Robust Gaussian Surface Reconstruction with Semantic Aware Progressive Propagation

Yusen Wang
wangyusen@whu.edu.cn
Wuhan University
School of Computer Science
Wuhan, Hubei, China

Huan Zhou
2023202210017@whu.edu.cn
Wuhan University
School of Computer Science
Wuhan, Hubei, China

Yu Jiang
jiangyu1181@whu.edu.cn
Wuhan University
School of Computer Science
Wuhan, Hubei, China

Chunxia Xiao*
cxxiao@whu.edu.cn
Wuhan University
School of Computer Science
Wuhan, Hubei, China

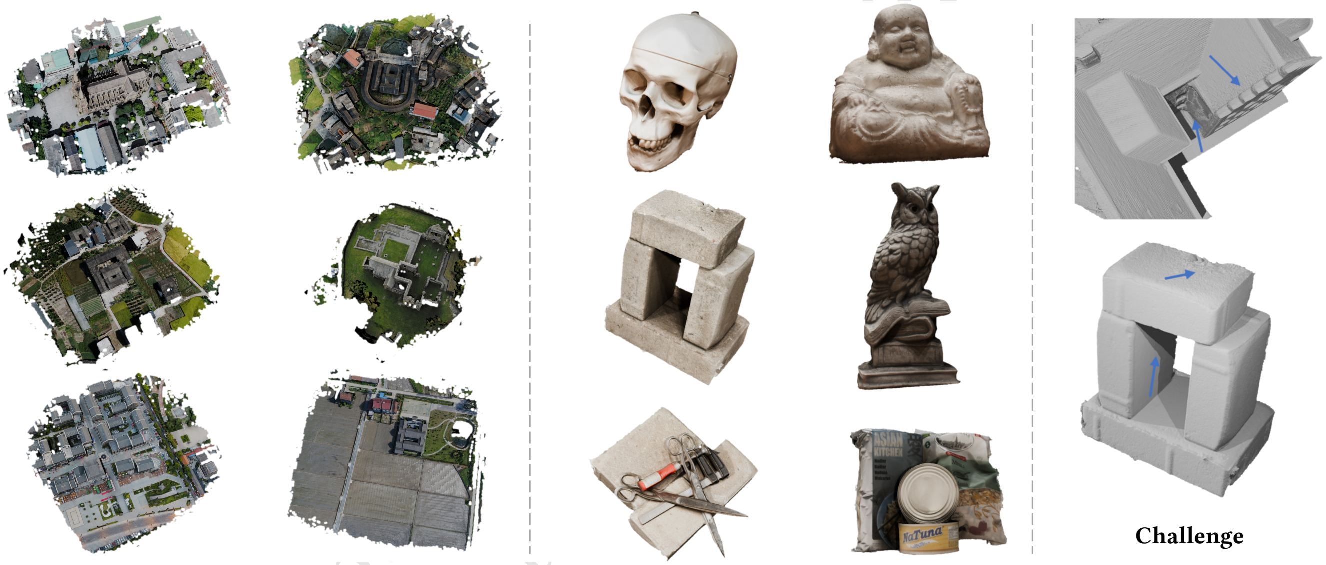

Figure 1: GSAPro exhibits excellent robustness for scenes of various scales. In some challenging positions, as indicated by the blue arrows in the figure, GSAPro can achieve better reconstruction accuracy than the SOTA methods.

Abstract

We propose GSAPro, a Gaussian Splatting based 3D surface reconstruction framework that exhibits robustness across diverse scales of scenes. Previous research has leveraged photometric consistency constraints or prior information as guidance to enhance the reconstruction accuracy. However, error estimation and noise inevitably exist in these priors. Applying a strict geometric filter removes a large amount of reliable information, resulting in a deterioration of the quality of guided reconstruction. Regarding possible errors in the initial guidance, GSAPro can continuously improve the accuracy of the guidance through a joint optimization strategy. The Gaussian Branch integrates reliable geometric and color constraints, thus providing more accurate geometric parameters for the Prior Branch compared to its current state guidance parameters. The Prior Branch, through photometric selection and propagation, obtains more accurate geometric parameters from the state geometric parameters and rendered parameters. Then GSAPro uses these parameters to guide the optimization of the Gaussian Branch. Regarding the problem of noise existing in the guidance, we train the Semantic Aware Module to predict the noise by utilizing the image information, thus improving the accuracy. Moreover, we also introduce a Distillation Module to mitigate the excessive splitting of Gaussians that is caused by the implementation of additional

*Corresponding author

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755649

constraints. Experiments demonstrate that our method exhibits SOTA performance and has stronger robustness against scenes of different scales.

CCS Concepts

• **Computing methodologies → Reconstruction.**

Keywords

3D Reconstruction, Gaussian Splatting

ACM Reference Format:
Yusen Wang, Huan Zhou, Yu Jiang, and Chunxia Xiao. 2025. Robust Gaussian Surface Reconstruction with Semantic Aware Progressive Propagation. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3755649

1 Introduction

Recently, 3DGS-based surface reconstruction [12, 14, 45] has gradually become a research hotspot due to its faster training speed and realistic rendering compared with Nerf-based methods [9, 32, 40]. The most advanced Gaussian Surface Reconstruction (GSR) methods flatten the Gaussians [12, 14] and introduce multiview photometric consistency [2, 6] to improve the reconstruction accuracy, which achieves extremely high accuracy on the small object [15] dataset. When generalizing them to more complex scenes [39], where the camera pose arrangements are disordered and the scene sizes vary significantly, the reconstruction quality drops noticeably.

Using the photometric consistency loss based on Normalized Cross Correlation(NCC) cost as a supervision [2, 6, 9] has many issues. First of all, it is not like depth, which is a clear optimization target that allows the depth loss to be made as close to zero as possible. In accurate regions, the NCC cost may be high because of illumination changes, and in noisy regions, it can be extremely low. Especially in the edge and high-frequency regions, the NCC cost usually cannot be used to effectively distinguish adjacent foreground and background pixels. Inaccurate supervision will degrade detail reconstruction accuracy. A high-threshold geometric filter can identify noise but has severe false-positive issues (Figure 2). Mislabeled regions are crucial for improving Gaussians' reconstruction accuracy (Table 3).

Using pre-computed information as priors to guide 3DGS or Nerf optimization is a common approach [5, 13, 33, 44]. It cuts computational load and is highly robust. However, the priors also contain noise, and the estimation fails to effectively utilize the global information that Gaussians can provide for self-improvement. The results reconstructed by these algorithms generally tend to be oversmooth.

