MC^2: Multi-view Consistent Depth Estimation via Coordinated Image-based Neural Rendering

Subin Kim*,1 Seong Hyeon Park*,1 Sihyun Yu[1] Kihyuk Sohn[2] Jinwoo Shin[1]
[1]KAIST [2]Google Research
{subin-kim, seonghyp, sihyun.yu, jinwoos}@kaist.ac.kr, kihyuks@google.com

Abstract

We are interested in achieving spatially accurate and temporally consistent depth estimates only from a stream of 2D RGB images. Despite the success of recent depth estimation methods, we find that this is still difficult since existing approaches often estimate depth only from 2D information and overlook how the scene exists in 3D space. To tackle the issue, we propose Multi-view Consistent depth estimation via Coordinated image-based neural rendering *(MC^2) which casts the depth estimation as a feature matching problem in 3D space, thereby constructing and aligning scene features directly in 3D space from 2D images. First, we introduce a rescaling technique that minimizes the ambiguity of the depth estimation obtained independently from each 2D image. Using 2D images and corresponding rescaled depths, we extract the context representation with our new transformer architecture consisting of three-way factorized attention. Moreover, to ensure alignment with 3D structures without explicit geometry modeling, we propose an ordinal volume rendering that respects the nature of 3D spaces. We perform extensive comparisons on casually captured scenes from various real-world datasets and significantly outperform previous work in depth estimation from a stream of 2D RGB images. Results highlight our method as a comprehensive framework that not only improves the accuracy of monocular estimates but also bridges the gap to multi-view consistent depth estimation that respects the 3D worlds existing in given images.*[1]

1. Introduction

Understanding the complex 3D geometry of the real world by observing a stream of 2D views is a core part of human intelligence, enabling complex cognitive abilities to predict and interact with the environment. To tackle this challenge, many recent works have focused on the translation of 2D RGB pixels to pixel-wise depth map, *i.e.*, depth estimation [5, 7, 32, 34, 35], even showing a quite accurate generalization performance for an unseen single image. However, these methods cast this problem as a per-pixel regression [12] or classification [3, 17, 19] on each 2D image, thus, they often show temporally inconsistent estimates across a stream of 2D images. While there have been efforts to mitigate the inconsistencies using additional supervision like optical flows [25, 46, 50], the direct adjustment between inconsistent estimates by imposing over-smooth transitions often neglects fine spatial details and becomes inaccurate. More critically, these approaches focus on the estimation of depth maps only at given RGB images, thus they cannot infer depths of unseen views.

In this paper, we focus on achieving both spatial and temporal consistency in depth estimation by mitigating the aforementioned shortcomings. The main idea is to *directly render a depth in 3D space* by incorporating recent image-based neural rendering techniques [18, 20]. This is different from conventional approaches that estimate or optimize depths from 2D-pixel spaces [5, 25, 46] or model explicit geometry [27, 33, 47]. This distinction allows our framework to more accurately comprehend 3D scenes, as our approach maintains spatial awareness (*e.g.*, recognizes revisited areas within the scene by performing synthesis in 3D space), thus fully exploiting the advantage of multi-view images. Specifically, we synthesize a depth map and corresponding RGB image for the target view by interpolating the pixel values from a given context set of input RGB images.

To synthesize a consistent and accurate depth map in 3D from a stream of 2D images, it is crucial to precisely match correspondences between given multi-view images. We conceptualize this problem as the task of constructing contextual feature sets from 2D inputs and establishing correspondences between these features in 3D spaces. Given the challenge of decompressing the intricate spatial relationships encoded in 2D imagery from constructed context feature sets, we propose *Multi-view Consistent depth estimation via Coordinated image-based neural rendering*, coined MC^2, consisting of three components (see Figure 1):

*Equal contribution.
[1]Visualizations are available at the website : https://subin-kim-cv.github.io/MC2/.

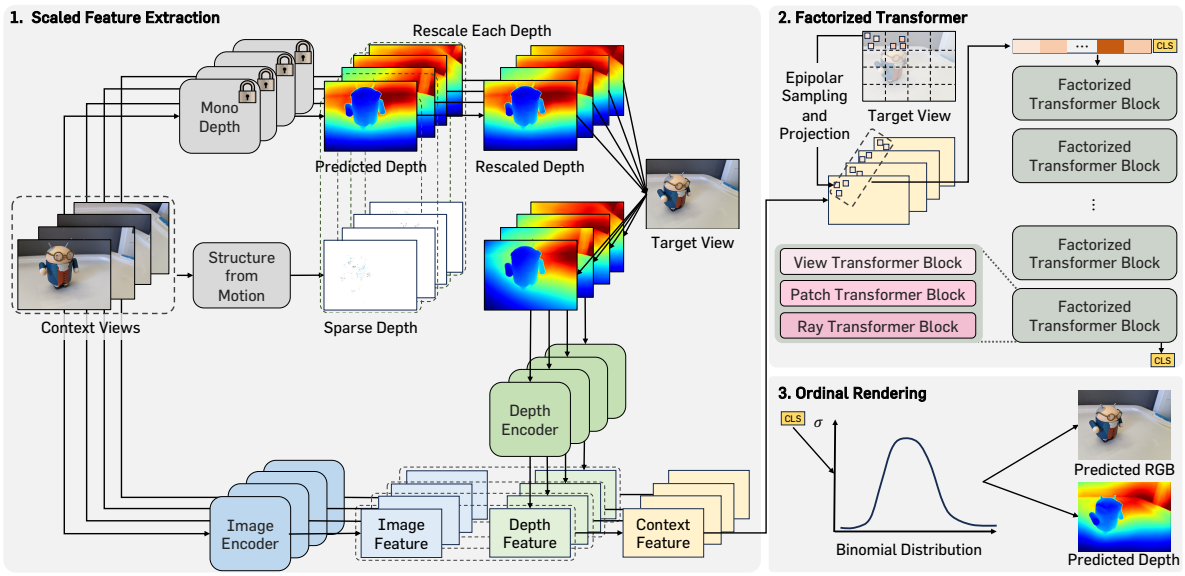

Figure 1. **Overview**. Overall illustration of our proposed method, MC^2. MC^2 synthesizes a depth map and corresponding RGB image at *arbitrary camera angles* using a stream of 2D RGB images as context features.

- **Scaled feature extraction**: MC^2 augments contextual feature sets with rescaled monocular depth estimates alongside the image features from 2D RGB images, thereby improving depth perception. Here, the insight for rescaling is twofold: 1) Real-world scenarios often present a discrepancy between monocular depth estimates and reconstructed 3D scenes where the camera pose for each image resides (*i.e.*, scale ambiguity). 2) Depth values are inherently relative, varying significantly across different camera views for the same object (Section 2.1).
- **Factorized attention for context features**: To efficiently compute the correspondence between the constructed features, we introduce a new transformer architecture [44] that consists of three different attentions. Specifically, each attention is computed with factorized inputs across three different dimensions: view, ray and pixel. By doing so, our architecture can efficiently handle long sequences of constructed features but still captures the correspondence well (Section 2.2).
- **Ordinal rendering**: Finally, we introduce ordinal rendering which focuses on the ordering of features in 3D space and respects the sequential nature of a ray in 3D. This new rendering scheme effectively synthesizes depths and provides flexibility even without relying on explicit geometry modeling, thus effectively handling even casually captured extensive scenes observed over long trajectories (Section 2.3).

