Supplementary Material
MorpheuS: Neural Dynamic 360° Surface Reconstruction from Monocular RGB-D Video

Hengyi Wang Jingwen Wang Lourdes Agapito
Department of Computer Science, University College London
{hengyi.wang.21, jingwen.wang.17, l.agapito}@ucl.ac.uk

1. Implementation Details

1.1. Data Pre-processing

We perform a set of data pre-processing steps to the raw RGB-D sequences before training MorpheuS.

Object Masks. For real-world datasets (KillingFusion, DeepDeform, iPhone), we extract foreground masks using off-the-shelf tools. Specifically, we use RVM [5] for humans and MiVOS [2] for other objects.

Coordinate System. Following NDR [1], we subtract out the rigid motion from the target object and convert the world coordinate frame to be centered at the target object using robust ICP [12]. The scale of the new coordinate frame is adjusted such that the object roughly fits in a unit sphere.

Pseudo Observations. As Zero-1-to-3 [7] assumes that all camera poses could be parameterised in polar coordinates (radius, polar and azimuth angles), i.e. the camera's viewing direction (z-axis) always perfectly points to the object centre (See the red cameras in Fig. 1). However, in real-world scenarios, this assumption does not hold because the camera's orientation does not depend on its translation w.r.t. the object, and thus the target object does not always appear in the middle of the image observation (See the green cameras in Fig. 1). To make the diffusion prior compatible with arbitrary camera poses in practical scenarios, for every real camera pose $\mathbf{T}_{wc} = \begin{bmatrix}\mathbf{R}_{wc} & \mathbf{t}_{wc}\end{bmatrix}$ we create a pseudo camera associated with it that satisfies the polar-coordinate constraint. The pseudo camera pose $\mathbf{T}'_{wc} = \begin{bmatrix}\mathbf{R}'_{wc} & \mathbf{t}'_{wc}\end{bmatrix}$ is computed by moving the original camera center on its image plane with its orientation fixed until the camera's z-axis passes through the object center, i.e. $(0, 0, 0)$:

$$\mathbf{t}'_{wc} = -\mathbf{R}_{wc}[:, 2] \cdot \mathbf{t}_{wc}, \quad \mathbf{R}'_{wc} = \mathbf{R}_{wc}, \tag{1}$$

where $\mathbf{R}_{wc}[:, 2]$ denotes the last column of the camera-to-world rotation matrix. Fig. 1 (first row) shows two examples in the snoopy and duck sequence, where real and pseudo cameras are shown in green and red respectively. We further create pseudo-observations from those pseudo cameras

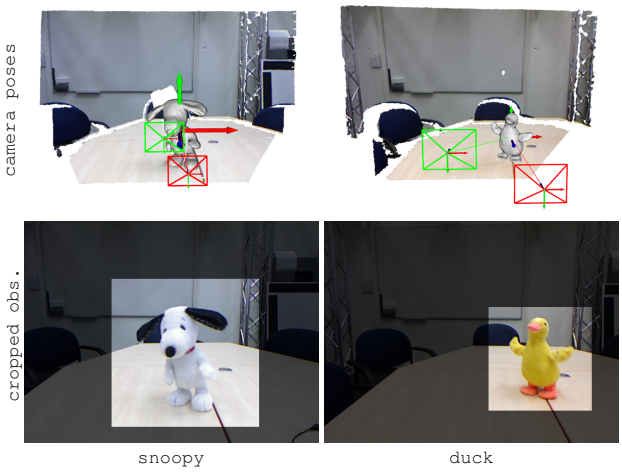

Figure 1. **Data Pre-processing.** In order to make the diffusion prior compatible with arbitrary real-world camera poses (shown in green) in casually captured video sequences, we create pseudo cameras that directly point to object center (shown in red) and obtain the pseudo-observations from the raw image.

by projecting the object center $(0, 0, 0)$ onto the image plane with pseudo camera poses to obtain the center pixel location for each frame and then cropping the raw image observations (RGB, depth and object mask) around the center pixel. See Fig. 1 (second row) for a demonstration.

Synthetic Dataset. For the 4 synthetic sequences (AMA-samba, AMA-swing, Eagle-1, Eagle-2) with per-frame GT meshes and multi-view real image observations, we perform the previous step for all the available camera views, but only one view is selected for optimizing our model.

1.2. Hyper-parameters

Deformation Field. Our deformation field consists of 3 major components: 1) Multi-resolution deformation code $\mathcal{V}_t(\cdot)$, 2) Deformation network $D(\cdot)$, and 3) Topology network $T(\cdot)$. The multi-resolution code has 3 levels, with the resolution of $[N/8, N/4, N]$, where N is the number of

frames in the sequence. The feature dimension of each level is set to be 16. For the deformation network and topology network, the number of the frequency band for positional encoding is set to be 6 and 4 respectively. The MLP used for those two networks consists of 6 hidden layers with 128 hidden units. The dimension of the ambient coordinate $\mathbf{x}'_a$ predicted by the topology network is set to be 2.

