

MatCha Gaussians: Atlas of Charts for High-Quality Geometry and Photorealism From Sparse Views

Appendix

In this appendix, we describe

- additional implementation details,
- details about our mesh extraction method,
- and additional qualitative results.

We also provide a [video](#) that offers an overview of the approach and showcases additional qualitative results.

1. Implementation Details

1.1. Initializing Charts

For initializing the charts using a monocular depth estimation model, we not only backproject the depth maps into 3D but also roughly adjust the scale of the depth estimates using a global affine rescaling model [8, 10]. Note that we can compute an explicit closed-form solution for this affine rescaling which executes in less than a second.

In our experiments, the size of our charts is proportional to the input views, and the longest sides of the charts have length $\max(h, w) = 512$. We rely on MAST3R-SfM [4] to obtain an SfM point cloud for aligning the charts.

1.2. Chart Deformation Model

We can adjust the resolution of the learnable charts encodings (*i.e.*, r) according to the density of the SfM points or the number of views. The sparser the SfM point cloud or the training images, the lower the resolution of the charts encodings. In other words, we can explicitly adjust the strength of the inductive bias in our chart deformation model according to the different scenarios. For small scenes with only 3 input views like the objects from the DTU [1] dataset, we use a small resolution parameter $r = 0.1$ for the charts encodings. In larger and unbounded scenes with 5 or 10 input views, we use a larger resolution parameter $r = 0.4$ for our charts encodings.

The other hyperparameters are constants and independent of the inputs. In practice, we set $d = 32$ and use an MLP with only 1 hidden layer. The number of channels in the hidden layer is 64. For aligning our charts with the initial SfM points, we optimize our model for 1000 iterations. For refining the charts, we optimize our model for 3000 iterations.

During the alignment with the SfM points, we deform the charts along the camera rays, as we empirically found it to be more robust. Moreover, deforming the charts along the camera rays enables very efficient computation of the mutual alignment loss, as in this case, the 3D to 2D mapping of our charts is equivalent to the camera screen projection

transform. To deform charts along the rays, we use a one-dimensional output layer for the MLP, and we compute the 3D deformation by multiplying the MLP output by the ray direction.

During the refinement with Gaussian surfel rendering, we first update the initial charts $\psi_i^{(0)}$ and replace them with the deformed charts ψ_i ; Then, we reinitialize the weights of the MLP and replace the output layer with a 3-dimensional layer in order to learn a full 3D deformation for the charts.

1.3. Refining the Manifold with Gaussian Surfels

During the second optimization stage, we rely on a photometric loss to refine the manifold. At each iteration, we render the manifold by first instantiating 2D Gaussian surfels on the surface, then rasterizing the Gaussians with a surfel rasterizer [6].

Photometric loss The photometric loss consists of an L1 loss \mathcal{L}_1 and a D-SSIM term \mathcal{L}_{D-SSIM} :

$$\mathcal{L}_{\text{photo}} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM}, \quad (1)$$

where we set $\lambda = 0.2$, following past 3DGS works [7].

Structure loss To preserve the fine geometry of our aligned charts, we maintain the structure loss but replace the depth estimates with the depth of our aligned charts. We also weight the structure loss using our confidence maps C_i estimated during the manifold alignment to the SfM points, which makes it a scale-accurate depth regularization robust to outliers.

To further regularize the geometry, we also use a depth-normal consistency loss and a depth distortion loss, as introduced in 2DGS [6].

Distortion loss The distortion loss prevents Gaussians from spreading around the surface. Since we instantiate Gaussian surfels on the manifold represented as a collection of charts, the distortion term enforces the surfaces of the different charts to align together and form a coherent manifold. For each pixel p , the distortion loss is given by

$$\mathcal{L}_d = \sum_{i,j} \omega_i \omega_j |z_i - z_j|, \quad (2)$$

where i and j represent the i -th and j -th Gaussian surfels intersected along the ray, z_i is the depth of the intersection point between the ray and the i -th Gaussian surfel, and ω_i is the blending weight of the i -th intersection.