We conduct comprehensive experiments to verify the effectiveness of MC^2 on various casually captured, in-the-wild datasets. Specifically, we evaluate our method on 1) the scenes reconstructed using the SfM pipeline and 2) the scenes reconstructed using visual odometry sensor data to validate the effectiveness across various scenarios where the 3D world exists. We demonstrate the strong performance of MC^2 where it significantly outperforms the prior depth estimation approaches in both 3D worlds. Considering that none of the previous methods achieves spatial accuracy and temporal consistency simultaneously, we emphasize that MC^2 achieves superior performance in all evaluation metrics measuring spatial accuracy and temporal consistency. We believe that our framework effectively incorporates monocular depth estimates to align them with the 3D world, thus broadening the scope and applicability of depth estimation across various applications where alignment between the 3D world and depth is crucial.

2. Method

Our goal is *an accurate depth estimation that are aligned with camera poses* by finding correspondences between context views within 3D space while synthesizing realistic views at arbitrary angles. We refer to Appendix A for preliminaries and review related work in Appendix B.

2.1. Scaled Feature Extraction

Relying solely on RGB images to extract image features $\boldsymbol{F}_i$ using 2D CNN encoders often fails to capture correspondences between images, especially in regions with no texture or where geometric disparities exist despite similar RGB values (See Appendix A). To address this limitation, we propose to incorporate depth estimates of the context views in addition to latent image features from the image encoder [21], denoted as $\boldsymbol{D}_i$, as 3D geometric priors. Here, we compute $\boldsymbol{D}_i$ using existing depth estimation methods (*e.g.*, ZoeDepth [5]). In practice, however, these depth estimates often do not match the camera poses computed by Structure-from-Motion (SfM) software (*e.g.*, COLMAP [36, 37]) as

Table 1. **Scaled depth estimation results.** Comparison of the depth estimation performance of MC^2 and baselines on ScanNet, and GMU-Kitchen datasets. Each scene in both datasets is reconstructed using SfM pipelines. **Bold** and underline indicate the best and runner-up, respectively. OB indicates out-of-bounds when calculating E_d.

Dataset	Type	Method	Metric Depth Accuracy				Pose-scaled Depth Accuracy			
			$\delta_1(\uparrow)$	REL($\downarrow$)	$E_d(\downarrow)$	$E_s(\downarrow)$	$\delta_1(\uparrow)$	REL($\downarrow$)	$E_d(\downarrow)$	$E_s(\downarrow)$
ScanNet	Monocular	DPT [10]	0.497	0.766	OB	0.727	0.361	0.806	OB	1.680
		ZoeDepth [5]	0.829	0.136	2.676	0.295	0.832	0.139	3.663	0.453
		DepthAnything [49]	0.396	0.545	7.538	0.658	0.251	0.503	40.381	1.606
	Multi-view	CVD [25]	0.341	0.427	3.654	1.692	0.301	0.456	11.820	0.768
		NVDS [46]	0.386	0.670	17.167	0.647	0.326	0.540	99.667	1.575
		IBRNet [45]	0.751	0.193	2.685	0.405	0.736	0.202	10.049	1.080
		GNT [42]	0.758	0.157	**2.173**	0.289	0.735	0.162	2.438	0.503
		MC^2 (ours)	**0.862**	**0.117**	2.429	**0.279**	**0.857**	**0.123**	**2.090**	**0.374**
GMU Kitchen	Monocular	DPT [10]	0.465	4.060	OB	OB	0.302	4.891	OB	75.877
		ZoeDepth [5]	0.620	0.380	15.299	1.176	0.562	1.315	17.874	1.142
		DepthAnything [49]	0.169	1.206	22.824	1.608	0.160	2.599	592.102	8.371
	Multi-view	CVD [25]	0.490	1.167	19.502	1.387	0.426	1.518	332.763	3.195
		NVDS [46]	0.329	0.646	17.717	12.303	0.292	1.543	140.230	2.439
		IBRNet [45]	0.720	0.384	14.169	1.164	0.501	1.308	36.602	1.507
		GNT [42]	0.647	0.390	14.831	1.167	0.400	1.250	9.886	1.083
		MC^2 (ours)	**0.759**	**0.331**	**13.992**	**1.129**	**0.716**	**1.220**	**9.393**	**0.629**

also mentioned in Appendix A.

To mitigate such a scale ambiguity in estimated depth maps, we introduce a depth rescaling to preprocess monocular depth estimations. MC^2 first obtain both monocular depth maps $\boldsymbol{D}_i$ and sparse depth $\boldsymbol{D}'_i$ from SfM for each context view $\boldsymbol{C}_i$. Since sparse reconstructions are noisy and have outliers, MC^2 excludes $\alpha\%$ of outlier minimum and maximum values of each estimation, and these values are not used in later stages. For the remaining valid pixel locations, MC^2 computes the median depth ratio $f_i > 0$ as an estimation of scale factor between SfM and monocular estimates by dividing $\boldsymbol{D}'_i$ with $\boldsymbol{D}_i$. After that, each $\boldsymbol{D}_i$ is scaled with a factor of f_i. This is simple yet effective at handling scale ambiguity in reconstructed scenes on-the-fly during optimization without needing the training or estimating additional values [33, 47]. Noteworthy, it also mitigates the inconsistencies in each monocular depth estimate by rescaling each mono estimated with the sparse depth maps which assume the coherent object despite its sparsity.

Ensuring Depth Consistency Across Views. Despite the rescaling that mitigates a scale ambiguity for each estimate, depth values still remain relative to the camera positions of the context views, resulting in different depth values in different context views for the same object. To utilize these rescaled depth values as additional sources for *consistent* feature matching of samples along a ray, MC^2 projects each scaled depth maps $f_i \cdot \boldsymbol{D}_i$ onto the querying target viewpoints using the camera poses $\boldsymbol{P}_i$ to obtain $\texttt{proj}(f_i \cdot \boldsymbol{D}_i)$. By adjusting the distance from the context view to the corresponding target viewpoint, it is possible to see how the scene would look from the target viewpoint. Furthermore, MC^2 extracts a 2D feature map $\boldsymbol{F}'_i$ for a projected depth map $\texttt{proj}(f_i \cdot \boldsymbol{D}_i)$ for each context view through a shared convolutional encoder network [21] to induce more locality in the projected depth maps $\texttt{proj}(f_i \cdot \boldsymbol{D}_i)$.

2.2. Factorized Attention for Context Features

We synthesize the target views from a set of context features $\{\boldsymbol{Z}_i\}_{i=1}^N$, where $\boldsymbol{Z}_i := (\boldsymbol{C}_i, \boldsymbol{F}_i, \boldsymbol{F}'_i) \in \mathbb{R}^{H\times W\times d}$. Here, it is crucial to identify the correspondence across N context views while selectively focusing along epipolar lines within the S samples of each context view. Therefore, we design a new transformer architecture consisting of repeating three different factorized transformer blocks inspired by the design of video vision transformers [1]. To achieve this, for a target image, MC^2 divides the target image into patches P, and samples S points along a ray of each pixel coordinate within the patch. Then, the model aggregates the projected 3D samples on the target ray to the N context views. Thus, the resulting context features are indexed by patch, view, and samples: $z \in \mathbb{R}^{P\times N\times S\times d}$.

