Supplementary Material
Dream-to-Recon: Monocular 3D Reconstruction with Diffusion-Depth Distillation from Single Images

Philipp Wulff[1] Felix Wimbauer[1,2] Dominik Muhle[1,2,3] Daniel Cremers[1,2]

[1]Technical University of Munich [2]MCML [3]SE3 Labs

philippwulff.github.io/dream-to-recon

In this supplementary material, we present additional implementation details, extended results, and ablation studies.

A. Video Results

We include video results showcasing multiple sequences from both KITTI-360 and Waymo, available on our project website. The videos illustrate per-frame volumetric reconstructions for our method alongside several baseline approaches.

B. Implementation Details

Model training. Our view completion model is built upon Stable-Diffusion-2.1-unCLIP, employing the standard ControlNet architecture for conditioning. For training data synthesis, we render novel views at a resolution of 192×360, followed by re-rendering the input views at 192×288, which are then upscaled to 512×768 to approximate the resolution used during the original training of Stable-Diffusion. The input image is derived from the re-rendered view. We use DDIM sampling during training and evaluate the model using UniPCMultiStep sampling with five steps. The training is initialized from the official Stable-Diffusion weights and runs for 20 epochs with a batch size of 20 and a constant learning rate of 10^{-5}. During ControlNet training, UniDepth is used for depth prediction. However, subsequent experiments showed that Metric3D yields more accurate predictions than UniDepth; therefore, we adopt Metric3D for training the scene reconstruction model model. To construct the pseudo-volume, we employ 48 coarse samples, 16 fine samples, and 16 ray samples placed within $\sigma = 2$ of the estimated depth.

The scene reconstruction model architecture utilizes the ResNet50-based U-Net backbone from Behind the Scenes. To ensure stability in mixed-precision training, batch normalization layers are added to the decoder portion of the backbone. The backbone outputs a frustum-aligned density grid of dimensions $192 \times 640 \times c$, where $c \in 64, 128$, with an inverse growth pattern along the z-axis. The resulting occupancy field has a resolution of $Z = 64$, inversely spaced between depths of $3m$ and $50m$.

Training is conducted using the UniPCMultiStepScheduler, with automatic mixed precision (AMP), caching, and batch size increases beginning from the second epoch.

Ground truth for evaluation. For each timestamp, a 3D occupancy grid is generated by removing all voxels that contain points from 300 consecutive LiDAR sweeps. The overall reconstruction accuracy (O_{acc}) is computed within a cuboid defined by $x = [-4m, 4m]$, $y = [-0.75m, 0m]$, and $z = [4m, 20m]$ in the input camera's coordinate frame. Given the higher frequency of dynamic objects in the Waymo dataset, we accumulate 300 LiDAR sweeps and retain occupancy only within the ground truth 3D bounding boxes. For Waymo, evaluation is performed on the validation set, as 3D bounding box annotations are not available in the test set.

Additionally, the metrics IEacc and IErec quantify the reconstruction's accuracy and recall, respectively. These metrics are particularly important as they measure reconstruction quality *beyond* what a monocular depth estimation (MDE) model can typically achieve.

C. Comparison With Render-Refine-Repeat Methods

Depth-based warping and inpainting are also used in other render-refine-repeat (RRR) works. These approaches generally tackle the Text-to-3D task and focus exclusively on novel view synthesis. We compare against two SOTA methods `LucidDreamer` and `RealmDreamer` (concurrent work), which use a comparable StableDiffusion model as us, and extract the geometry of the scene. For fairness, we test images from their papers and from Waymo, as seen in Fig. 1. There are to two main differences: **1)** Due to the focus on NVS, the resulting geometry is subpar, even for