Depth-Normal consistency loss The depth-normal consistency loss aims to align the normals of the closest Gaussian surfels along the ray with the gradient of the depth map. In our case, this term encourages the surfaces of the different charts to have the same orientation. For a pixel p , the normal consistency loss at p is given by

$$\mathcal{L}_n = \sum_i \omega_i (1 - \mathbf{n}_i^T \mathbf{N}_p), \quad (3)$$

where \mathbf{n}_i is the normal of the i -th Gaussian surfel along the ray and \mathbf{N}_p is the normal at pixel p computed from the gradient of the depth map.

Optimization loss The complete loss for refining the charts is

$$\mathcal{L}_{\text{refine}} = \mathcal{L}_{\text{photo}} + \lambda_{\text{struct}} \mathcal{L}_{\text{struct}} + \lambda_d \mathcal{L}_d + \lambda_n \mathcal{L}_n, \quad (4)$$

where we set $\lambda_{\text{struct}} = 1$, $\lambda_d = 500$, and $\lambda_n = 0.25$. We refine the representation for 3000 iterations, and introduce \mathcal{L}_d and \mathcal{L}_n only after 600 iterations.

1.4. Extracting a Surface Mesh from the Manifold

We propose two different approaches for extracting a surface mesh from our manifold representation, depending on the scene complexity and the desired level of detail.

Direct Mesh Extraction For scenes with moderate complexity or extreme sparse-view setups (e.g., 3 views on DTU), we can directly extract a surface mesh from our manifold using a custom multi-resolution TSDF fusion approach, or a custom implementation of the adaptive tetrahedralization from Gaussian Opacity Fields [11]. Since we describe our tetrahedralization in the main paper, we focus on providing additional details about the multi-resolution TSDF below.

We render depth maps from our manifold and fuse them into several TSDF volumes with different resolutions. The lower the resolution, the larger the bounding box used for applying the TSDF algorithm. Then, we merge the TSDF volumes and remove the overlapping regions. Note that, in a sparse-view scenario, the number of depth maps is very low, so that integrating depth maps for computing the TSDF volumes is very fast and takes less than a minute.

Our multi-resolution approach allows us to accurately reconstruct both foreground objects and background regions with a decent number of vertices, which is crucial for unbounded scenes. However, even though our multi-resolution TSDF is very fast, it generally erodes the geometry and creates holes in the extracted surface. In this regard, we recommend using the tetrahedralization for extracting meshes.

Free Gaussians Refinement For scenes requiring finer geometric details, particularly in large unbounded environments, we propose an additional refinement step that leverages our manifold as a strong geometric prior. Instead of directly extracting the mesh, we first let Gaussian surfels get freely optimized in 3D space for a few iterations while strongly constraining them with our manifold representation.

For this, we freeze the manifold but unfreeze the Gaussians' parameters (position, scale, and rotation) and regularize them using depth maps rendered from the manifold through a combination of our refinement loss and an L1 depth loss with a weighting factor $\lambda_{\text{depth}} = 0.75$. We also use our confidence maps to weigh the depth regularization, but not the structure loss that relies on the derivatives of the depth. Indeed, applying the structure loss everywhere in the scene enables regularization of Gaussians located even in low-confidence areas, where normal maps and curvature maps still provide a reliable supervision signal despite of inaccurate depth values.

This refinement step is particularly effective because our manifold provides scale-accurate regularization, unlike traditional depth-based regularization methods that often struggle with scale ambiguity. The manifold acts as a reliable geometric prior that prevents Gaussians from diverging while letting them recover fine surface details that might not be fully captured by the manifold representation alone.