Regarding the problem of errors existing in the pre-computed and initial guidance, the guidance used by GSAPro can be self-updated through the interaction between rendering and numerical calculation. Regarding the noise existing in the guidance, GSAPro can accurately identify it. In addition, regarding the problem of excessive splitting caused by the introduction of additional guidance, GSAPro adopts Distillation [8] to suppress it.

Specifically, GSAPro, which is composed of a Gaussian Branch and a Prior Branch, adopts a joint optimization strategy. The Gaussian Branch periodically provides the Prior Branch with the geometric parameters obtained from the rendering of Gaussians. The Prior Branch will compare the geometric parameters in its current state with the new parameters rendered by the Gaussian Branch. It will record the parameters that can minimize the photometric consistency loss among them, and attempt to find better parameters for the surrounding areas through the PatchMatch propagation. Eventually, the optimal parameters are obtained to replace the state parameters of the Prior Branch and guide the optimization of the Gaussian Branch.

Since the Prior Branch adopts the photometric criterion and propagation, a large amount of noise exists in the output optimal parameters. Directly using noisy priors degrades the reconstruction accuracy. To address this issue, we create a dataset named BlendedPM based on the BlendedMVS dataset and train our Semantic Aware Module on it. This module incorporates image semantic information and can mark the noisy regions in the priors output by the Prior Branch more efficiently than the geometric filter. In addition, GSAPro also utilizes the Distillation Module to periodically remove the unimportant Gaussians in the scene, so as to ensure computational efficiency.

In general, our main contributions are as follows:

- The joint optimization strategy, where the Gaussian Branch and the Prior Branch assist each other to improve accuracy and completeness.
- The BlendedPM dataset, which has 61,896 pairs of data.
- The Semantic Aware Module, which can effectively mark the regions that do not conform to the image semantics, has a strong generalization ability across different datasets.

Experiments conducted on DTU and BlendedMVS datasets show that our method has SOTA performance and exhibits stronger robustness to scenes of different scales.

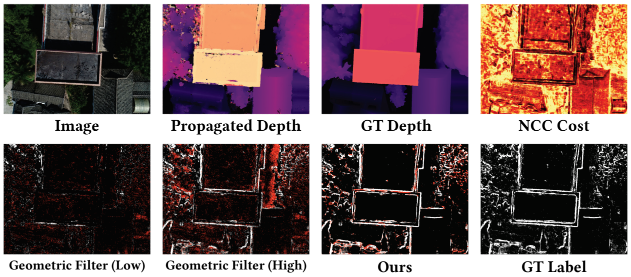

Figure 2: Comparison of Noise Filters. The noise in the propagated depth is shown by the GT label, which can not be filtered by NCC cost. Low-threshold geometric filters struggle to identify these noises, while high-threshold ones cause many false positives. Still, these positions matter for guiding Gaussian optimization. White pixels represent correctly identified positions, while red ones represent incorrectly identified areas.

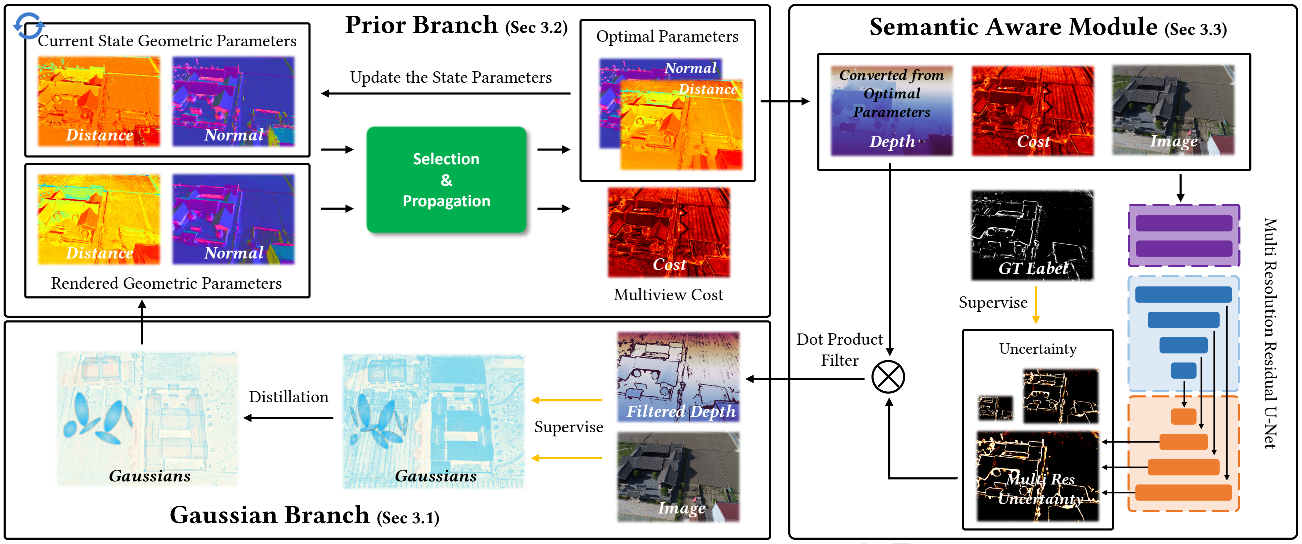

Figure 3: Method overview. The GSAPro adopts the joint optimization strategy. The Prior Branch maintains a set of state parameters during the optimization, and the accuracy of these parameters can be self-improved. The Prior Branch obtains more accurate optimal geometric parameters from the rendering results of the Gaussian Branch and its own state parameters through selection and propagation based on photometric consistency, and then uses these more accurate parameters as the new state parameters. Since propagation is used in the Prior Branch, a large amount of noise inevitably exists. We construct the BlendedPM dataset and train our Semantic Aware Module on it, whose core is to use the Multi-resolution Residual U-net to predict the locations of noises. Finally, the denoised reliable depth will be used to guide the optimization of the Gaussian Branch. In addition, we also introduce the Distillation module to alleviate the excessive splitting of Gaussians caused by the introduction of additional constraints.

2 Related Work

2.1 Multi View Consistency

The Multi-View Stereo (MVS) methods [10, 21, 29, 30, 38] based on feature matching have the characteristics of high robustness, fast reconstruction speed, as well as a complete and stable pipeline. Among them, the PatchMatch-based algorithm [17, 29, 31, 37] estimates a depth map for each view. It is characterized by fast speed and relatively high completeness. Since PatchMatch still uses the relatively local criterion of photometric consistency, its performance will degrade in scenarios such as lighting changes, shadows, non-Lambertian objects, and weak texture regions. Therefore, both traditional methods and deep learning-based methods [1, 11, 20, 36, 47] attempt to increase the receptive field to improve matching performance.