For each view, patch, and ray transformer block of our architecture, we reshape the hidden state into different as $\mathbb{R}^{P\cdot S\times N\times d}$, $\mathbb{R}^{N\cdot S\times P\times d}$, and $\mathbb{R}^{P\cdot N\times S\times d}$ (respectively) and feed it to those encoder blocks. Remarkably, we find our view transformer design captures correspondence between views even in real-world complex situations, including occlusion, motion, and camera blur that may include low-quality frames. Moreover, because the patch transformer block is computed within the same patches, it induces more geometric locality to the extracted features. It also allows capturing the geometric properties of the target pixels, *e.g.*, shape edges, compared with using only two transformer blocks that consider pixel-wise correspondence without locality. Since the patches are derived from the target view and projected onto the context views, the geometric properties of each projected patch are effectively incorporated by reflecting the relative camera transformation between the target and context views. Lastly, the ray transformer block focuses on the relationship among samples within the same views.

We concatenate [cls] token to the context feature z and use the [cls] output from our architecture to compute a depth map in the following section. We empirically find that using [cls] token is crucial for disentanglement of depth from color (*e.g.*, the spot pattern should not be visible in synthesized depth maps but it should be distinct in color maps), compared with previous approaches using weighted average feature pooling [42, 45] or cross-attention [40].

2.3. Ordinal Rendering

To render a depth map and corresponding color for a query target view, we use the [cls] output from the stacked factorized transformer blocks discussed earlier in Section 2.2. In conventional neural rendering, this involves volume rendering [20, 23, 27, 45]. However, we find that this volume rendering, which predicts independent density value for each 3D point, is often problematic in image-based neural rendering [45] leading to incorrectly inferred geometry. This is because image-based neural rendering often lacks explicit modeling of density weights for absolute positions $\mathbf{x}$ but rather tends to simply blend RGB values, as explained in Appendix A. See Appendix E.2 for more details

To address this, instead of predicting an independent density weight for a 3D point, MC^2 models the density weights of samples on a ray as a probability distribution. Specifically, we adopt a binomial distribution based on the understanding that a ray will only intersect a single surface and that intersections should be denoted as 1 (for surfaces) or 0 (for empty spaces). This approach ensures that the density weights are ordered, which preserves the sequential nature of the ray within 3D. To this end, MC^2 predicts a two-channel output for both the mode q and the temperature t, to compute the probability score over the k^{th} sample index as given by the equation: $p(k; S, q) = \binom{S}{k} q^k (1-q)^{N-k}$ where S is the total number of samples following the design of recent works [2, 5]. Then, we apply $\texttt{softmax}(\{\log(p_k)/t\}_{k=1}^{S})$, which yields normalized values to ensure numerical stability. The temperature parameter t modulates the sharpness of the distribution, while the softmax normalization preserves the unimodality of the logits. Finally, we compute the final predicted color for the target view using the above probability scores as follows: $\tilde{\boldsymbol{C}}(\mathbf{r}) = \sum_{k=1}^{S} p_j(k) c_j(k)$.

3. Experiment

In this section, we provide an empirical evaluation of MC^2 to verify the suitability of our framework for estimating the depth of real-world videos. Recall that we are interested in estimating the depth at which the 3D scene exists, so we consider two settings: 1) The scene is reconstructed with an unknown arbitrary scale. In this setting, the camera poses are obtained using an SfM pipeline and thus often do not match the real world in metric scale. In addition, we consider the second setting where 2) camera poses are obtained using odometry sensors in the capture tools (aligned with the real world in metric scale) in Section 3.2. Please refer to Appendix C for a detailed experimental protocol.

3.1. Evaluation under SfM Reconstruction

We present the main results in the first scenario where the 3D world is reconstructed using SfM pipelines, thus not residing in the real-world. We conduct experiments with two in-the-wild, casually captured video datasets: ScanNet [8] and GMU Kitchen [14], both containing depth information. See Appendix C and Appendix D for a detailed description of the experimental setup for the first scenario and baselines.

Table 1 summarizes the experimental results of MC^2 and baselines. Overall, MC^2 significantly outperforms all the prior depth estimation methods by a large margin, leading to better scores across both aspects of spatial accuracy and temporal consistency. We visualize the qualitative depth estimation results of MC^2 and baselines in Appendix E.

3.2. Evaluation with Real-world Camera Poses

Next, we consider the scenario where the 3D world reconstructed from 2D images is aligned with the real-world to evaluate metric depth estimation performance. For the metric depth estimations, we conduct experiments on the iPhone dataset [13] that provides camera poses aligned with real-world scales. Please refer to Appendix C for a detail.

Table 2. **Unscaled depth estimation results.**

		Metric Depth Accuracy			
Type	Method	$\delta_1(\uparrow)$	REL($\downarrow$)	$E_d(\downarrow)$	$E_s(\downarrow)$
Monocular	ZoeDepth [5]	0.023	1.719	0.290	0.086
Multi-view	**MC^2 (ours)**	**0.377**	**0.561**	**0.171**	**0.049**

Table 2 summarizes a quantitative comparison between MC^2 and ZoeDepth. Our model, MC^2 demonstrates significantly better performance than the previous state-of-the-art metric depth model and these superior experimental results underline the efficacy of MC^2 for capturing the consistent depth estimate between long ranges of frames while accurately inferring the real-world metric depth by considering multi-view images as context to find correspondences between them. We visualize qualitative results in Appendix E.

4. Conclusion

In this paper, we propose MC^2, a novel framework for multi-view consistent depth estimation through image-based neural rendering. In contrast to previous approaches that treat depth estimation on a per-pixel basis, the main idea of MC^2 is to estimate depth in 3D space using contextual feature sets obtained from a stream of 2D RGB images. We believe that MC^2 has established new paradigms in depth estimation by enabling consistent and accurate 3D scene understanding that respects the residing 3D world.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 3

[2] Christopher Beckham and Christopher Pal. Unimodal probability distributions for deep ordinal classification. In *International Conference on Learning Representations*, pages 411–419. PMLR, 2017. 4

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 1

[4] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Localbins: Improving depth estimation by learning local distributions. In *European Conference on Computer Vision*, pages 480–496. Springer, 2022. 2

[5] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 1, 2, 3, 4, 8

[6] Amlaan Bhoi. Monocular depth estimation: A survey. *arXiv preprint arXiv:1901.09402*, 2019. 2

[7] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023. 1, 2, 3

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 4, 2

[9] Congyue Deng, Jiawei Yang, Leonidas Guibas, and Yue Wang. Rethinking directional integration in neural radiance fields. *arXiv preprint arXiv:2311.16504*, 2023. 1

[10] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 3, 2

[11] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. *arXiv preprint arXiv:2306.08637*, 2023. 3

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 1

[13] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Dynamic novel-view synthesis: A reality check. In *NeurIPS*, 2022. 4, 2, 3

[14] Georgios Georgakis, Md Alimoor Reza, Arsalan Mousavian, Phi-Hung Le, and Jana Košecká. Multiview rgb-d dataset for object instance detection. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 426–434. IEEE, 2016. 4, 2

[15] 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. 1

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1

[17] Ruibo Li, Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, and Lingxiao Hang. Deep attention-based classification network for robust depth prediction. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV 14*, pages 663–678. Springer, 2019. 1, 3

[18] Xingyi Li, Zhiguo Cao, Huiqiang Sun, Jianming Zhang, Ke Xian, and Guosheng Lin. 3d cinemagraphy from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4595–4605, 2023. 1

