Supplementary Material for *CloseUpShot*

Yuqi Zhang, Guanying Chen, Jiaxing Chen, Chuanyu Fu,
Chuan Huang, Shuguang Cui,

Contents

1. More Details

1.1. Occlusion-aware Noise Suppression

Algorithm 1 presents the details of the depth dilation adopted in the proposed Occlusion-aware Noise Suppression. Figure S1 shows visualization examples of the depth dilation process. Owing to the sparsity of the point cloud, the warped RGB and depth images are often sparse, and background points may leak through foreground gaps, resulting in unreliable conditioning. To address this, we dynamically dilate the depth map based on image density, filtering out such interfering points, and these regions are then compensated by the low-resolution warping results, producing dense conditioning. It is worth noting that most edge regions are also filtered out. We consider this reasonable, since the depth predictions from the pretrained estimator are relatively unreliable near object boundaries.

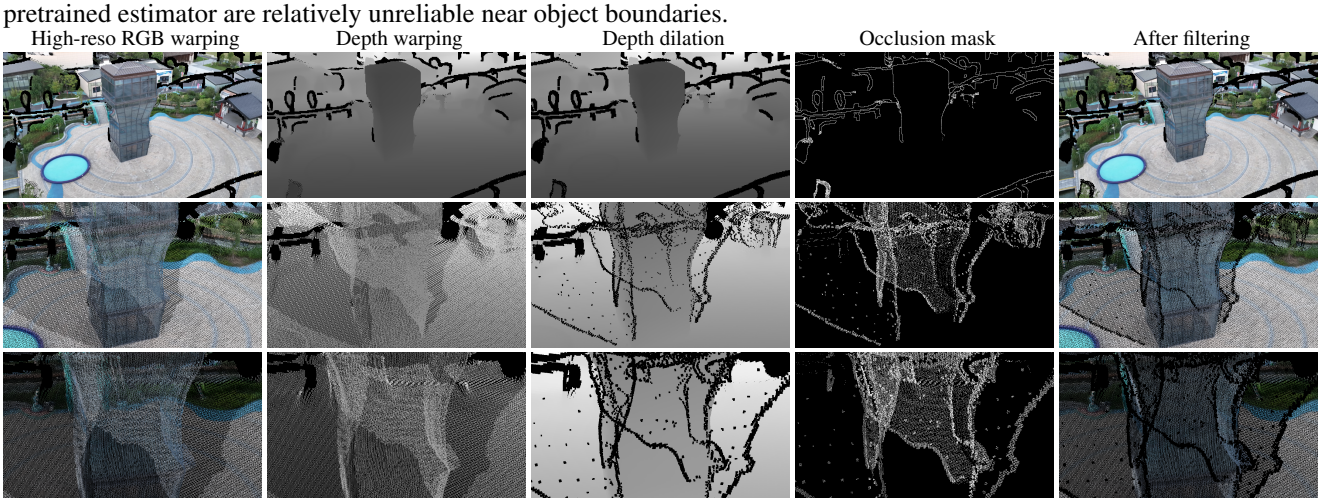


Figure S1. Visualization of the occlusion-aware noise suppression.

Algorithm 1 Occlusion-aware Depth Dilation

Input: Warped depth map D_{warp}
Output: Updated depth map D_{out}
Other parameters: M denotes the valid mask of D_{warp}; $\mathbf{p}$ denotes a pixel in D_{warp}; τ_D denotes the depth threshold; $k_{\max}$ and $k_{\min}$ denote the maximum and minimum kernel sizes.