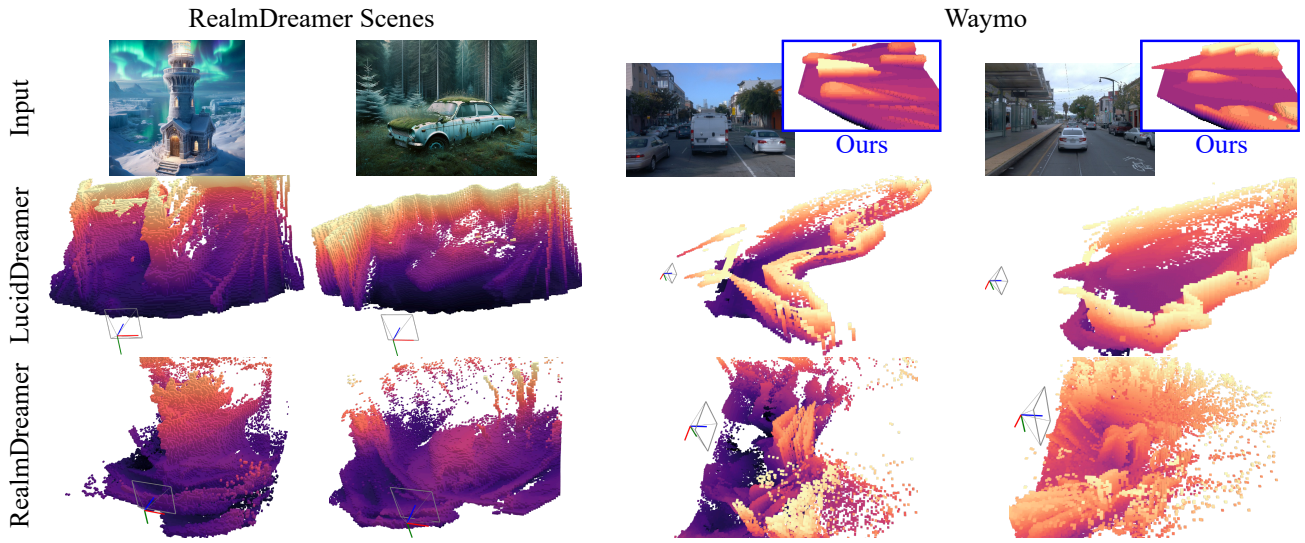

Figure 1. **Render-refine-repeat Comparison.** Occupancy reconstruction for different RRR methods against ours (marked blue).

Depth gradients	Optical flow	PSNR ↑	Abs Rel ↓	RMSE ↓
✓	✗	**20.920**	**0.133**	**0.093**
✗	✓	18.707	0.179	0.121
✓	✓	18.933	0.179	0.118

Table 1. **Qualitative occlusion detection strategies.** Strategies during inference in the novel view. at 192×640 resolution

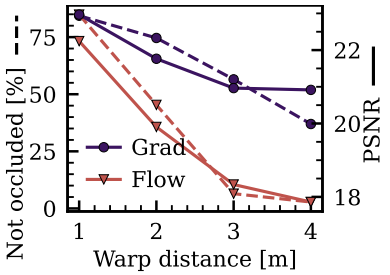

Figure 2. **VCM.** Gradient vs. flow-based occl. detection against warp distance.

example images from their papers. Here, our *synthetic occupancy field* comes into play. **2)** Because of the focus on Text-to-3D rather than Image-to-3D, existing RRR methods work well for images generated using the diffusion model itself (often in cartoon or fantasy-style). Here, the diffusion model can inpaint effectively even without finetuning. However, when using real images, the inpainting produces poor results. Our novel *VCM* with the corresponding warp-based training solves this issue.

D. Additional Experiments

Quantitative effect of occlusions. We measure the effect of the occlusion maps for different methods also quantitatively in Tab. 1. As observed in the main paper, the depth gradient- based approach performs best.

Effect of Warp Distance and Occlusion Strategy on VCM. As seen in Fig. 2, our gradient-based occlusion detection strategy provides more precise masks ($\rightarrow$ more visual context) than the flow-based baseline. Thus, the VCM produces better synthetic views. When performing larger viewpoint changes, the quality degrades slightly. In our experiments, novel view poses are on average $\sim 3\,\mathrm{m}$ away from the source view(s).

Statistical Importance. We report the mean and variance where possible in Tab. 2 *a)*. The variance remains within limits, meaning that the final results are reliable.

Impact of supervision via the SOF. To test the effect of our geometry synthesis independently of the VCM, we build the synthetic occupancy field directly from real multi-view data (the same as in BTS) and report the results in Tab. 2 *b)*. While the resulting geometry is better than the one produced by BTS, it is still inferior to the geometry resulting from the synthetic views produced by the VCM. This suggests that both the VCM and our novel way of obtaining scene geometry contribute to our strong performance.

Inference Latency. While the SOF geometry is of high-

	Method	O_{acc} ↑	IE_{acc} ↑	IE_{rec} ↑	O_{acc} ↑	IE_{acc} ↑	IE_{rec} ↑
a)	BTS	$92_{0.64}$	$69_{4.24}$	$64_{8.37}$	$95_{0.74}$	$63_{10.71}$	$94_{1.88}$
	BTS-D	$92_{0.70}$	$70_{3.96}$	$66_{8.24}$	$95_{0.77}$	$61_{11.66}$	$96_{1.11}$
	Ours	$93_{0.49}$	$72_{3.51}$	$74_{6.67}$	$97_{0.35}$	$73_{6.82}$	$96_{1.35}$
	Ours (Distilled)	$90_{0.86}$	$71_{3.84}$	$71_{9.00}$	$96_{0.43}$	$72_{7.10}$	$93_{2.69}$
b)	**Ours**	$93_{0.49}$	$72_{3.51}$	$74_{6.67}$	$97_{0.35}$	$73_{6.82}$	$96_{1.35}$
	Ours w/ real MV	$93_{0.50}$	$71_{3.91}$	$66_{7.62}$	$97_{0.25}$	$71_{7.07}$	$91_{3.25}$

Table 2. **Scene Reconstruction.** *a)* Results on KITTI-360 / Waymo in % with variances. *b)* Comparing synthesized views vs. real multi-view data for building the synthetic occupancy field.

quality, the generation process is costly at $5303\frac{\text{ms}}{\text{frame}}$ on an A100 GPU. The distilled scene reconstruction model only takes about $75\frac{\text{ms}}{\text{frame}}$ (>70x faster) and is real-time capable. It also produces more stable results, while the original synthesized geometry sometimes contains artifacts. Both aspects are crucial for the applications we target (autonomous driving, robotics, etc.).