Canonical Field. We represent the SDF and color of the canonical field with two Hash grids $\mathcal{V}_s$ and $\mathcal{V}_c$. Both Hash grids have 16 levels, and the feature dimension at each level is set to 2. The Hash features $\mathcal{V}_s(\mathbf{x}'_m)$ and $\mathcal{V}_c(\mathbf{x}'_m)$ are obtained via concatenating the tri-linear interpolated feature vectors at each level. In order to perform the geometric initialization, the Hash feature of the SDF field $\mathcal{V}_s(\mathbf{x}'_m)$ is concatenated with the 3D coordinate of the query point $\mathbf{x}'_m$. Ideally, one can also use joint encoding strategy [11] as long as preserving 3D coordinates only and masking out the rest part. The SDF decoder $f_\gamma(\cdot)$ takes in the 3D coordinate, Hash feature, and the ambient coordinate, predicts the SDF value and a 16-D geometric feature $\mathbf{h}$. The geometric feature and Hash feature of the color grid are then fed to the color decoder $f_\alpha(\cdot)$ for decoding the color values. Both decoders are 3-layer MLPs with 64 hidden units.

Optimization. We train MorpheuS with Adam [3] optimizer and an EMA decaying of 0.95 for $E_{\max} = 2000$ epochs. We adopt the following scheduling strategy for the learning rate μ:

$$\mu = \begin{cases} \mu_1 & \text{if } E \leq 0.5E_w \\ \mu_1 + \frac{2E-E_w}{E_w}(\mu_2 - \mu_1) & \text{if } E \leq E_w \\ \mu_2(\cos(\frac{E-E_w}{E_{\max}-E_w}\pi)\frac{1-k}{2} + \frac{1+k}{2}) & \text{if } E > E_w \end{cases}, \tag{2}$$

where E is the epoch and $E_w = 200$ is the number of warm-up epochs. A small initial learning rate $\mu_1 = 5e-6$ is used for better initializing the canonical field during the first phase of the warm-up stage ($E \leq 0.5E_w$). In the second phase ($0.5E_w < E \leq E_w$) of the warm-up stage, the learning rate is then linearly increased to $\mu_2 = 5e-4$. After the warm-up stage, the learning rate is scheduled following a cosine annealing protocol. The value of k is set to be 0.05.

For each epoch, the optimization alternates between real and virtual views and the ratio of sampled virtual views and real views is set to be 0.1. For the training of real view, at each iteration, we randomly sample a batch of 2048 rays from one single frame. For the training of the virtual view, we render the full image of the frame with down-sampled resolution. The resolution is set to be around 64×64 in the warm-up stage and 128×128 in the second stage to fit our 24G GPU memory. NeRFAcc [4] is used to speed up the training via efficient sampling. The resolution of the occupancy grid is set to be 128, and the render step size is set to be 0.01.

In order to achieve more robust optimization and speed up the convergence, we adopt a coarse-to-fine training strategy, where a modulation ratio term is used to control the bandwidth of the Hash grid and coordinate encoding λ^b:

$$\lambda^b = \min\left(0.25 + \frac{E}{E_{\max}}, 1.0\right) \cdot \lambda^b_{\max}. \tag{3}$$

We use Zero-1-to-3 [7] as our diffusion prior. The guidance scale is set to 5.0. The time-step range is $[0.02, 0.5]$ in the warm-up stage and $[0.02, 0.2]$ in the second stage.

Method	Metric	AMA		BANMo		iPhone	Avg.
		Samba	Swing	eagle1	eagle2	Teddy	
NDR	**mPSNR.**↑	7.97	9.89	14.29	14.80	**9.26**	11.24
	mSSIM↑	0.326	0.397	0.241	0.263	**0.254**	0.296
	mLPIPS↓	0.457	0.463	0.514	0.497	0.442	0.475
Ours	**mPSNR.**↑	**10.73**	**11.35**	**15.37**	**16.93**	9.02	**12.68**
	mSSIM↑	**0.493**	**0.510**	**0.269**	**0.319**	0.239	**0.366**
	mLPIPS↓	**0.328**	**0.354**	**0.507**	**0.447**	**0.360**	**0.399**

Table 1. **Per-scene quantitative results on novel view synthesis.** For synthetic datasets, we compare 8 GT views that span 360 degrees. For real-world dataset, we compare 2 GT views provided. Note that Teddy is the only sequence among all real-world scenes used in this paper that has additional GT views for evaluation. Due to the data preprocessing and the optimization of the camera pose, the non-semantic metrics here may not accurately reflect the actual performance.