Compute valid-pixel density: $\rho = \frac{1}{HW} \sum_{\mathbf{p}} M(\mathbf{p})$
Determine dilation kernel size: $k(\rho) = k_{\max} - (k_{\max} - k_{\min})\rho$
Mask invalid pixels with $+\infty$:

$$\hat{D}(\mathbf{p}) = \begin{cases} D_{\text{warp}}(\mathbf{p}), & M(\mathbf{p}) = 1 \\ +\infty, & M(\mathbf{p}) = 0 \end{cases}$$

Apply min-pooling with kernel size $k(\rho)$: $D_{\text{dilate}} = -\operatorname{MaxPool}\big(-\hat{D};\, k(\rho)\big)$
Identify invalid or penetrated pixels: $\mathcal{U} = \{\mathbf{p} \mid M(\mathbf{p}) = 0 \quad \vee \quad (D_{\text{warp}}(\mathbf{p}) - D_{\text{dilate}}(\mathbf{p}) > \tau_D)\}$
Update depth map:

$$D_{\text{out}}(\mathbf{p}) = \begin{cases} D_{\text{dilate}}(\mathbf{p}), & \mathbf{p} \in \mathcal{U} \\ D_{\text{warp}}(\mathbf{p}), & \text{otherwise} \end{cases}$$

return D_{out}

1.2. The Easy and Hard Set Division on DL3DV-Benchmark

We divide the DL3DV-Benchmark into easy and hard subsets following the protocol of ViewCrafter, which separates them based on the frame sampling stride. Building upon the implementation of DepthSplat, we adopt two configurations by setting the interval between the first and last video frames to 50 and 100, respectively. A larger frame sampling stride corresponds to faster camera motion and greater viewpoint changes, which we therefore refer to as the hard setting. Figure S2 illustrates an example of the easy and hard splits for a scene. The first row corresponds to the easy set, and the second row corresponds to the hard set. The leftmost and rightmost images denote the reference views, while the middle images show the point cloud projections for regular novel view synthesis.

Figure S2. An example of easy and hard sets of DL3DV-Benchmark. The first row corresponds to the easy set, and the second row corresponds to the hard set. The leftmost and rightmost images denote the reference views, while the middle images show the point cloud projections for regular novel view synthesis.

1.3. Computational Overhead

Table S1 summarizes the time and GPU memory costs for each inference stage, measure on a single NVIDIA A100 GPU with 80 GB of memory. Our model generates 25 images in a single inference pass and consists of several stages. First, a DDIM sampling is performed to generate multi-view images. Next, these images are fused using global consistency fusion

to obtain the global structure context. This global structure context is then used to guide the diffusion model for a second inference, producing the final diffusion outputs. Finally, 3DGS optimization is applied for 3D reconstruction.

Table S1. Computational overhead of each inference stage.

Stage	Inference stage 1	Global structure fusion	Inference stage 2	3DGS optimization	Total
Processing time (seconds)	121	2	126	30	279
Peak GPU memory (G)	43.2	-	45.2	2.1	-

2. More Ablation Studies

2.1. Effect of Decoder Finetune

Table S2 presents ablation results: the first two columns show the performance of combining the baseline with decoder finetuning, while the last two columns report our method with and without decoder finetuning. From these comparisons, we observe that decoder finetuning consistently improves PSNR by about 1 dB. However, this improvement mainly comes from better texture learning, and it cannot fundamentally correct structural errors. As illustrated in Figure S3, the baseline method suffers from structural mistakes (e.g., errors at the top of buildings or in the faces of sculptures), whereas our method generates more plausible results, with or without decoder finetuning. This further validates the effectiveness of our proposed module. Therefore, the role of decoder finetuning is to enhance texture quality rather than rectify structural inaccuracies.

Table S2. The effect of decoder finetune.