After this Gaussian refinement stage, we extract the final mesh using the same multi-resolution TSDF fusion or tetrahedralization approaches described above, but now applied to the refined Free Gaussians representation. This two-stage approach allows us to recover very fine geometric details while maintaining the overall accuracy and robustness of our manifold representation.

2. Additional Results and Details

Surface Reconstruction Fig. 1 shows qualitative results. Our method can reconstruct high-quality surfaces across different scenarios, from bounded objects (DTU [1] dataset) to unbounded scenes (Tanks&Temples [9] and Mip-NeRF 360 [2] datasets), using varying numbers of, but sparse, input views (3, 5, and 10 views). For each example, we show a rendered view, the estimated depth map, surface normals, and the extracted mesh, which collectively show the consistency of our reconstruction across different representations. For the objects from the DTU dataset, we directly extract the mesh from the manifold representation using our multi-resolution TSDF fusion approach. For the unbounded scenes from the T&T and Mip-NeRF 360 datasets, we first refined free Gaussians around the manifold as explained in the previous section, then extracted the mesh using the same multi-resolution TSDF fusion approach.

In the 3-view scenarios (first two rows), our method suc-

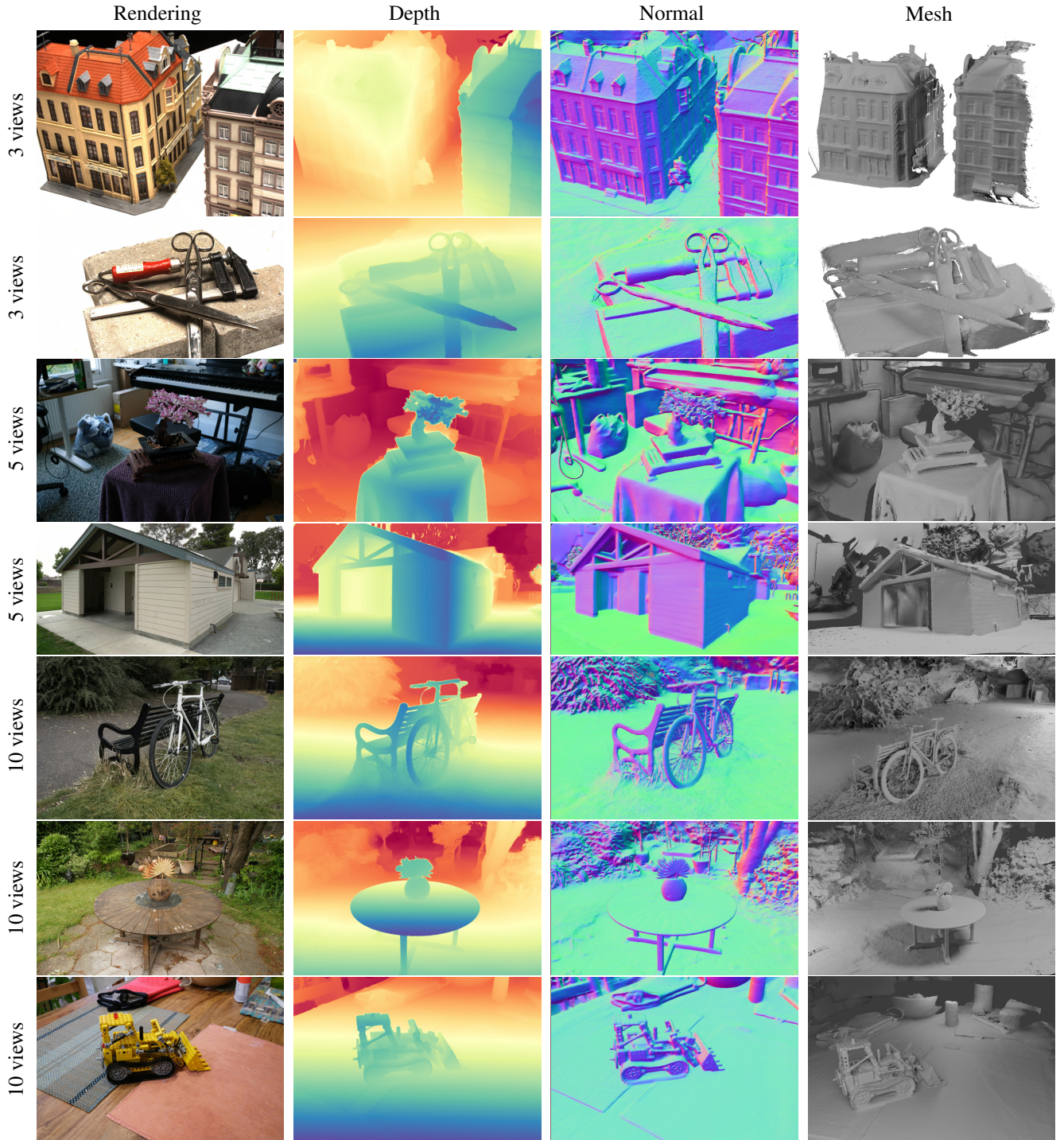


Figure 1. **Qualitative reconstruction results across different scenarios and numbers of input views.** We show results on both bounded objects from DTU [1] (first two rows, 3 views) and unbounded scenes from Tanks&Temples [9] and Mip-NeRF 360 [2] (middle and bottom rows). For each example, we show (from left to right): the rendered novel view, estimated depth map, surface normals, and the extracted mesh. For bounded objects (DTU), meshes are extracted directly from our manifold representation, while for unbounded scenes, we first refine free Gaussians around the manifold before mesh extraction. Note how our method maintains consistent quality across different scenarios, from small objects to large-scale scenes with complex backgrounds.