[19] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022. 1

[20] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023. 1, 4, 2

[21] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017. 2, 3

[22] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 1

[23] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022. 4

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[25] 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. 1, 3, 2

[26] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[27] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. 1, 4, 2

[28] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1

[29] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 4, 5

[30] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 4, 5

[31] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 1

[32] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Advances in Neural Information Processing Systems*, pages 12179–12188, 2021. 1, 2

[33] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 1, 3, 2, 4, 5

[34] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. 2023. 1

[35] Saurabh Saxena, Junhwa Hur, Charles Herrmann, Deqing Sun, and David J Fleet. Zero-shot metric depth with a field-of-view conditioned diffusion model. *arXiv preprint arXiv:2312.13252*, 2023. 1, 2

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 1, 9

[37] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 1, 9

[38] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. *Visual Communications and Image Processing 2000*, 4067:2–13, 2000. 1

[39] Marek Šimoník. 3d reconstruction on ios. 2018. 3, 9

[40] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pages 156–174. Springer, 2022. 4

[41] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 1

[42] Mukund Varma T, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, and Zhangyang Wang. Is attention all that neRF needs? In *International Conference on Learning Representations*, 2023. 3, 4

[43] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018. 3

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[45] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 3, 4, 1, 2, 5

[46] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9466–9476, 2023. 1, 3, 2

[47] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5610–5619, 2021. 1, 3

[48] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 1

[49] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024. 3

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1

[51] Zhoutong Zhang, Forrester Cole, Richard Tucker, William T Freeman, and Tali Dekel. Consistent depth of moving objects in video. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021. 2

MC^2: Multi-view Consistent Depth Estimation via Coordinated Image-based Neural Rendering

Supplementary Material

Website: https://subin-kim-cv.github.io/MC2/

A. Preliminaries

We review image-based view synthesis and scale ambiguity in multi-view geometry that are essentials of our proposed method.

A.1. Image-based View Synthesis

The goal of image-based view synthesis is to render novel views by using pixel values from input RGB images as a context set [38]. Recently, this approach combined with deep learning techniques stands out for its efficiency in handling casually captured long, unbounded scenes including an outdoor scene [20]. This has been achieved as an interpolation-based approach, it often does not require explicit geometry for memorizing the scenes compared to dense representations such as neural radiance fields [27, 31], voxel grids [22, 28, 41], or 3D Gaussian splatting [16, 48] which requires the volume construction of proxy geometry for every 3D point.

Formally, given a stream of N input views $\{\boldsymbol{C}_i, \boldsymbol{P}_i\}_{i=1}^{N}$ as RGB images $\boldsymbol{C}_i \in \mathbb{R}^{H\times W\times 3}$ and their corresponding known camera poses $\boldsymbol{P}_i \in \mathbb{R}^{3\times 4}$, image-based view synthesis renders the target RGB image $\boldsymbol{C}$ from an arbitrary camera angle $\boldsymbol{P}$. This is achieved by calculating a 3D point $\mathbf{x} \in \mathbb{R}^3$ of each 2D-pixel coordinate $\mathbf{s} \in \mathbb{R}^2$ of $\boldsymbol{C}$, projecting $\mathbf{x}$ back onto each of the 2D context views, and then interpolating the RGB values at those projections. However, since each $\mathbf{x}$ of the target $\boldsymbol{C}$ are not known a priori,[2] a ray $\mathbf{r} \in \mathbb{R}^3$ with a view direction $\mathbf{d} \in \mathbb{R}^3$ is cast from each pixel coordinate $\mathbf{s}$ to accumulate the density of 3D sample points on the ray. Specifically, image-based rendering samples S query 3D points $\{\mathbf{x}_j\}_{j=1}^{S}$ along a ray from near depth to far depth, and projects into all context views using the respective camera poses. Then, the crucial task is to perform feature matching across these N context views for the S sampled points along the epipolar lines to estimate the convergence to a 3D point on the ray.

Image-based Neural Rendering. Recently, IBRNet [45] has integrated this classical image-based rendering with recent neural volume rendering (*i.e.*, NeRF [27]). Specifically, for each context view $\boldsymbol{C}_i$, IBRNet uses a convolutional encoder network [15] to obtain dense features from each RGB image $\boldsymbol{C}_i$ and extract features at the pixel location $\mathbf{s}$ using these feature maps. When considering S samples along the ray $\mathbf{r}$, IBRNet employs a ray transformer that processes the sequence of aggregated features along the ray to predict per-sample colors and densities (c_j, σ_j). The final pixel color $\tilde{\boldsymbol{C}}(\mathbf{r})$ for the ray is then computed from this sequence of colors and densities using standard NeRF volumetric rendering techniques [27].

However, we find that image-based neural rendering can easily prioritize the simpler task of blending colors using features obtained from contextual views over the more complex task of establishing correspondence between sampled contextual features along rays in 3D space.[3] This occurs because image-based neural rendering operated in 2D-pixel space often lacks a comprehensive understanding of the 3D scene [9], leading to a preference for color blending over accurate correspondence matching of 3D points. This phenomenon is easily verified as shown in Appendix E.2, where depth estimates aligned with camera poses suggest that inferred geometry could be interpreted as correspondence matching points. This challenge is compounded when the model relies solely on RGB images, as RGB matching can occur even when each projected pixel location in the context views represents different objects, in the absence of geometric priors.

A.2. Scale Ambiguity in Multi-view Geometry

Typically, Structure-from-Motion (SfM) methods such as COLMAP [36, 37] are used in multi-view stereo to reconstruct the scene from 2D images by registering camera parameters (extrinsic and intrinsic) for each frame $\boldsymbol{C}_i$. During the multi-view stereo process, a fused sparse 3D point cloud is derived, and per-view depth maps are obtained by projecting the fused 3D point clouds as a by-product of multi-view stereo. However, the resulting depth map D_i' tends to be extremely sparse and has many outliers. Furthermore, the registered camera poses are not exactly aligned with the real-world *metric* scale, as SfM reconstructs scenes *only to scale*, *i.e.*, each scene is scaled by arbitrary, individual scale factors f_i to align with real-world metric poses, leading to scale ambiguity. Thus, although learning-based monocular depth estimates provide a dense and quite accurate estimate for a single image [5, 34, 35], there is often a mismatch in scale between the SfM reconstructions and the learning-based depth estimates.

[2] The obtained reconstructions with camera poses from Structure-from-Motion software are extremely sparse with severe outliers.

[3] Blending one color may be sufficient to combine three RGB vectors.

B. Related Work

Depth Estimation. Much of the work addresses the long-standing problem of interpreting 3D spaces from flat 2D RGB images by estimating depth from only given 2D images [7, 32]. In particular, recent advances aim at estimating depth in absolute physical units (*e.g.*, meters) from a single RGB image [5, 35]. However, with only a single 2D image, there is an inherent challenge in inferring spatial relationships [6], and although multiple images are fed to infer geometry, they do not have the ability to account for the spatial relationship between them, as the depths are estimated independently for each image [46], thus exhibiting temporal flickering. To mitigate this, there have been efforts to incorporate the spatial relationship between the given images, such as the 3D camera pose information [25, 51]. However, they still require the depth estimates to be obtained independently at the pixel level, so the overall quality is limited by the initial estimates. Compared to existing approaches that estimate or optimize multi-view depths already translated into pixel space, we shift the problem directly to feature matching in 3D space, which is a novel setting.