Method	PSNR↑	SSIM↑	LPIPS↓
Baseline	17.60	0.555	0.413
+ Decoder finetune	18.69	0.600	0.379
Ours w/o Decoder finetune	18.75	0.618	0.369
+ Decoder finetune	**19.66**	**0.643**	**0.346**

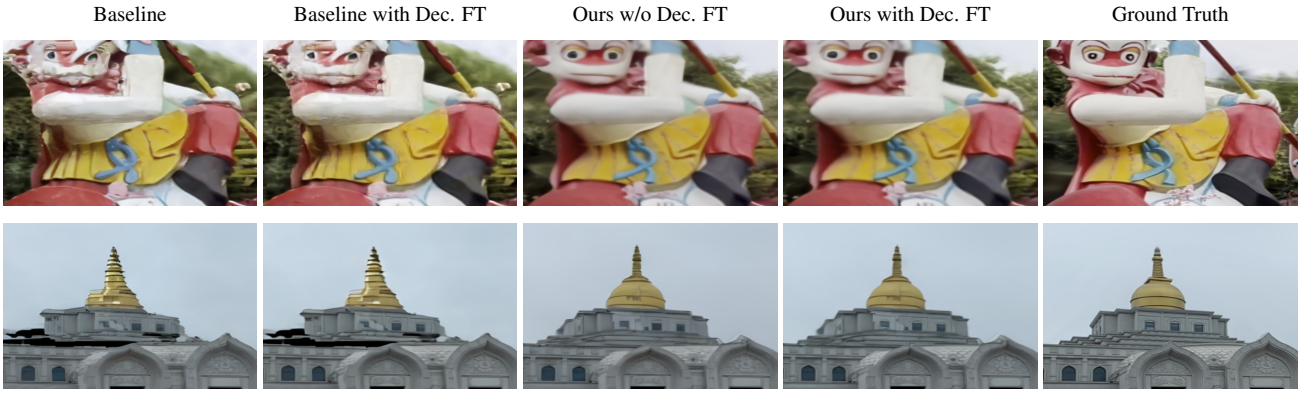


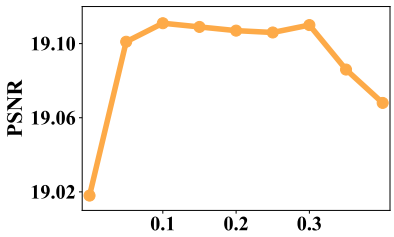
Figure S3. The effect of decoder finetune (Dec. FT).

2.2. Chosen of Hyper-parameters

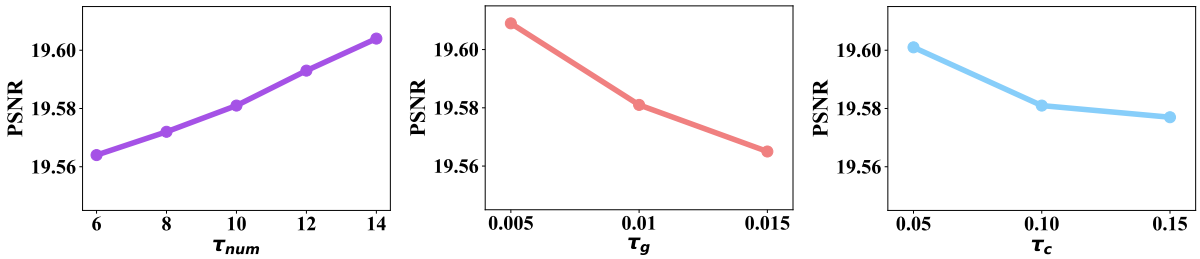
We set $\tau_D = 0.2$ by default in all the experiments. Figure S4 (a) shows **the effect of τ_D in the proposed occlusion-aware noise suppression**. A smaller value means more pixels will be filtered out, which may include some useful ones, while a larger value fails to effectively remove noisy pixels.

In addition, we conduct ablation studies on the three hyperparameters, $\tau_{num}, \tau_g, \tau_c$, used in the **global structure guidance** module. Since our video diffusion model generates 25 frames, we set $\tau_{num} = 10$ by default. The other two parameters, τ_g and τ_c, are set to 0.01 and 0.1, respectively. As shown in Fig. S4 (b), a larger τ_{num} indicates a stricter consistency constraint, while smaller values of τ_g and τ_c enforce stricter evaluations, leading to better results. More importantly, varying these

hyper-parameters has only a minor impact on the quality of diffusion-generated results (with a maximum PSNR difference of about 0.04), demonstrating the robustness of our method.