The PatchMatch-based method [29, 37] outperforms the 3DGS-based method in terms of the speed of acquiring geometric priors. However, in terms of the completeness of the scene and the accuracy of object edge reconstruction, it performs worse than the globally optimized 3DGS-based method. In addition, when using the PatchMatch-based method to guide the GSR, the problem of excessive noise urgently needs to be solved. For the patches corresponding to adjacent foreground and background pixels located at the edges of objects, their Normalized Cross Correlation (NCC) generally exceeds the threshold. This makes it impossible for PatchMatch to distinguish between the foreground and the background through NCC, thus resulting in the emergence of a large amount of noise. The introduction of such inaccurate supervision [34] will lead to the degradation of the reconstruction accuracy.

2.2 Gaussian Surface Reconstruction

Given the similarities between NeRF [24, 25] and 3DGS [16, 41, 43], many ideas from the NeRF field [18, 22, 26, 27, 35] can be easily transplanted into 3DGS to improve the reconstruction accuracy. In terms of Gaussian representation, [3, 23, 42, 48] combine 3DGS with an SDF which is more like using 3DGS to replace the function of Neuralangelo's [19] proposal net to guide the sampling position. However, the computational time is much longer than that of 3DGS. [12, 14] flatten the Gaussians, which ensure the consistency under different viewpoints. These methods inherit the fast training speed of 3DGS and achieve good geometric reconstruction accuracy, gradually becoming the mainstream approach in GSR.

Incorporating geometric constraints into GSR can significantly enhance the reconstruction accuracy. For example, the methods in [2, 6, 7] introduce multiview consistency loss, and have achieved extremely excellent reconstruction results. These algorithms increase the computational load and encounter the degradation of reconstruction quality in complex scenarios [39]. NCC-based photometric consistency can only serve as an evaluation metric and cannot, like depth loss, point to an accurate direction for optimization. Because of illumination changes, the NCC cost in regions with accurate information may be considerably high; in contrast, the NCC cost in noisy regions can be incredibly low. Therefore, there is usually a lot of noise in the NCC cost. Some other methods [5, 44] use additional algorithms to calculate priors, such as depth maps, and then utilize these priors to constrain the optimization. However, since prior information often contains a large amount of noise, it can cause the reconstructed details to become blurred. GaussianPro [4] adopts PatchMatch to densify the Gaussians to improve the rendering. It uses NCC as a criterion, filters out reliable points from the geometric information rendered by GSR, and adds new Gaussians at these positions. However, it lacks continuous geometric loss guidance, all the candidates of PatchMatch are derived from the inaccurate rendering results, and the efficiency of the geometric filter is unstable. Moreover, it also has to face the previously mentioned issues regarding the use of NCC.

3 Method

Section 3.1 describes the Gaussian representation used by GSAPro and the Distillation Module integrated from [8]. Section 3.2 describes how the Prior Branch updates the state parameters. Section 3.3 describes the construction of the BlendedPM dataset and the training process of the Semantic Aware Module.

3.1 Gaussian Branch

The Gaussian Branch adopts the same rasterizer as [2], which uses flattened Gaussians $\{G_i | i = 1, \cdots, N\}$ to represent the scene. For each pixel, the rasterizer uses volumetric alpha blending to combine the alpha-weighted colors obtained from all Gaussians sorted by depth:

$$C = \sum_{i \in N} T_i \alpha_i c_i, \quad T_i = \prod_{j=1}^{i-1}(1-\alpha_j), \tag{1}$$

where c_i is the view-dependent color represented by spherical harmonic coefficients, α_i is calculated by multiplying G_i^{2D} and its opacity o_i. The G_i^{2D} can easily be derived from G_i according to the covariance matrix and the Gaussian center position [16].

In order to provide geometric parameter candidates for the Prior Branch, first of all, we need to render the depth and normal map. The normal map N_{gs} under camera coordinate is obtained through alpha blending:

$$N_{gs} = \sum_{i \in N} R_c^T n_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{2}$$

where n_i is the normal corresponding to the direction of the minimum scale factor of the G_i under the world coordinate system, R_c is the rotation from the camera to the world. Follow the [2], the unbiased depth D_{gs} is derived from the distance map $Dist_{gs}$:

$$Dist_{gs} = \sum_{i \in N} d_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \quad d_i = \left(R_c^T\left(t_{G_i} - t_c\right)\right)^T \left(R_c^T n_i\right), \tag{3}$$

where d_i is the distance from the camera center t_c to the 3D plane parameterized by the normal n_i and center position t_{G_i} of G_i. The depth map D_{gs} can be determined from the $Dist_{gs}$ and N_{gs}:

$$D_{gs}(p) = \frac{Dist_{gs}(p)}{N_{gs}(p)K^{-1}\tilde{p}}, \tag{4}$$

where $p = [u, v]^T$ indicates the pixel position on the image plane, K is the intrinsic matrix, $\tilde{p}$ is the homogeneous representation of p. The rendered geometric parameters $Hypo_{gs}$ that are provided to the Prior Branch as candidates are defined as follows:

$$Hypo_{gs}(p) = [N_{gs}(p), Dist_{gs}(p)]. \tag{5}$$

Distillation Module. The Gaussian Branch will lead to the excessive splitting of the Gaussians in order to fit the additional depth constraints (Table 3). In order to make the training of the Gaussian Branch more efficient, we have also incorporated the simplification strategy from [8] into the Gaussian Branch. After a certain number of iterations, the maximum value of $T_i\alpha_i$ in each view where a specific Gaussian G_i participates in rendering is assigned to G_i as its contribution. Then, we normalize the obtained weights to the range of (0,1) and calculate a Cumulative Distribution Function. Next, we remove the Gaussians that contribute the least in the bottom 1% of the CDF. Experiments show that the 1% of Gaussians with the smallest CDF contributions account for 50% of the total number (Figure 9).

Optimization Loss. The loss function for the Gaussian Branch optimization is:

$$L_{gs} = 0.8 * L_{color} + 0.2 * L_{SSIM} + 100 * L_{scale} + L_{geo}, \tag{6}$$

$$L_{scale} = |\min(s_1, s_2, s_3)|_1, \tag{7}$$

$$L_{geo} = Mean(Tanh(Abs(D_{gs} - D_{pb}^{sem}))), \tag{8}$$

where L_{color} and L_{SSIM} are the L1 and SSIM loss between the rendered image and the GT, L_{scale} is the regularization term used for flattening the Gaussian, L_{geo} is the difference between the filtered depth D_{pb}^{sem} by Semantic Aware Module and the depth D_{gs} rendered by the Gaussian Branch.