Neural Rendering. Recently, neural scene renderings have shown great potential in parameterizing complex 3D scenes as a neural network by mapping 5D coordinates to RGB values and densities [27]. Intriguingly, these approaches have exhibited strong generalization power to synthesize plausible renderings at novel camera poses. While the results are seemingly satisfactory, they often suffer from inaccurate correspondence modeling [13] with ground-truth 3D scenes. This often leads to degraded quality for novel views or causes temporal flickering between views. In addition, previous approaches to scene rendering [10, 13, 33] have shown that incorporating depth observations into 3D scene reconstruction facilitates the synthesis of realistic views in a variety of scene reconstruction situations, *e.g.*, few-shot novel view synthesis, dynamic view modeling, reduced artifacts, or faster training. In contrast, we are interested in depth estimation through neural rendering, a direction contrary to prior efforts.

C. Experimental Details

Training Details. MC^2 is pre-trained our model with IBR collected dataset [45] and LLFF datasets [26], consisting of various real-world datasets captured from handheld cellphone captures using COLMAP [36, 37] to estimate the camera pose and scene bounds for the captures. We train our model using 8 context views with a batch size of 288 rays for $300k$ iterations and a patch size of 4 for a total of 16 rays per patch. We use the AdamW optimizer [24] with a learning rate of $4e^{-4}$ for the image and depth coder and $2e^{-4}$ for other components to facilitate capturing of features for matching, combined with a cosine schedule to gradually reduce the learning rate. All network parameters in MC^2 are optimized end-to-end using a combined loss function that includes a mean square error (MSE) term on the color output and a scale-invariant log loss [4] for on the depth output, comparing the predicted depth to the rescaled target view derived from monocular depth estimates as described. The model was then fine-tuned over $5k$ iterations per scene, using the same training configurations as in the corresponding pretraining phase. All experiments were conducted on a 4 GPU (NVIDIA A100 40GB) and 24 instances of a virtual CPU (Intel® Xeon® CPU @ 2.20GHz).

For dynamic scenes, where the epipolar constraints are violated by the motion of moving objects, we integrate MotionMLP as proposed in DynIBar [20], which addresses dynamics in image-based neural rendering. Inspired by DynIBar, we also optimize MotionMLP with our networks to aggregate multi-view image features in scene motion-adjusted ray space, which allows us to correctly reason about spatio-temporally varying geometry and appearance. Unlike DynIBar, which separates scenes into static and dynamic parts with two different models for each, we simplify training by optimizing a single network to handle both static and dynamic parts. In addition, we do not employ additional training strategies such as cross-time rendering, and we do not add additional supervision (*e.g.*, optical flow) as in DynIBar to guide the optimization of MotionMLP for better handling of moving objects for oblique novel views. Nevertheless, we believe that advanced techniques to address dynamics could improve the optimization in handling complex motions within the scenes.

Dataset Details. We consider 3 benchmark datasets to evaluate our method compared to baselines. For ScanNet and GMU-Kitchen datasets, we use three sampled scenes from each dataset and each sampled scene contains 200 frames in total. For all these frames, we run COLMAP [36, 37] to obtain camera poses and sparse 3D point clouds. Depth estimates from our approach and baselines were evaluated on every 4th frame of the total frames.

- **ScanNet** [8]: To compare the performance of depth estimation on indoor scenes, we perform experiments on the ScanNet dataset with the following scenes: `scene0710_00`, `scene0736_00`, `scene0770_00`, each consisting of 200 frames in total. We resize the RGB image to half the resolution of 480×640, following the ground truth depth map provided in the dataset, and sample the images at stride 2 to handle a larger field of view.

- **GMU-Kitchen** [14]: To compare the performance of depth estimation in realistic cluttered environments, we perform experiments on the GMU-Kitchen dataset with the following scenes: `gmu_scene_001`, `gmu_scene_004`,

gmu_scene_006, each consisting of 200 frames in total. We resize the RGB image and the corresponding depth map to half the original resolution, 960×540, and sample the images with stride 2 to handle a larger field of view.

- **iPhone** [13]: To compare the performance of depth estimation in the metric scale, we consider the iPhone dataset. Each sequence in the iPhone dataset is captured by the handheld iPhone using the Record3D [39] mobile applications, which provide camera poses and RGB and corresponding lidar sensor depth information. We evaluate the following scenes: apple, paper-windmill, teddy, mochi-high-five, wheel, and sample 50 consecutive frames in each.

Evaluation Details. For evaluation, we measure depth accuracy using two different scale factor to show how well the estimated depth is aligned with the ground truth depth and obtained depth map respects to the reconstructed 3D scenes, respectively. First, we compute a scale factor by comparing the estimated depth maps with the ground truth depth, following existing evaluation setups [17, 25, 46, 49]. Additionally, to measure how the estimated depth matches the 3D world in which the reconstructed scene resides, we also compute a scale factor by comparing the estimated depth maps to a sparse point cloud obtained from SfM. By doing so, we show how well the estimated depth is aligned with the ground truth depth, and obtained depth map respects the reconstructed 3D scenes, respectively.

Evaluation Metrics. We evaluate the video depth estimation considering both spatial accuracy and temporal consistency. First, we adopt the commonly applied metrics and evaluation setups [43, 49] which measure the spatial accuracy of depth estimation by computing the absolute relative error (REL) and δ_1 with the median ground-truth scale strategy due to the scale ambiguity. In particular, we derive a single scale factor that matches the 3D world reconstructed from the image stream with all the depth estimates for the video and apply this to all depth estimates. We then calculate frame-wise accuracy and average this across the entire video. While this single scale factor ensures average consistency in depth estimations, it doesn't account for variations due to median scaling. Therefore, to better evaluate temporal consistency, we design two metrics, E_s and E_d following Luo *et al*. [25]. E_s measures the temporal instability of depth maps by tracking the 2D-pixel points in a video. To track pixel points, we use a recent point tracker [11]. Ideally, the tracked 2D points should converge to a single point in 3D, thus we compute the Euclidean distances between 3D points over consecutive frames. Moreover, E_d measures the accumulated errors; although 3D-tracked points may appear stable over successive frames, errors can be accumulated, resulting in drift over time. To quantify the drift for a given 3D point, we compute the maximum eigenvalue of the covariance matrix of the 3D trace, which intuitively indicates the dispersion of the 3D points over time.

D. Description of Baseline Methods

In this section, we briefly describe the baselines of depth estimation that we consider for evaluating our framework. We compare MC^2 with baselines of two different model types: monocular depth estimation and multi-view aware approaches. For all of the baseline methods, we sincerely follow their reported experimental setups.

Monocular Depth Estimations. For the monocular depth estimation approaches, we consider the following approaches for baselines.

- **DPT** [10] is the relative depth estimation network that deals with the large-scale variations in different types of environments by factoring out the scale factor, thus depth predictions per pixel are only consistent **relative** to each other across image frames and the scale factor is unknown. We use DPT-L in our comparisons
- **ZoeDepth** [5] is a monocular metric depth estimation network, and we utilize ZoeD-M12-NK which is trained jointly on both indoor and outdoor domains.
- **DepthAnything** [49] proposes to train the relative depth estimation network using large-scale unlabeled RGB images without depth annotations. We perform experiments using ViT-L encoder for DepthAnything.

Multi-view Depth Estimations. For the depth estimation approaches that incorporate multi-view images for consistent depth estimation across views, we compare with the following approaches for baselines. First, we compare with existing depth estimation that mainly uses data-driven priors to enforce smoothness between consecutive frames in a video.