2. Additional Analysis

2.1. Novel view synthesis

We show additional quantitative and qualitative results in Tab 1 and Fig. 2 respectively. For quantitative results on real-world datasets, Teddy is the only sequence that has GT reference views for evaluation. Since our training data is monocular RGB-D data, there is only one viewpoint for each timestamp. This makes novel view synthesis in this problem setting quite challenging. From the results, we could find that our method produces competitive results on novel view synthesis with consistently better perceptual quality thanks to the use of diffusion prior. Note that due to data preprocessing and the optimization of the camera pose, the non-semantic metrics, PSNR, and SSIM, may not reflect the perceptual quality. A small shift in camera poses will significantly affect those metrics, especially given the fact that we need to estimate the masked PSNR and SSIM, i.e., mPSNR and mSSIM.

2.2. Choices of Different Diffusion Priors

We present a comparative analysis of the generation quality of various diffusion priors in Fig. 3. Point-E [8] is a diffusion model based on point clouds with feed-forward generation capability. However, its generation may exhibit limitations in accurately fitting the original observations and

Figure 2. **Qualitative results on novel view synthesis.** We evaluate on Teddy scene which is the only sequence among all real-world scenes used in this paper that has additional GT views for evaluation. Thanks to the use of diffusion priors, our method can achieve high-quality novel synthesis given a monocular RGB-D video.

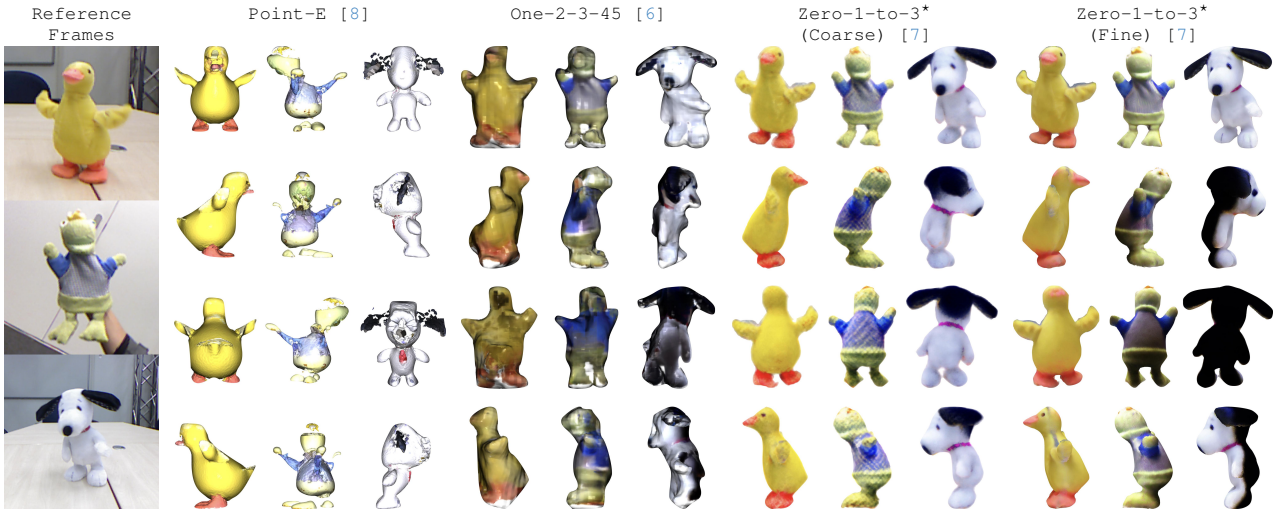

Figure 3. **Comparison of different diffusion priors.** We mainly compare: 1) Point-E [8]: Point cloud-based diffusion model, feed-forward generation 2) One-2-3-45: Generalizable neural surface reconstruction with Zero-1-to-3 [7], feed-forward generation 3) Zero-1-to-3⋆ [7]: we use Stable-Dreamfusion [10] repository to perform image-to-3D with SDS from Zero-1-to-3, denoted as Zero-1-to-3⋆ [7]. The coarse stage uses NeRF with Zero-1-to-3 [7] for optimization. The fine stage uses DMTet [9] with Zero-1-to-3 [7] for optimization.

struggle with uncommon objects (e.g. the Frog Prince toy). One-2-3-45 [6] is another feed-forward generation model, which integrates generalizable neural surface reconstruction with Zero-1-to-3 [7], achieving high-quality image-to-3D generation on synthetic images with remarkable speed. Nevertheless, real-world images often introduce challenges

Figure 4. **Failure case analysis.** We showcase the limitations of MorpheuS. The inherent scale ambiguity and challenging real-world scenarios like motion blur and complicated target object pose can hinder the performance of RGB-based diffusion models such as Zero-1-to-3⋆[7], causing undesired artifacts like the Janus effect. MorpheuS inherits the same limitations but could achieve better results thanks to the leverage of temporal information from the video sequence and regularization on the canonical shape.