3.2 Prior Branch

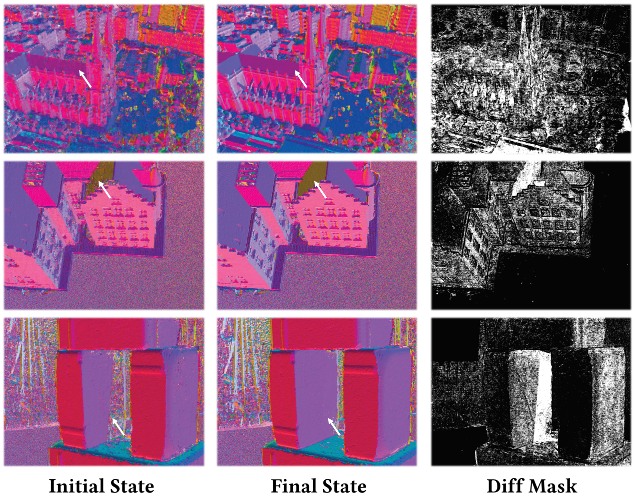

Figure 4: Changes in the state parameters of the Prior Branch. The first two columns display the normal maps of the state parameters in the initial state and at the end of the optimization. The Diff mask marks the regions where the final state parameters are superior to the initial state parameters.

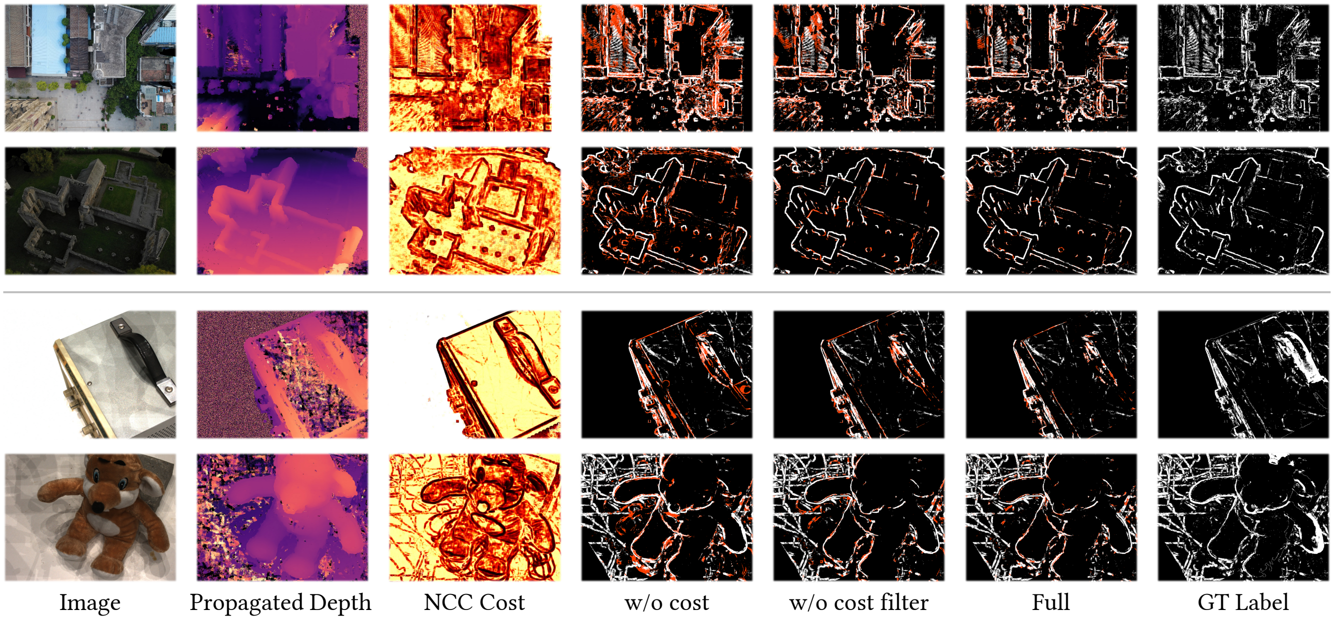

Figure 5: Visualization results of the noise identified by the Semantic Aware Module. The network is trained on the BlendedPM dataset and tested on the BlendedMVS and DTU datasets. White pixels represent correctly identified positions, while red ones represent incorrectly identified areas.

The Prior branch maintains the state parameters $Hypo_{pb}$ throughout the optimization (Figure 4). During the iteration, it continuously selects the optimal solution from the output $Hypo_{gs}$ of the Gaussian Branch and its own state parameters $Hypo_{pb}$ using the NCC cost as the criterion. Then, it attempts to propagate its parameters to the surrounding areas to obtain more accurate geometric information $Hypo_{opt}$. After denoising by the Semantic Aware Module, this information will be used to guide the optimization of the Gaussian Branch.

Before the optimization begins, the state parameters $Hypo_{pb}$ of the Prior Branch are randomly initialized. During optimization, the Prior Branch will continuously update its state parameters $Hypo_{pb}$. First, the Prior Branch needs to select parameters that are superior to the current state $Hypo_{pb}$ from candidates sourced from various ways through the Ncc Cost E_{NCC}. The sources of candidates are as follows: 1. $Hypo_{gs}$ rendered from the Gaussian Branch; 2. the state parameters $Hypo_{pb}$ of the Prior Branch itself; 3. $Hypo_{pertb}$ generated by adding random noise to the state parameters $Hypo_{pb}$. Adding perturbations is a commonly used strategy in MVS to help escape local optima [10, 28, 30, 37].

Next, we use the NCC cost E_{NCC} as a criterion to select a parameter $Hypo_{mid}$ from the candidates that can minimize the E_{NCC}:

$$Hypo_{mid}(p) = \arg\max_{candidate} E_{NCC}(candidate), \tag{9}$$

$$candidate \in \{Hypo_{gs}(p), Hypo_{pb}(p), Hypo_{pertb}(p)\}. \tag{10}$$

Next, we use the PatchMatch technique to perform three iterations on $Hypo_{mid}$, aiming to propagate the reliable regions to the surrounding areas to obtain the optimal parameters $Hypo_{opt}$. The $Hypo_{opt}$ will be used to replace the state parameters $Hypo_{pb}$ of the Prior Branch, and serve as the candidate for the next iteration. Then, $Hypo_{opt}$ is converted into depth D_{pb} according to Eqn 4. D_{pb} together with the corresponding E_{NCC} map is taken as the input of the Semantic Aware Module.