- **CVD** [25]: Following their experimental setups, we additionally run COLMAP with multi-view stereo to obtain more dense depth maps and more accurate camera poses compared to ones that we utilize during training.
- **NVDS** [46]: We use NVDS with the DPT network which performs better than NVDS combined with MiDas [7] in their reports for consistent depth estimates.

Moreover, we compare MC^2 with existing image-based view synthesis which synthesizes target images by interpolating the context features, such as ours, which can incorporate multiple images simultaneously to synthesize the depth map and corresponding RGB images for the target viewpoint. In particular, we exclude test images in context views and target views when optimizing IBRNet [45], GNT [42], and ours to synthesize depth for *unseen* viewpoints.

Table 3. **Scaled depth estimation results.** Comparison of the depth estimation performance of MC^2 and baselines which require geometry modeling on ScanNet and GMU-Kitchen datasets. Each scene in both datasets is reconstructed using SfM pipelines.

Dataset	Type	Method	Metric Depth Accuracy				Pose-scaled Depth Accuracy			
			$\delta_1(\uparrow)$	REL($\downarrow$)	$E_d(\downarrow)$	$E_s(\downarrow)$	$\delta_1(\uparrow)$	REL($\downarrow$)	$E_d(\downarrow)$	$E_s(\downarrow)$
ScanNet	NeRF	Roessle *et al.* [33]	0.510	0.308	4.128	0.465	0.276	0.701	119.390	1.885
	Image-based	**MC^2 (ours)**	0.862	0.117	2.429	0.279	0.857	0.123	2.090	0.374
GMU Kitchen	NeRF	Roessle *et al.* [33]	0.640	0.441	14.462	1.180	0.346	1.288	30.668	1.478
	Image-based	**MC^2 (ours)**	0.759	0.331	13.992	1.129	0.716	1.220	9.393	0.629

Table 4. **Unscaled depth estimation results.** Comparison of the depth estimation of MC^2 and baselines on iPhone dataset. Each sequence on the iPhone dataset is reconstructed using a visual odometry sensor, aligned with real-world metric depth.

Dataset	Type	Method	Metric Depth Accuracy			
			$\delta_1(\uparrow)$	REL($\downarrow$)	$E_d(\downarrow)$	$E_s(\downarrow)$
iPhone	NeRF	Nerfies [29]	0.076	0.563	0.041	0.032
		HyperNeRF [30]	0.028	0.675	0.051	0.045
	Image-based	**MC^2 (ours)**	0.377	0.561	0.171	0.049

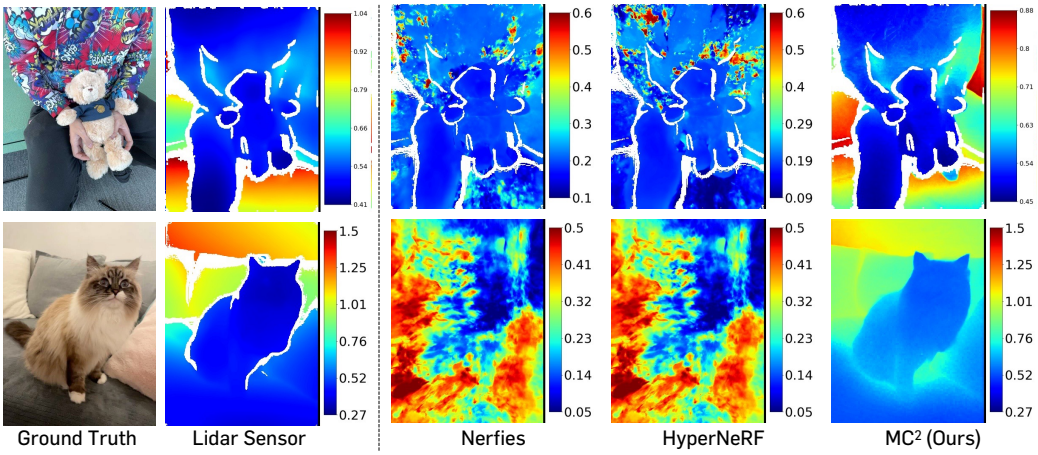

Figure 2. **Video depth estimation results.** Illustration of depth estimates from MC^2 and baselines. Each video is captured with a handheld iPhone Pro. The color bar on the right is in meters (m).

- **IBRNet** [45] uses a ray transformer that estimates radiance and volume density at continuous 5D locations by allowing visibility reasoning over larger spatial scales. Compared to IBRNet, we propose a view and patch transformer. We utilize pretrained IBRNet on their official code implementation and additionally fine-tune the network for $10k$ following their fine-tuning strategies.

- **GNT** [42] utilizes a ray and view transformer. Compared to ours, they compute cross-attention given a camera viewpoint as query and context features as key and values in attention. As shown in Appendix E.4, we found that such an architectural design often leads to entanglement of color and depth. We utilize pretrained GNT on their official code implementation and additionally fine-tune the network for $10k$ following their fine-tuning strategies.

E. More Experimental Results

E.1. Depth Estimations relying on 3D Volumes.

We additionally compare with the neural scene rendering approaches that model scenes using the 3D geometry volume of NeRF, as opposed to image-based neural renderings such as IBRNet, GNT, and ours that do not rely on geometry fitting.

First, to verify the efficacy of our depth rescaling scheme without explicit geometry, we consider Roessle *et al*. [33] as a baseline that obtains dense depth maps adapted to the given camera poses. Specifically, Roessle *et al*. [33] employs depth completion on the SfM depth to obtain dense depth priors. For this, they train a depth completion network before optimizing NeRF. After training the depth completion network, one can obtain dense depth priors from the trained network, and then supervise the geometry recovered by NeRF using them.

As shown in Table 3, ours achieve better depth estimation results across all metrics. In addition, our MC^2 is computationally efficient, since we use only a rescaling scheme to adjust the depth maps to the given camera poses on the fly, without additional training of monocular depth estimation networks.

Next, we compare with dynamic neural scene rendering approaches to conduct depth estimation with moving objects. For this, we consider Nerfies [29], HyperNeRF [30] as baselines and the results is shown in Table 4.

Although Nerfies and HyperNeRF have lower E_d and E_s scores, this is because E_d and E_s are used to measure consistency with a point tracker by tracking consecutive depth maps, rather than measuring the accuracy of depth estimates. Since both methods rely on 3D geometry to model the scene, they may achieve higher scores than ours regardless of the accuracy of the estimated depth maps, as evidenced by a significantly lower δ_1 score for them compared to ours. We also visualize the depth estimation results qualitatively in Appendix F.

E.2. Volume Rendering in Image-based View Synthesis.

