffer from poor realism. The model preserves the original RGB details but misestimates relative object distances and applies incorrect global scaling, resulting in an unnatural scene. Although BP-NET [37] restores overall depth structures, its reconstruction results contain a lot of noise, ultimately degrading the user's visual experience and the accuracy of the VR-teleoperation. In contrast, Ours methods generate the most accurate and visually consistent results, with sharper object boundaries, smoother depth maps, and high-fidelity 3D reconstruction. These improvements highlight the robustness of our method in achieving superior reconstruction quality.

4.4. Temporally Consistency

To demonstrate the effectiveness of our temporally consistent point cloud denoising approach, we use a VR to move Spot through a dynamic scene, rendering and presenting a series of keyframes. As shown in Fig. 5, our method improves noisy and unstable reconstructed scenes. Moreover, our approach produces temporally consistent scene reconstructions and preserves the geometry of objects in the scene. Thus high-quality visual renderings and seamless transitions are generated in real-time for dynamic scenes.

5. Conclusions

In this paper, we presented a real-time system for depth completion and temporally consistent 3D reconstruction tailored for VR-teleoperated robots. By introducing sensor-specific data augmentation, a lightweight multi-scale multi-modal fusion network, and a spatial-temporal geometry-aware filter, our approach effectively addresses the challenges of hardware limitation, challenged environment and domain mismatching. Experimental results demonstrated the superiority of our method over state-of-the-art approaches in both quantitative and qualitative evaluations, achieving high-quality, temporally stable point clouds rendered in real-time. These advancements significantly improve immersive VR experiences and enhance precision in robot-environment interactions.

References

[1] Doris Aschenbrenner, Michael Fritscher, Felix Sittner, Markus Krauß, and Klaus Schilling. Teleoperation of an industrial robot in an active production line. *IFAC-PapersOnLine*, 48(10):159–164, 2015. 1

[2] Calvin Bauer, Janeth Meraz, Are Oelsner, James Tompkin, and Stefanie Tellex. Ghost in the robot: Virtual reality teleoperation for mobile manipulation. Unpublished manuscript, 2024. 2, 3

[3] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10023–10032, 2019. 2

[4] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2361–2379, 2019. 2

[5] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI conference on artificial intelligence*, pages 10615–10622, 2020. 2

[6] Arun CS Kumar, Suchendra M Bhandarkar, and Mukta Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 283–291, 2018. 2

[7] Morteza Dianatfar, Jyrki Latokartano, and Minna Lanz. Review on existing vr/ar solutions in human–robot collaboration. *Procedia CIRP*, 97:407–411, 2021. 1

[8] Boston Dynamics. Spot® - the agile mobile robot, 2016. Accessed: 2024-12-06. 2, 6

[9] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2423–2436, 2019. 1, 2, 3

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[12] Chetan Kapoor and Delbert Tesar. Integrated teleoperation and automation for nuclear facility cleanup. *Industrial Robot: An International Journal*, 33(6):469–484, 2006. 1

[13] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014. 2

[14] Numair Khan, Eric Penner, Douglas Lanman, and Lei Xiao. Temporally consistent online depth estimation using point-based fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9119–9129, 2023. 2

[15] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 2, 4

[16] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 16–22. IEEE, 2018. 1, 2

[17] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 2

[18] Manuel Lang, Oliver Wang, Tunc Aydin, Aljoscha Smolic, and Markus Gross. Practical temporal consistency for image-based graphics applications. *ACM Transactions on Graphics (ToG)*, 31(4):1–8, 2012. 2

[19] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, Chong Zhang, et al. A multi-scale guided cascade hourglass network for depth completion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 32–40, 2020. 1, 2

[20] Siyuan Li, Yue Luo, Ye Zhu, Xun Zhao, Yu Li, and Ying Shan. Enforcing temporal consistency in video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1145–1154, 2021. 2