3.3 BlendedPM and Semantic Aware Module

Since the Prior Branch employs the PatchMatch-based technique, it is inevitable that incorrect information will be propagated (Section 2.2). Such errors cannot be identified through photometric consistency. When using a strict geometric filter, the noise at some edge areas still cannot be detected, and a large number of false positives will occur (Figure 2). The geometric information of these false-positive regions is crucial for improving the accuracy of the Gaussian Branch (Table 7). These regions may be wrongly marked as noises due to the relatively strict reprojection error or the small number of times they are observed.

We found that given an image and the output D_{pb} of the Prior Branch, the human can easily identify which regions are incorrect. Therefore, we attempt to use a neural network to identify the regions in the D_{pb} that do not conform to the image semantics.

To this end, we created the BlendedPM dataset based on the BlendedMVS [39] dataset. We utilized the PatchMatch and Propagation module in the Prior Branch to process each BlendedMVS image, thereby generating the corresponding D_{pm} and E_{NCC} cost map $Cost_{pm}$. It contains 61,896 groups of data. Each group contains the depth map D_{pm} after 2 to 5 PatchMatch iterations with a randomly initialized $Hypo$, the corresponding cost map $Cost_{pm}$, the corresponding GT uncertainty label $Label_{pm}$, the corresponding RGB image I and the GT depth D_{gt} from BlendedMVS. We generate the GT uncertainty labels $Label_{pm}$ based on the differences between

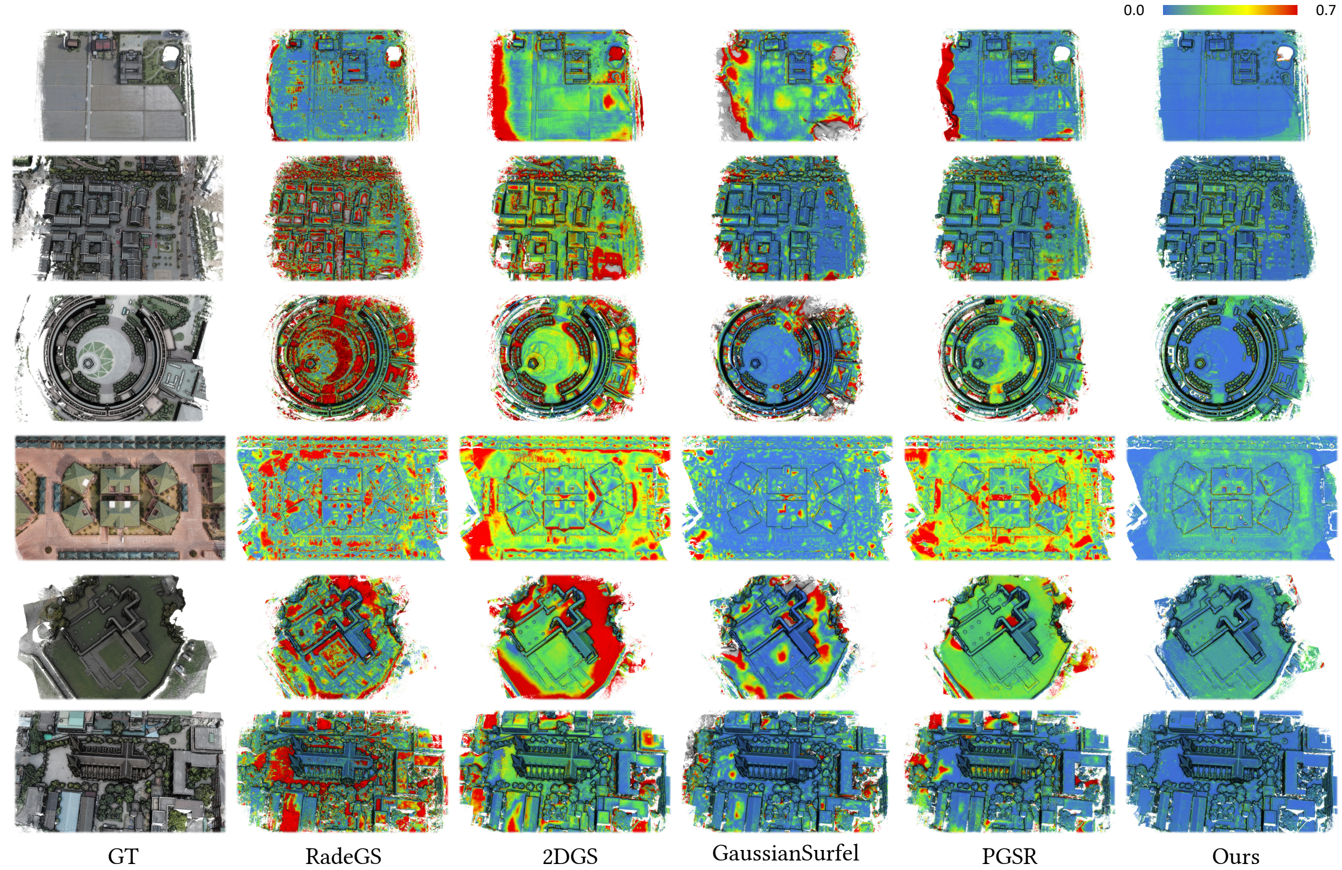

Figure 6: Visualization results comparison with SOTA methods on BlendedMVS dataset.

the GT depth D_{gt} and the PatchMatch depth D_{pm}:

$$Label_{pm} = \frac{Abs(D_{pm} - Dgt)}{max(D_{pm})} \cdot (Cost_{pm} < 0.35) > 0.005. \quad (11)$$

This formula means that we select the positions where the NCC cost is less than 0.35 but the relative depth error is greater than 0.5% as the positive examples that need to be predicted.

Since most of the regions can be filtered out by $Cost_{pm}$, we focus our task on identifying the noise that is very difficult to filter. Next, we use a multiresolution residual U-Net to predict the error regions. The inputs contain the RGB image I, the normalized depth map D_{pm}^{filter} filtered by $Cost_{pm} < 0.35$, and the E_{NCC} cost map $Cost_{pm}$, while the output is a set of multiresolution uncertainty maps $Uncert$, which are used to indicate the degree of difference from the image semantics per pixel y_p.