As shown in Figure 3, IBRNet typically assigns the highest weight to the last index in the sample along the ray. This phenomenon occurs primarily when IBRNet struggles to identify correspondences between context features derived solely from image encoders, especially in textureless regions (*e.g*., walls, flat surfaces). We conjecture that this happens because IBRNet's primary focus is on plausible view synthesis rather than accurate depth estimation. Such a bias manifests itself in the following ways: objects that are predicted to be farther away and thus to have greater depth tend to show only small pixel shifts across different viewpoints, even with significant changes in camera perspective; on the other hand, objects that are perceived to be closer show pronounced pixel movement within the images, even with minimal changes in camera angle. Thus, when optimizing image-based neural rendering networks for color image synthesis, there is a strong bias toward synthesizing images in which the relative motion of objects matches their expected real-world behavior. This prioritizes visual realism at the expense of accurate depth estimation, especially in the absence

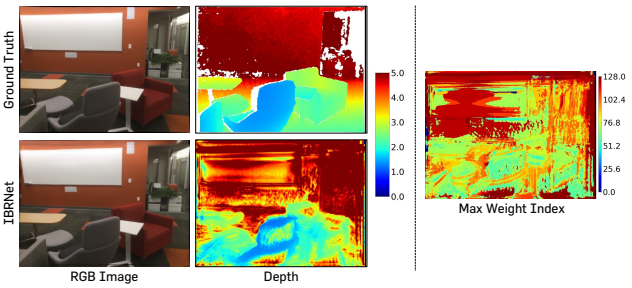

Figure 3. **Qualitative results of IBRNet on ScanNet.** Illustration of depth estimates and corresponding synthesized RGB images from IBRNet [45]. The color bar on the right side of the depth maps indicates the depth scale in meters. Additionally, the index value with the largest predicted density weight from IBRNet is shown (right), and the color bar indicates the sample index in the range of 0 to 127.

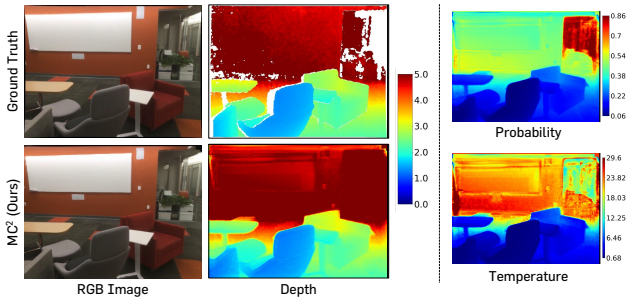

Figure 4. **Qualitative results of MC^2 on ScanNet.** Illustration of depth estimates and corresponding synthesized RGB images from MC^2. The color bar on the right side of the depth maps indicates the depth scale in meters. Probability and temperature are also visualized.

of explicit geometric modeling such as image-based view synthesis.

E.3. Ordinal Rendering.

We introduce ordinal rendering, which models density weights along a ray rather than as isolated densities for individual 3D points. This approach, illustrated in Figure 4, allows for a more accurate final depth map. We visualize the probability values that indicate the confidence in the corresponding predicted weights at the sample index associated with the model, strongly considering that the actual depth falls within the range of that bin. In addition, by adjusting the temperature values, one can control the smoothness of the distribution, with higher temperatures resulting in a more even probability distribution across all depth bins. This helps to reduce the dominance of a single, overly confident depth prediction in favor of a more balanced, average depth output. This method differs from IBRNet in that it avoids assigning disproportionate weight to the extremes of the sample range and instead controls the smoothness of the distribution through temperature adjustments. With this ar-

Table 5. **Ablation study.** Results show that our main contributions—rescaled depth feature, three-way factorized transformer, and ordinal rendering—lead to a significant performance gain in depth estimations.

Depth Rescaling	Three-way Transformers	Ordinal Rendering	Photometric		Metric Depth Accuracy				Pose-based Depth Accuracy			
			PSNR(↑)	LPIPS(↓)	δ_1(↑)	REL(↓)	E_d(↓)	E_s(↓)	δ_1(↑)	REL(↓)	E_d(↓)	E_s(↓)
✗	✓	✓	19.079	0.208	0.663	0.364	30.218	1.946	0.618	1.420	26.962	1.528
✓	✗	✓	17.792	0.245	0.453	0.487	32.018	1.994	0.378	1.511	35.403	2.008
✓	✓	✗	18.076	0.285	0.656	0.372	30.260	1.948	0.621	1.447	31.890	1.648
✓	✓	✓	**20.213**	**0.165**	**0.799**	**0.330**	**27.485**	**1.867**	**0.755**	**1.409**	**5.564**	**0.665**

chitectural design, we predict the distribution of depths while respecting the inherent order and spacing of depth levels and addressing issues of discretization and arbitrariness present in non-ordinal approaches.

E.4. Ablation studies.

To verify the effectiveness of each component, we conduct an ablation study of our model `gmu_scene_001` under the GMU Kitchen datasets by removing each component and then measuring the depth estimation performance in addition to the photometric synthesis results of the target view. First, when removing depth rescaling, we exclude only the rescaling and projection component, while retaining all other depth-related components, including depth encoders and shift-invariant loss. When removing the three-way factorized transformers, we replace our proposed three-way factorized transform blocks with ray transformers. Finally, when removing the ordinal rendering, we replace ordinal rendering with alpha blending following conventional neural rendering approaches or image-based view synthesis approaches. As verified in Table 5, without any of the components consisting of MC^2, the quality of depth estimation gets dramatically worse, which validates how MC^2 effectively estimates depth.

F. More Qualitative Results

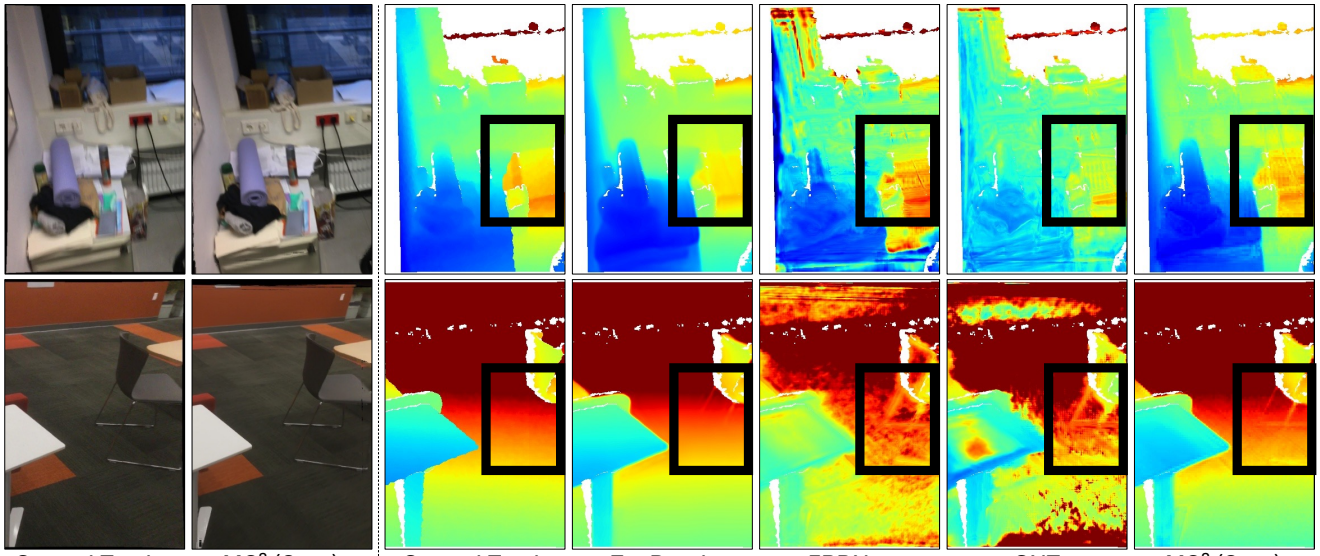

Figure 5. **Qualitative comparison on ScanNet.** Illustration of depth estimates and the corresponding synthesized RGB images. MC^2 renders a spatially accurate depth map while capturing thin structures, *e.g.*, wire and radiator (upper) and chair legs (bottom), which is often missing even with Ground Truth obtained with lidar sensors. In addition, MC^2 identifies the difference between shape and color while synthesizing deblurred color images in high quality with corresponding accurate depth maps. On the other hand, IBRNet and GNT struggle to distinguish this, demonstrated in the texture pattern observed at synthesized depth maps.