	Mip-NeRF 360 [2]		Tanks&Temples [9]		DeepBlending [5]	
	10%Q PSNR \uparrow	Avg PSNR \uparrow	10%Q PSNR \uparrow	Avg PSNR \uparrow	10%Q PSNR \uparrow	Avg PSNR \uparrow
5 training views						
2DGS [6]+MASt3R-SfM [4]	15.37	20.84	14.23	16.42	15.84	19.86
GOF [11]+MASt3R-SfM [4]	15.78	21.24	13.69	16.50	15.58	19.87
MAtCha (Ours)	18.18	21.90	15.33	17.30	17.22	20.60
10 training views						
2DGS [6]+MASt3R-SfM [4]	19.94	24.31	16.63	19.59	14.06	21.14
GOF [11]+MASt3R-SfM [4]	20.99	24.50	16.81	19.59	12.61	21.12
MAtCha (Ours)	21.55	25.10	17.96	20.38	17.41	22.98

Table 1. **Quantitative evaluation of Novel View Synthesis in sparse-view scenarios across multiple real-world datasets.** We evaluate our method against baselines on three challenging datasets: Mip-NeRF 360 [2], Tanks&Temples [9], and DeepBlending [5]. Baselines consist of recent state-of-the-art approaches augmented with MASt3R-SfM [4] for more robustness to sparse-view scenarios. For each dataset and method, we report both the average PSNR and the 10% quantile PSNR (10%Q PSNR) which better reflects performance on challenging views and better capture the ability of a method to generalize to novel viewpoints. Results are shown for both 5-view and 10-view scenarios, demonstrating our method’s superior performance across different sparsity levels.



Figure 2. **Qualitative comparisons of novel view synthesis with MVSplat360 [3] on an unbounded scene with 5 training images.** Our method can render more photorealistic images than the concurrent feed-forward novel view synthesis method in sparse view scenarios.

successfully recovers detailed geometry despite the extreme sparsity of input views. The 5-view and 10-view examples (middle and bottom rows) demonstrate how our approach scales to larger unbounded scenes while maintaining reconstruction quality throughout the scene, including distant background regions.