We use weighted BCE loss as the loss function of the Semantic Aware Module:

$$L_{sa} = -\sum_{i=1}^{n} w_p y_p \log(\hat{y}_p) + (1 - w_p)(1 - y_p) \log(1 - \hat{y}_p), \quad (12)$$

where $y_p \in [0, 1]$ is the Semantic Aware Module output to indicate if p is a noise pixel, $\hat{y}_p$ is the GT label of the pixel p (if $\hat{y}_p = 1$, it is the case that needs to be identified), w_p is used to control the weight of this sample in the overall BCE loss. Since the regions that need to be identified usually account for a relatively small proportion, we use the weighted BCE loss to balance the influence of positive and negative samples on the network training. For positive samples, we use the ratio of the number of positive and negative samples in each case as w_p. For negative samples, we use the reciprocal of this ratio as w_p.

Therefore, the depth D_{pb}^{sem} used to guide the Gaussian Branch is ultimately defined as:

$$D_{pb}^{sem} = D_{pb} \cdot (Uncert < 0.5). \quad (13)$$

After the training is completed, we obtain the depth map by rendering from the Gaussian Branch. Then, we use DepthFusion to generate a point cloud for evaluation or TSDF Fusion to generate a mesh model.

4 Experiments

4.1 Experimental Setup

We conduct experiments on the DTU [15] dataset and the more challenging BlendedMVS [39] dataset. The scene sizes in the BlendedMVS dataset vary greatly, and the distribution of camera poses

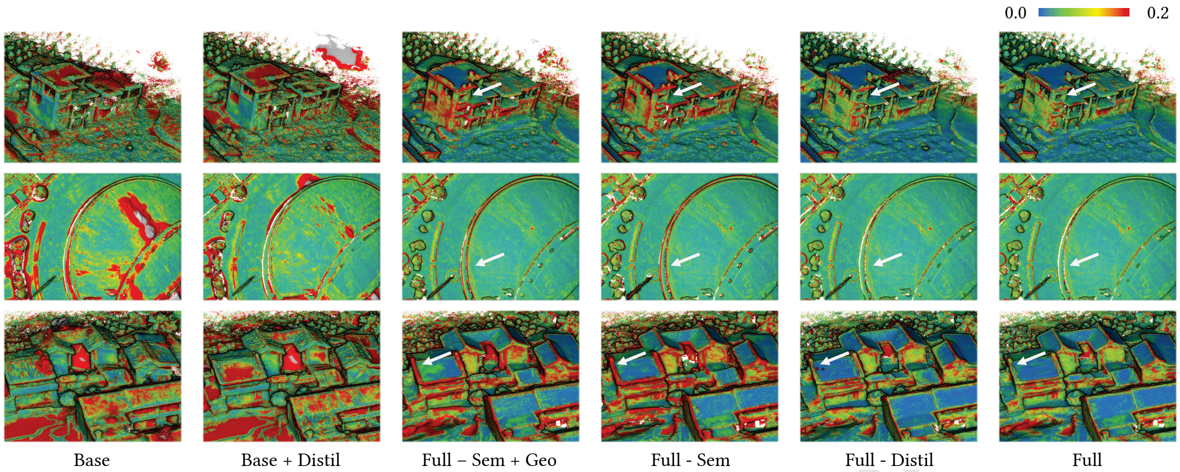

Figure 7: Visualization results of the ablation study.

Table 1: Quantitative evaluation of reconstruction with existing SOTA methods.

	BlendedMVS								DTU			
	Prec ↑	Recall ↑	F-score ↑	Acc ↓	Comp ↓	Overall ↓	PSNR↑	SSIM↑	CD ↓	Time ↓	PSNR↑	SSIM↑
RadeGS	59.776	73.337	65.632	0.178	0.136	0.157	24.49	0.68	0.69	47min	33.92	0.92
2DGS	66.368	82.927	73.504	0.161	0.102	0.132	26.70	0.77	0.80	33min	33.75	0.91
GaussianPro	67.723	78.978	72.087	0.158	0.117	0.138	27.93	0.82	1.08	21min	34.11	0.93
PGSR	72.764	85.039	78.140	0.144	0.096	0.120	28.62	0.84	0.53	37min	32.65	0.92
GaussianSurf	82.124	90.362	85.985	0.118	0.076	0.097	25.34	0.76	0.88	18min	28.13	0.88
Ours	87.301	93.960	90.267	0.106	0.069	0.088	26.95	0.79	0.52	37min	33.15	0.91

is more complex. We use 15 DTU test scenes as [14] and 17 BlendedMVS large scenes for the experiments. We compare GSAPro with two methods without geometric constraints: 2DGS [14] and RadeGS [46], one PatchMatch-based method: GaussianPro [4] with the regularization term used for flattening the Gaussians, and two most advanced methods that utilize multiview constraints: GaussianSurf [6] and PGSR [2]. The threshold for fusing the depth maps together is set to 0.001 for the DTU dataset and 0.05 for the BlendedMVS dataset.

Our Semantic Aware Module's residual U-net has 1.136M parameters. The proposed BlendedPM dataset consists of 95 scenes and 61,896 data pairs. For validation, we utilize a total of 3,972 data pairs from 20 scenes. We use the AdamW optimizer to train the model for 35 epochs on a single NVIDIA RTX 3090 GPU, which takes approximately 20 hours. The learning rate is gradually reduced from 0.001 to 0.0001.

4.2 Noise Identification

We present the visualization (Figure 5) and the numerical analysis (Table 2) results of the identification of stubborn noise by our method. The Semantic Aware Module is only trained on the BlendedPM dataset and directly performed inference on the DTU dataset. From the visualization results, it can be seen that our method has excellent robustness. When we remove the Cost $Cost_{pm}$ from the input of the network (w/o Cost) or do not use the cost $Cost_{pm}$ to filter the input depth $Depth_{pm}$ (w/o Cost Filter), there will be a large number of false positives. This is because not providing the additional constraints is equivalent to increasing the training difficulty of the network. The network outputs the detection results solely based on the depth and image information. Therefore, the performance is inferior to that of the complete full model.

Table 2: Numerical comparison of our Semantic Aware Module's settings and with geometric filter.

	BlendedMVS			DTU (without training)		
	Prec ↑	Recall ↑	F-score ↑	Prec ↑	Recall ↑	F-score ↑
geo filter high	34.302	72.989	44.035	20.433	84.425	31.114
geo filter low	64.653	12.523	18.173	68.408	16.989	24.442
w/o cost	49.429	92.765	64.190	56.524	74.914	63.509
w/o cost filter	68.418	75.920	71.778	67.882	58.999	62.075
full	72.042	75.400	73.515	73.930	57.398	63.637

We also compare the detection efficiency with the geometric filter. We use the minimum resolution of the Depth Fusion as the strict threshold and use a value that differs by 0.5% from the maximum depth of the GT depth as the loose threshold. The number of support points is 3. When a geometric filter with a high threshold is used for detection, a large number of false positives occur. However, when a geometric filter with a lower threshold is used, noise regions cannot be detected.