[21] Zhaoshuo Li, Wei Ye, Dilin Wang, Francis X Creighton, Russell H Taylor, Ganesh Venkatesh, and Mathias Unberath. Temporally consistent online depth estimation in dynamic scenes. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3018–3027, 2023. 2

[22] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *Proceedings of the aaai conference on artificial intelligence*, pages 1638–1646, 2022. 1, 2

[23] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 2, 4

[24] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018. 2

[25] Fangchang Ma, Luca Carlone, Ulas Ayaz, and Sertac Karaman. Sparse depth sensing for resource-constrained robots. *The International Journal of Robotics Research*, 38(8):935–980, 2019. 1

[26] Curtis W Nielsen, Michael A Goodrich, and Robert W Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics*, 23(5):927–941, 2007. 1

[27] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 120–136. Springer, 2020. 2

[28] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 5(4):6813–6820, 2020. 2

[29] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3313–3322, 2019. 2, 4

[30] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 5

[31] George R Schmidt, Geoffrey A Landis, and Steven R Oleson. Human exploration using real-time robotic operations (herro): A space exploration strategy for the 21st century. *Acta Astronautica*, 80:105–113, 2012. 1

[32] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2

[33] Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors. *arXiv preprint arXiv:2406.01493*, 2024. 2

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 6

[35] Patrick Stotko, Stefan Krumpen, Max Schwarz, Christian Lenz, Sven Behnke, Reinhard Klein, and Michael Weinmann. A vr system for immersive teleoperation and live exploration with a mobile robot. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3630–3637. IEEE, 2019. 1

[36] YH Su, YQ Xu, SL Cheng, CH Ko, and Kuu-Young Young. Development of an effective 3d vr-based manipulation system for industrial robot manipulators. In *2019 12th Asian Control Conference (ASCC)*, pages 1–6. IEEE, 2019. 1

[37] Jie Tang, Fei-Peng Tian, Boshi An, Jian Li, and Ping Tan. Bilateral propagation network for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9763–9772, 2024. 1, 2, 4, 5, 6, 7

[38] Yufei Wang, Ge Zhang, Shaoqian Wang, Bo Li, Qi Liu, Le Hui, and Yuchao Dai. Improving depth completion via depth feature upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21104–21113, 2024. 1, 2, 5, 6

[39] David Whitney, Eric Rosen, Elizabeth Phillips, George Konidaris, and Stefanie Tellex. Comparing robot grasping teleoperation across desktop and virtual reality with ros reality. In *Robotics Research: The 18th International Symposium ISRR*, pages 335–350. Springer, 2019. 1

[40] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12747–12756, 2021. 1, 2, 4, 5, 6

[41] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. *arXiv preprint arXiv:2005.00559*, 2020. 2

[42] Zhiqiang Yan, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet++: efficient repetitive image guided network for depth completion. *arXiv preprint arXiv:2309.00655*, 2023. 1

[43] Zhiqiang Yan, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet++: efficient repetitive image guided network for depth completion. *arXiv preprint arXiv:2309.00655*, 2023. 2

[44] Zhiqiang Yan, Yuankai Lin, Kun Wang, Yupeng Zheng, Yufei Wang, Zhenyu Zhang, Jun Li, and Jian Yang. Tri-perspective view decomposition for geometry-aware depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4874–4884, 2024. 2

[45] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 5, 6, 7

[46] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1725–1734, 2019. 2

[47] Zhiyong Zhang, Aniket Gupta, Huaizu Jiang, and Hanumant Singh. Neuflow v2: High-efficiency optical flow estimation on edge devices. *arXiv preprint arXiv:2408.10161*, 2024. 2, 4

[48] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 30: 5264–5276, 2021. 2

[49] Wending Zhou, Xu Yan, Yinghong Liao, Yuankai Lin, Jin Huang, Gangming Zhao, Shuguang Cui, and Zhen Li. Bev@ dc: Bird's-eye view assisted training for depth completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9233–9242, 2023. 1, 2