such as high-frequency noise and diverse illumination conditions. One-2-3-45 can fail in these real-world scenarios, and result in artifacts like shadows, inconsistent geometry, missing details, etc. Zero-1-to-3⋆ [10] is another line of work that performs SDS using the Zero-1-to-3 [7]. The test-time optimization can effectively get rid of the inconsistent prediction generated by Zero-1-to-3 and result in a coherent geometry (See Fig. 3). Thus, we prefer knowledge distillation from Zero-1-to-3 [7] over other feed-forward generation models.

2.3. Canonical Space Regularization

We provide more details about the ablation experiments on canonical space regularization. Recall the points used for canonical space regularization in Eq. 12 of our main paper:

$$\mathbf{x}'_{\text{reg}} = \{\mathbf{x}_t, T(\phi(\mathbf{x}_t), \mathcal{V}_t(t))\}. \tag{4}$$

Note that $\mathbf{x}_t$ are sampled directly in the observation space with the deformation network being shortcutted. To encourage the local smoothness of the SDF gradient, a small perturbation $\delta\mathbf{x}_t$ is applied to the $\mathbf{x}_t$:

$$\tilde{\mathbf{x}}'_{\text{reg}} = \{\mathbf{x}_t + \delta\mathbf{x}_t, T(\phi(\mathbf{x}_t + \delta\mathbf{x}_t), \mathcal{V}_t(t))\}. \tag{5}$$

The difference between the gradient of those sets of points $\left\|\nabla_s(\mathbf{x}'_{\text{reg}}) - \nabla_s(\tilde{\mathbf{x}}'_{\text{reg}})\right\|^2$ is computed as the regularization loss $\mathcal{L}_{cano}$. This loss can effectively constrain the hyper-dimensional canonical field and prevent trivial or ambiguous solutions (e.g. thin geometry with texture carved in it).

In the ablation experiments, we also experiment with other two variants, both of which involve the use of the deformation network. As opposed to Eq. 12 of our main paper, the points used for regularization:

$$\mathbf{x}' = \{\mathbf{x}_t + D(\phi(\mathbf{x}_t), \mathcal{V}_t(t)), T(\phi(\mathbf{x}_t), \mathcal{V}_t(t))\}, \tag{6}$$

are sampled in the observation space and then deformed to the canonical space. For the perturbation vector, we experimented with *obs. perturb.*: applying the perturbation to the points in the observation space $\mathbf{x}_t$ before deforming to the canonical space, and *cano. perturb.*: applying the perturbation to the deformed points in the canonical space $\mathbf{x}_t + D(\phi(\mathbf{x}_t), \mathcal{V}_t(t))$ and $T(\phi(\mathbf{x}_t), \mathcal{V}_t(t))$. We find that performing regularization in both ways can lead to over-smooth geometry and trivial solutions (See Fig. 7 in the main paper).

2.4. Failure Cases Analysis

We further analyze the failure cases of MorpheuS and the challenges for RGB-based diffusion models. We show our result on the challenging *haru* sequence from the iPhone dataset. Zero-1-to-3 [7] is trained on each individual reference frame for a reference. The results are shown in Fig. 4.

Real-world video captures often have challenging scenarios such as motion blur and complicated articulated poses of the target object (See the images of a dog in Fig. 4). These challenges coupled with the inherent scale ambiguity of RGB observations can hinder accurate shape fitting in the generation process of RGB-based diffusion models, sometimes resulting in undesired artifacts like the Janus effect (See the head of the dog in the first row of Zero-1-to-3⋆[7] in Fig. 4 and the erroneous beak on the back side of the duck in the third row of Zero-1-to-3⋆ (Fine) in Fig. 3).

Our MorpheuS also relies on an RGB-based diffusion model and thus also inherits the same challenges and difficulties. However, the leverage of temporal information from the entire video sequence and the implicit regularization of the canonical field can allow MorpheuS to alleviate the above-mentioned problems in Zero-1-to-3. For instance, the Janus effect can be eliminated. See the head of the dog in our reconstruction result.

References

[1] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. In *NeurIPS*, 2022. 1

[2] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *CVPR*, 2021. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[4] Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: Efficient sampling accelerates nerfs. *arXiv preprint arXiv:2305.04966*, 2023. 2

[5] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *WACV*, 2022. 1

[6] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In *NeurIPS*, 2023. 3

[7] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023. 1, 2, 3, 4

[8] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2, 3

[9] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *NeurIPS*, 2021. 3

[10] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion. 3, 4

[11] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *CVPR*, 2023. 2

[12] Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE TPAMI*, 2021. 1