Novel View Synthesis Tab. 1 of the main paper provides results of novel view synthesis in sparse-view settings across three challenging real-world datasets of unbounded scenes: Mip-NeRF 360 [2], Tanks&Temples [9], and DeepBlending [5]. Specifically, we follow 3DGS [7] and use the scenes *Playroom* and *Dr. Johnson* for evaluation on the DeepBlending dataset. For the T&T dataset, we use the standard split of 6 scenes as used in 2DGS [6] and GOF [11] but removed *Courthouse* and *Meetingroom*, as these very large scenes are not suitable for sparse-view scenarios with only 5 or 10 input views.

We consider two scenarios with 5 and 10 training views, respectively. For each dataset, we built training sets of 5 and

10 input views and evaluated on a set of 10 test views, including both easy views with high overlap with the training views and much more challenging views with very limited overlap. For fair comparison, we augment recent state-of-the-art methods (2DGS [6] and GOF [11]) with MASt3R-SfM [4], as it provides better robustness in sparse-view scenarios.

We report both average PSNR and 10% quantile PSNR (10%Q PSNR) metrics. The 10%Q PSNR is the PSNR value below which 10% of the test views fall. Average PSNR provides an overall measure of reconstruction quality. In contrast, the 10%Q PSNR specifically captures accuracy on the most challenging views as well as the ability of a method to generalize to novel viewpoints. This metric is particularly relevant to sparse-view settings where some novel viewpoints may have very limited overlap with input views.

As shown in Tab. 1, our method consistently outperforms the baselines across all datasets and metrics. In the 5-view scenario, we achieve significant improvements over the baselines. The performance gap remains substantial even when increasing to 10 input views, where our method maintains superior reconstruction quality across datasets. This consistent performance advantage demonstrates the effectiveness of our chart-based representation and refinement approach in handling sparse-view scenarios.

Notably, our method shows particular strength in maintaining quality for challenging views, as evident in the larger improvements in 10%Q PSNR compared to average PSNR. This suggests that our chart-based representation, combined with the robust deformation model and multi-stage refinement process, helps maintain consistency even in regions with limited overlap in views.

We also qualitatively compare our method with MVSplat360 [3], a concurrent method for feed-forward novel view synthesis in sparse-view settings. Fig. 2 shows rendering results of MVSplat360 and our method from novel viewpoints by using 5 views. MVSplat360 suffers from a domain gap from training data and limited image resolution due to training of its feed-forward networks, leading to unrealistic rendering results.

References

- [1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjarholm Dahl. Large-Scale Data for Multiple-View Stereopsis. *IJCV*, pages 1–16, 2016. 1, 2, 3
- [2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5460–5469. IEEE, 2022. 2, 3, 4
- [3] Yuedong Chen, Chuanxia Zheng, Haoifei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. MVSplat360: Feed-Forward 360 Scene Synthesis from Sparse Views. In *NeurIPS*, 2024. 4, 5
- [4] Bardienus Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MAST3R-SfM: a Fully-Integrated Solution for Unconstrained Structure-from-Motion. *arXiv preprint arXiv:2409.19152*, 2024. 1, 4
- [5] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel J. Brostow. Deep blending for free-viewpoint image-based rendering. *ACM TOG*, 37(6): 257, 2018. 4
- [6] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In *ACM SIGGRAPH Conference Papers*, pages 1–11, 2024. 1, 4
- [7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM TOG*, 42(4), 2023. 1, 4
- [8] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A Hierarchical 3D Gaussian Representation for Real-Time Rendering of Very Large Datasets. *ACM TOG*, 43(4), 2024. 1
- [9] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM TOG*, 36(4), 2017. 2, 3, 4
- [10] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. DN-Splatter: Depth and Normal Priors for Gaussian Splatting and Meshing. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025. 1
- [11] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian Opacity Fields: Efficient Adaptive Surface Reconstruction in Unbounded Scenes. *ACM TOG*, 2024. 2, 4