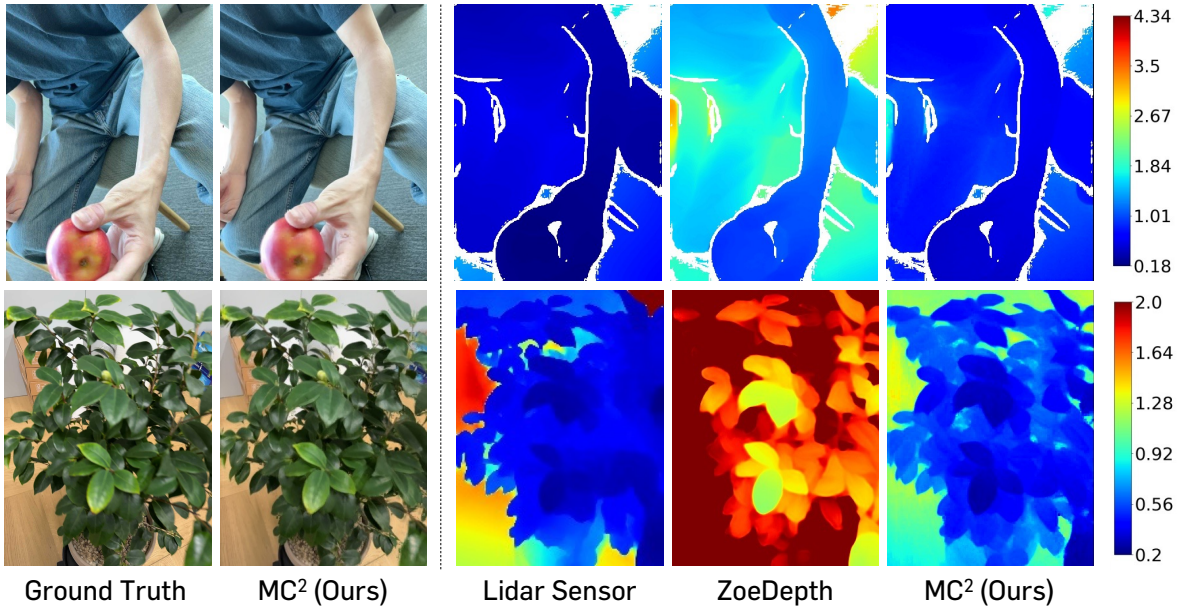

Figure 6. **Qualitative video depth estimation results.** Illustration of depth estimates and the corresponding synthesized RGB images. Each video is captured with a handheld iPhone Pro. The color bar on the right is in meters (m). MC^{2}successfully measures the distance even for unseen views between the camera and the object of interest by considering multiple viewpoints, while monocular metric depth estimate from ZoeDepth [5] overpredicts the object of interest compared to real metric units by predicting it to be farther than 2 meters. In addition, MC2 achieves spatially accurate depth estimates, even more than lidar depth measurements. For example, MC2 successfully measures the distance between the object and the background while capturing nuanced distances, especially in the plant example (below) where lidar measurements fail to distinguish the plant from the wall and finer gaps between the leaves with notoriously noisy around edges.

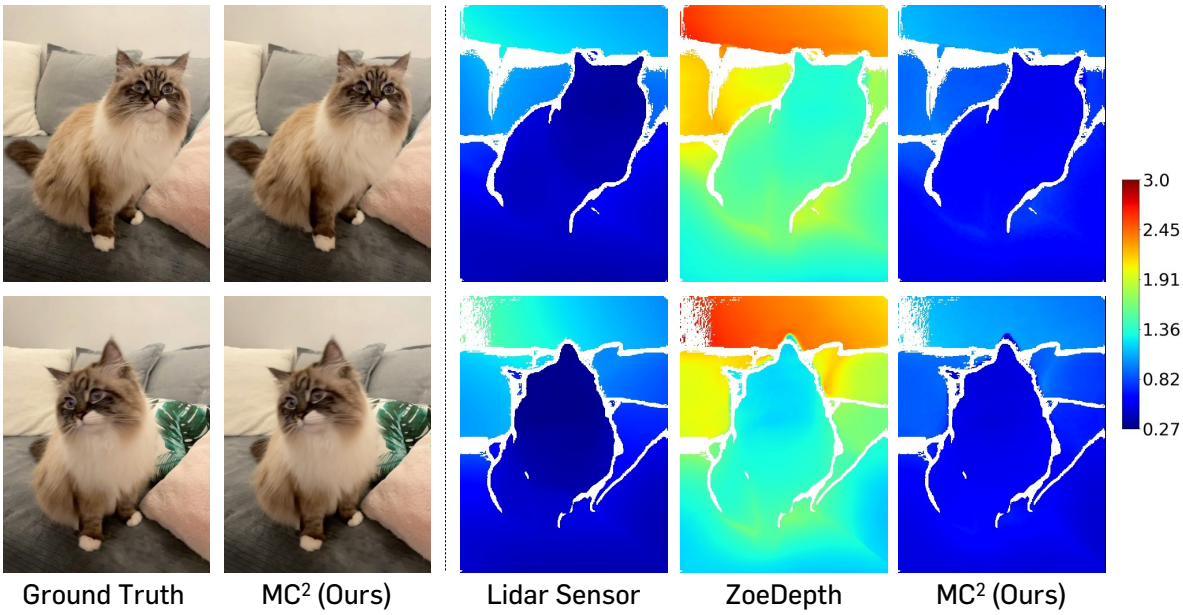

Figure 7. **Qualitative comparison on iPhone dataset.** Illustration of depth estimates and the corresponding synthesized RGB images on mochi-high-five sequence. The color bar on the right is in meters (m). MC^2 provides consistent depth estimates over time, capturing subtle movements such as a cat turning its head while remaining seated. On the other hand, ZoeDepth provides inconsistent estimates for the front of the sofa where the cat is sitting, as shown by the differences between the top and bottom images.

G. Limitation

Limitation and Future Works. MC^2 relies on precomputed camera poses, often derived from Structure-from-Motion, such as COLMAP [36, 37] or provided by the mobile app, such as Record3D [39]. An interesting future direction is to integrate the MC^2 with the reconstruction of the scenes. This could include the development of algorithms capable of refining imprecise camera poses or, alternatively, optimizing camera poses in parallel with other aspects of scene understanding. This approach could improve the accuracy and applicability of MC^2 in various real-world scenarios, leading to improved depth estimation and scene reconstruction results.

Potential Negative Societal Impact. While depth estimation by MC^2 can be beneficial for various applications such as autonomous driving, 3D modeling, and augmented reality, the emergence of unexpected behavior or undesirable artifacts within MC^2 can lead to misrepresentations of real-world environments. In domains that rely heavily on accurate data for critical decisions, such as surveillance and autonomous vehicle navigation, the introduction of unexpected behaviors or artifacts must be carefully managed. To ensure the reliability of systems using depth estimation, it is essential to conduct thorough investigations and implement robust mitigation strategies to minimize potential risks, thereby increasing the overall safety and effectiveness of these applications.