Planar Gaussian Splatting
— Supplementary Material —

Farhad G. Zanjani Hong Cai Hanno Ackermann Leila Mirvakhabova Fatih Porikli

Qualcomm AI Research[1]

{fzanjani, hongcai, hackerma, lmirvakh, fporikli}@qti.qualcomm.com

This supplementary document includes further implementation details of the proposed Planar Gaussian Splatting method, which is discussed in Section A. Additionally, we provide an evaluation of geometry reconstruction and present more quantitative results in Section B.

A. Implementation Details

We initialize the Gaussian locations and color with the sparse point cloud obtained by running SfM on training images. We train the model on a single NVIDIA GeForce RTX 2080 Ti for 15K iterations. We follow almost all the hyperparameters introduced in [16], except that we increase the densification threshold on the gradient to 0.001 in order to limit the number of cloning/splitting Gaussians and using only two degrees of spherical harmonics because a high-quality rendering is not the primary objective of this work. Cutting edges of RAG is performed by setting a threshold of 10 cm on planar distances and 20 degrees for the surface normal cosine distance. The recursive mean-shift is run once after 2k iterations of training and repeated every 100 training iterations, and the number of update steps (Eq. 8) performed each time is set to 10 with γ equal to 60. The number of neighbours (K) is equal to 30 and the KNN is recomputed after applying any refinement to the 3D Gaussian field, including cloning, splitting, and culling of the Gaussians. Local planar alignment is applied every 500 iterations.

A.1. Efficient implementation of holistic separability

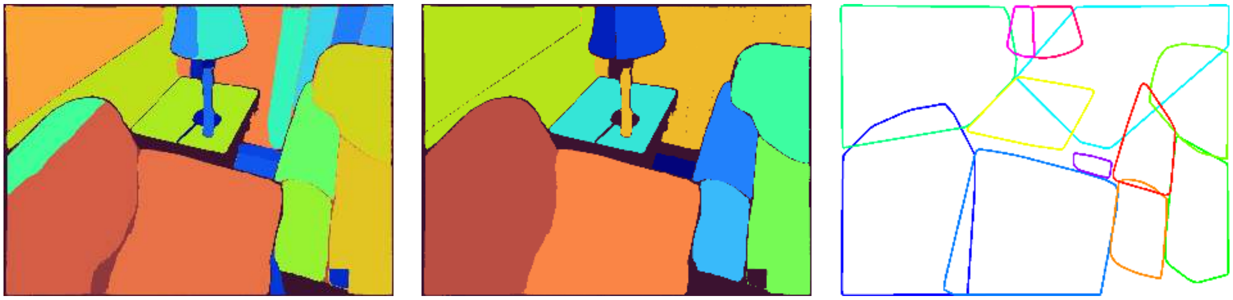
Applying recurrent mean-shift to the descriptor vectors of all Gaussian points in the scene enhances the separability among descriptors corresponding to distinct planes. The mean-shift update computation involves recursively evaluating Eq. (8) for a specified number of steps:

$$\mathbf{Z} \leftarrow \mathbf{Z} \cdot (\eta \cdot \mathbf{K} \cdot \mathbf{D}^{-1} + (1 - \eta) \cdot \mathbf{I}),$$

where $\mathbf{Z} \in \mathbb{R}^{N \times d}$ denotes the matrix of descriptors of length d (e.g., $d = 3$ in our experiments) for the entire scene with N number of Gaussian points (typically in the order of millions). To compute pairwise distances using the von Mises-Fisher kernel, we encounter a challenge: the computation of $\mathbf{K} \in \mathbb{R}^{N \times N}$ is quadratic with respect to the number of Gaussian points in the scene. This becomes infeasible in terms of computational memory.

In our implementation, we tackle this issue by initially estimating the kernel on a random subset of samples with M points (where $M \ll N$). As a result, the estimated kernel $K \in \mathbb{R}^{M \times M}$ and the update vectors (right-hand side of the equation) are propagated and shared across all KNN samples within the chosen set. The random sampling process (without replacement) continues until an update vector has been computed for all samples or their neighbors. Estimating the mean-shift updates for a large number of Gaussian points in the scene using this sampling strategy takes approximately one second on the GPU, and the required memory can be accommodated within the available resources. Notably, in our experiments, holistic separability occurs every N =100 iterations, introducing only minor overhead to the training time.

[1]Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.



(a) (b) (c)

Figure 5. Example of merging SAM proposals and lifting their boundary points into 3D. The boundary points of each segments is used for estimating a Gaussian distribution in 3D as a leaf node of GMT.

A.2. Local grouping of Gaussians primitives

In Section 3.2, we address the construction of the Gaussian Mixture Tree (GMT). Rather than directly utilizing the Gaussian points at the leaf nodes — a computationally expensive approach, especially when dealing with a large number of points in a 3D scene — we adopt an alternative strategy. The GMT is formed by computing a set of Gaussian distributions over point clusters derived from all training images. Each cluster corresponds to points belonging to SAM proposal masks after refinement (achieved through merging via RAG).

Figure 5 illustrates this process: (a) SAM Proposals represent distinct regions; (b) the refined version merges the proposals on the same planar surface; (c) boundary points of each mask proposal are lifted into 3D space using the rendered depth map and camera parameters. For each set of boundary 3D points, the mean and covariance matrix of a Gaussian distribution is computed. By estimating the Gaussian distributions from all training images collectively, the leaf nodes of the Gaussian Mixture Tree are determined. These nodes are subsequently used to construct the tree, as explained in Section 3.2.