4.3 Reconstruction

We report the visual (Figure 6 and 8) and numerical (Table 1) comparison results between GSAPro and various SOTA algorithms on the BlendedMVS and DTU datasets. Our algorithm has achieved the same performance as the SOTA algorithms in the DTU scenes. Moreover, in the more challenging BlendedMVS dataset, both the reconstruction accuracy and completeness of our GSAPro are superior to all existing SOTA algorithms.

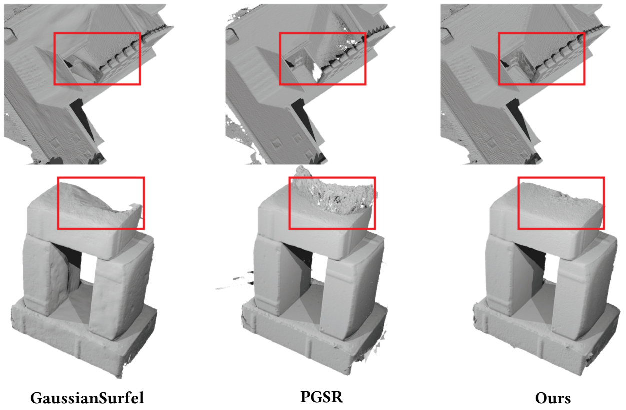

Figure 8: The visualization comparison on DTU dataset. The areas in the box are difficult-to-reconstruct regions.

In complex scenes, the methods [2, 6] that utilize multiview consistency significantly outperform other methods [14, 46] in terms of both reconstruction accuracy and completeness. These methods require calculating the NCC between the rendered image and the auxiliary views during each iteration. Due to the instability of the NCC and the fact that some areas of the scene may not be rendered well, which affects the multiview consistency calculation, there will often be large geometric errors in certain parts of the scene. GSAPro directly uses reliable depth denoising by the Semantic Aware Module to guide the optimization of the Gaussian Branch, which makes the reconstruction more stable. Moreover, the accuracy of the state parameters stored in the Prior Branch keeps increasing, thus making the guidance information more accurate and complete.

4.4 Ablation Study

Table 3: Quantitative results of the ablation study for GSAPro. Since the number of points involved in the evaluation will reach 30M, a slight numerical improvement can signify a substantial enhancement in details.

	Prec ↑	Recall ↑	F-score ↑	Avg Gaus	Improve ↑
Base	71.540	84.587	77.277	1,161,484	0%
Base + Distil	69.426	82.315	75.045	354,006	-5.58%
Full - Sem + Geo	85.823	93.295	89.173	766,807	26.80%
Full - Sem	86.339	93.647	89.616	968,923	27.69%
Full - Distil	87.442	94.003	90.354	2,021,544	29.88%
Full	87.301	93.960	90.267	880,797	29.06%

We conduct the ablation study on BlendedMVS to better illustrate the functions of each component of our GSAPro. For the visualization results, please refer to Figure 7. And for the quantification results, please refer to Table 3.

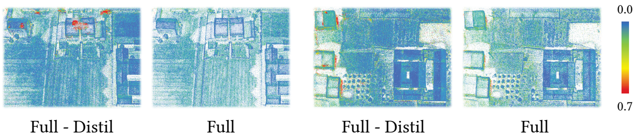

Figure 9: The difference between the Gaussians and the GT point cloud. The Gaussians with lower weights that are removed by Distillation usually appear in the regions far from the GT surface.

Base is the baseline of our GSAPro, which only uses the rasterizer of [2] and the RGB Loss. **Distil** is the Distillation module from [8] that has been integrated into GSAPro to reduce the number of Gaussians. When the Distillation module is used to prune the Gaussians, the number of Gaussians is significantly reduced, the training speed is improved, and there is basically no loss in accuracy (**Full-Distil** vs **Full**).

Full-Sem+Geo replaces the Semantic Aware Module with a strict geometric filter. Since it is difficult for the geometric filter to filter out the noise regions and it will mark a large number of false positive areas, the reliable depth-guided information in a single view is reduced. As a result, the final reconstruction result is inferior to that of GSAPro and even worse than directly using photometric consistency for filtering.

From the comparison between **Full-Sem** and **Full**, we can see that the Semantic Aware Module can effectively improve the precision of the detail reconstruction.

It should be noted that, due to the large size of the scene, the reconstruction quality of most parts is quite similar, and the proportion of high frequency areas and edges with differences is relatively small. Therefore, the numerical differences will not be that obvious.

5 Conclusion

We introduce GSAPro, a novel Semantic Aware surface reconstruction approach, which exhibits robustness across diverse scales of scenes. Regarding the issue that there may be errors in the guidance information, GSAPro proposes a joint optimization strategy to simultaneously enhance the accuracy of the state parameters in the Prior Branch and the reconstruction accuracy of the Gaussian Branch. Regarding the problem that the noise existing in the multiview NCC cost and the pre-computed priors information cannot be accurately identified by the geometric filter, we conduct the training of the Semantic Aware Module on our BlendedPM dataset. This module is able to efficiently detect those noisy regions, and consequently, it remarkably improves the reconstruction accuracy of the details in the Gaussian Branch. Experiments on the DTU dataset demonstrate that our method achieves the SOTA performance. On the more challenging BlendedMVS dataset, GSAPro shows its performance superiority.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (No. 62372336 and No. 61972298) and Wuhan University Huawei GeoInformatices Innovation Lab.

References

[1] Chenjie Cao, Xinlin Ren, and Yanwei Fu. [n. d.]. MVSFormer++: Revealing the Devil in Transformer's Details for Multi-View Stereo. In The Twelfth International Conference on Learning Representations.

[2] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. 2024. PGSR: Planar-based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction. arXiv preprint arXiv:2406.06521 (2024).

[3] Hanlin Chen, Chen Li, and Gim Hee Lee. 2023. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. arXiv preprint arXiv:2312.00846 (2023).

[4] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. 2024. Gaussianpro: 3d gaussian splatting with progressive propagation. In Forty-first International Conference on Machine Learning.

[5] Xiao Cui, Weicai Ye, Yifan Wang, Guofeng Zhang, Wengang Zhou, and Houqiang Li. 2024. Streetsurfgs: Scalable urban street surface reconstruction with planar-based gaussian splatting. arXiv preprint arXiv:2410.04354 (2024).

[6] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. 2024. High-quality surface reconstruction using gaussian surfels. In ACM SIGGRAPH 2024 Conference Papers. 1–11.