(a) Effect of threshold τ_D in noise suppression



(b) Effect of hyper-parameteres used in global structure guidance

Figure S4. Effect of Hyper-parameteres. (a) shows effect of the threshold τ_D in noise suppression, while (b) presents the effect of τ_{num}, τ_g, τ_c used in global structure guidance.

2.3. Soft Pixel Confidence in 3DGS Optimization

We further conduct an additional experiment on pixel confidence, comparing the soft confidence approach with the hard confidence used in the main paper. Specifically, we compute the pixel-level confidence as the ratio of valid correspondences (passing the consistency check) over the total correspondences. As shown in the Table S3, the difference between the two confidence strategies is marginal. This is likely because the regions with the largest differences between the hard and soft schemes, typically areas with low overlap or along object boundaries, are rarely observed during 3DGS optimization and are inherently less reliable.

Table S3. Ablation of pixel confidence in 3DGS optimization.

Method	Close-up View			Original View		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Pixel confidence (Hard)	19.80	**0.637**	**0.391**	20.33	**0.656**	**0.371**
Pixel confidence (Soft)	**19.81**	0.636	**0.391**	**20.34**	**0.656**	0.372

2.4. Soft Number Threshold in Global Structure Guidance

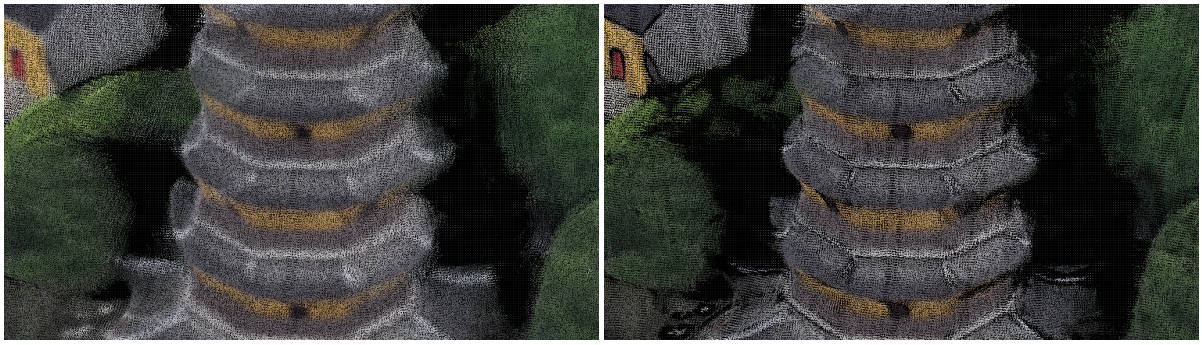
To further evaluate the impact of the number threshold, we introduce a soft threshold based on the ratio of valid correspondences over the total correspondences, where the soft threshold is set to $\tau_{num} > 0.5$, instead of the previous hard threshold setting ($\tau_{num} > 10$) in the main paper. The average results of the two settings are reported in Table S4, showing that the performance difference is not significant.

Table S4. Ablation of number threshold in global structure guidance.

Setting	Close-up View			Original View		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Hard threshold	**19.61**	**0.608**	**0.358**	**19.93**	0.602	0.360
Soft threshold	19.57	**0.608**	**0.358**	19.91	**0.605**	**0.359**

2.5. Effect of Color Consistency

Figure S5 presents a visualization example illustrating the effect of color consistency. As shown in the figure, with the color consistency constraint, the fused point cloud exhibits improved texture quality and produces sharper, clearer images.