[7] Lue Fan, Yuxue Yang, Minxing Li, Hongsheng Li, and Zhaoxiang Zhang. 2024. Trim 3D Gaussian Splatting for Accurate Geometry Representation. CoRR (2024).

[8] Guangchi Fang and Bing Wang. 2024. Mini-Splatting: Representing Scenes with a Constrained Number of Gaussians. arXiv preprint arXiv:2403.14166 (2024).

[9] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. 2022. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. Advances in Neural Information Processing Systems 35 (2022), 3403–3416.

[10] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. 2015. Massively parallel multiview stereopsis by surface normal diffusion. In Proceedings of the IEEE International Conference on Computer Vision. 873–881.

[11] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. 2020. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2495–2504.

[12] Antoine Guédon and Vincent Lepetit. 2024. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5354–5363.

[13] Huasong Han, Kaixuan Zhou, Xiaoxiao Long, Yusen Wang, and Chunxia Xiao. 2025. Ggs: Generalizable gaussian splatting for lane switching in autonomous driving. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 39. 3329–3337.

[14] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2d gaussian splatting for geometrically accurate radiance fields. In ACM SIGGRAPH 2024 conference papers. 1–11.

[15] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. 2014. Large scale multi-view stereopsis evaluation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 406–413.

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics 42, 4 (July 2023).

[17] Hongjie Li, Yao Guo, Xianwei Zheng, and Hanjiang Xiong. 2024. Learning deformable hypothesis sampling for accurate patchmatch multi-view stereo. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38. 3082–3090.

[18] Zongcheng Li, Xiaoxiao Long, Yusen Wang, Tuo Cao, Wenping Wang, Fei Luo, and Chunxia Xiao. 2023. NeTO: neural reconstruction of transparent objects with self-occlusion aware refraction-tracing. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 18547–18557.

[19] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. 2023. Neuralangelo: High-Fidelity Neural Surface Reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8456–8465.

[20] Jie Liao, Yanping Fu, Qingan Yan, Fei Luo, and Chunxia Xiao. 2021. Adaptive depth estimation for pyramid multi-view stereo. Computers & Graphics 97 (2021), 268–278.

[21] Jie Liao, Yanping Fu, Qingan Yan, and Chunxia Xiao. 2019. Pyramid multi-view stereo with local consistency. In Computer Graphics Forum, Vol. 38. Wiley Online Library, 335–346.

[22] Chunjie Luo, Fei Luo, Yusen Wang, Enxu Zhao, and Chunxia Xiao. 2024. Dlcarecon: dynamic loose clothing avatar reconstruction from monocular videos. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38. 3963–3971.

[23] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 2024. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting. ACM Transactions on Graphics (TOG) 43, 6 (2024), 1–12.

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In European conference on computer vision. Springer, 405–421.

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

[26] Jiongming Qin, Fei Luo, Tuo Cao, Wenju Xu, and Chunxia Xiao. 2024. HS-Surf: A Novel High-Frequency Surface Shell Radiance Field to Improve Large-Scale Scene Rendering. In Proceedings of the 32nd ACM International Conference on Multimedia. 6006–6014.

[27] Christian Reiser, Stephan Garbin, Pratul Srinivasan, Dor Verbin, Richard Szeliski, Ben Mildenhall, Jonathan Barron, Peter Hedman, and Andreas Geiger. 2024. Binary opacity grids: Capturing fine geometric detail for mesh-based view synthesis. ACM Transactions on Graphics (TOG) 43, 4 (2024), 1–14.

[28] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In Conference on Computer Vision and Pattern Recognition (CVPR).

[29] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. 2016. Pixelwise view selection for unstructured multi-view stereo. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. Springer, 501–518.

[30] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In European Conference on Computer Vision (ECCV).

[31] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. 2021. Patchmatchnet: Learned multi-view patchmatch stereo. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 14194–14203.

[32] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021).

[33] Yusen Wang, Zongcheng Li, Yu Jiang, Kaixuan Zhou, Tuo Cao, Yanping Fu, and Chunxia Xiao. 2022. NeuralRoom: Geometry-Constrained Neural Implicit Surfaces for Indoor Scene Reconstruction. ACM Transactions on Graphics (TOG) 41, 6 (2022), 1–15.

[34] Yusen Wang, Kaixuan Zhou, Wenxiao Zhang, and Chunxia Xiao. 2024. MegaSurf: Scalable Large Scene Neural Surface Reconstruction. In Proceedings of the 32nd ACM International Conference on Multimedia. 6414–6423.

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

[36] Jiang Wu, Rui Li, Haofei Xu, Wenxun Zhao, Yu Zhu, Jinqiu Sun, and Yanning Zhang. 2024. Gomvs: Geometrically consistent cost aggregation for multi-view stereo. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 20207–20216.

[37] Qingshan Xu, Weihang Kong, Wenbing Tao, and Marc Pollefeys. 2022. Multi-scale geometric consistency guided and planar prior assisted multi-view stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 45, 4 (2022), 4945–4963.

[38] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. 2018. Mvsnet: Depth inference for unstructured multi-view stereo. In Proceedings of the European Conference on Computer Vision (ECCV). 767–783.

[39] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. 2020. BlendedMVS: A Large-scale Dataset for Generalized Multi-view Stereo Networks. Computer Vision and Pattern Recognition (CVPR) (2020).

[40] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. 2021. Volume rendering of neural implicit surfaces. Advances in Neural Information Processing Systems 34 (2021), 4805–4815.

[41] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. 2024. Absgs: Recovering fine details in 3d gaussian splatting. In Proceedings of the 32nd ACM International Conference on Multimedia. 1053–1061.

[42] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. 2024. Gsdf: 3dgs meets sdf for improved rendering and reconstruction. arXiv preprint arXiv:2403.16964 (2024).

[43] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. 2024. Mip-splatting: Alias-free 3d gaussian splatting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 19447–19456.

[44] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. 2022. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. Advances in neural information processing systems 35 (2022),

25018–25032.

[45] Zehao Yu, Torsten Sattler, and Andreas Geiger. 2024. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. ACM Transactions on Graphics (TOG) 43, 6 (2024), 1–13.

[46] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. 2024. RaDe-GS: Rasterizing Depth in Gaussian Splatting. arXiv preprint arXiv:2406.01467 (2024).

[47] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. 2020. Visibility-aware multi-view stereo network. arXiv preprint arXiv:2008.07928 (2020).

[48] Wenyuan Zhang, Yu-Shen Liu, and Zhizhong Han. 2024. Neural Signed Distance Function Inference through Splatting 3D Gaussians Pulled on Zero-Level Set. In Advances in Neural Information Processing Systems.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009