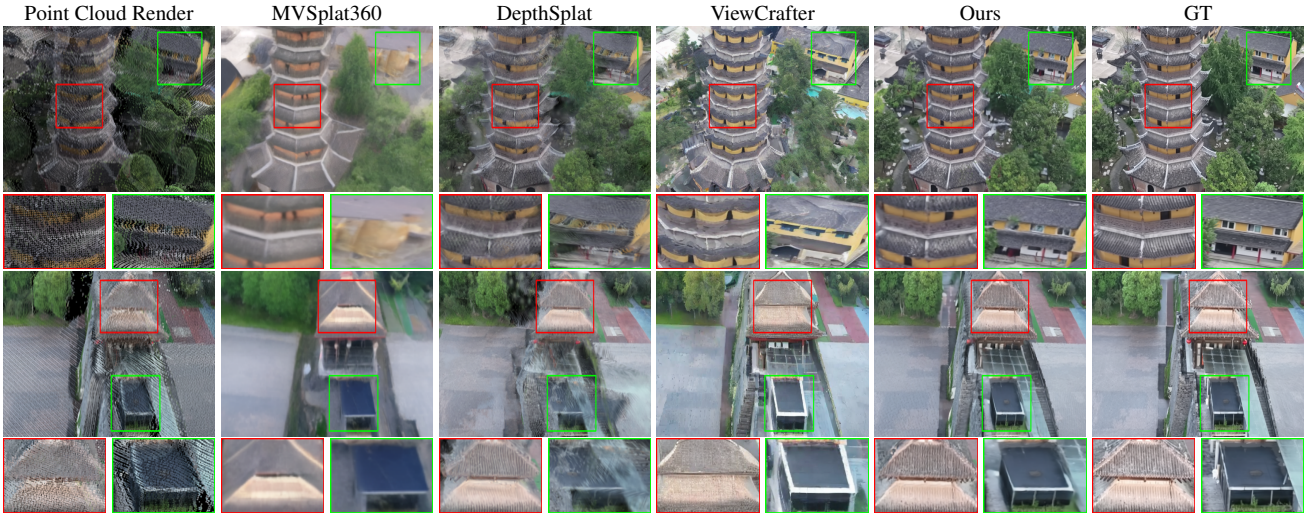


Without color consistency With color consistency

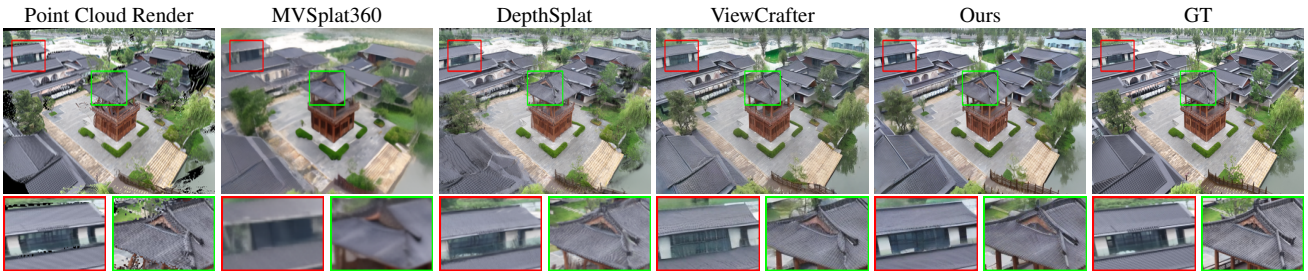
Figure S5. Effect of color consistency in the global structure guidance.

3. Qualitative Results on the DL3DV-Drone Dataset

Figure S6 present the qalitative comparisons on the DL3DV-Drone dataset, which shows that our reconstructions exhibit higher fidelity to the ground truth against other method, particularly in fine-grained details. For instance, our method better preserves the textures on the tower in the close-up view and the window appearance of the courtyard scene in regular view.



(a) Close-up View



(b) Original View

Figure S6. Qualitative results of sparse-view novel view synthesis on the DL3DV-Drone dataset. We present the results of (a) close-up view and (b) regular view synthesis, respectively. The color boxes highlight the difference among the methods for better comparison.