B. Additional Evaluations

This section provides more quantitative and qualitative evaluation of the Planar Gaussian Splatting (PGS).

Table 4. Performance of geometric reconstruction on ScanNet.

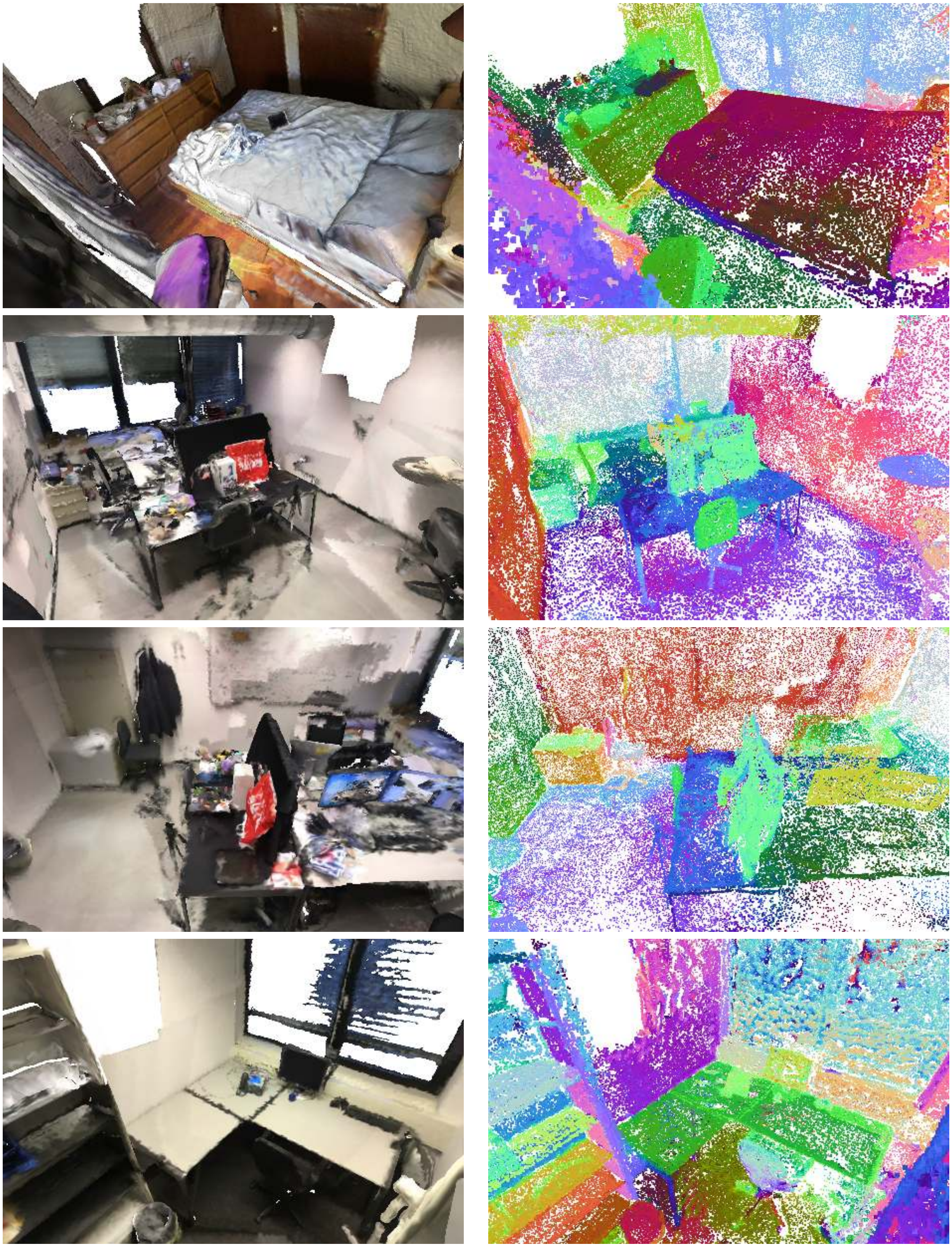
Method	Accuracy↓	Completeness ↓
PlanarRecon [40]	0.154	0.187
PGS (ours)	0.137	0.118

B.1. Geometric reconstruction

In addition to evaluating 3D plane instance segmentation, we assess the performance of both PlanarRecon and the proposed Planar Gaussian Splatting on geometric reconstruction. This evaluation involves measuring the distances between the ground-truth 3D meshes and the surfaces of 3D planes. We adhere to common practice and report reconstruction quality using Accuracy and Completeness metrics (see [27] for their mathematical definitions). The accuracy quantifies the mean distance of the reconstruction points from the ground truth. Completeness measures the extent to which the ground truth is recovered and is defined as the mean distance of the ground truth points to the reconstruction points. In Table 4, we present the measured distances for the ScanNet scenes. Notably, the predicted planes by the proposed PGS exhibit lower distances compared to the ground truth mesh.

B.2. Visualization of plane descriptors of Gaussian field

Figure 5 shows samples scenes of ScanNet and Replica, and the learnt plane descriptors. The visualized points are the centers of Gaussians where are colorized by the learnt descriptors. The distinct colors assigned to different 3D plane instances illustrate the effectiveness of our learnt descriptors in 3D Gaussian field.



(a) (b)

Table 5. Visualization of learnt plane descriptors by planar Gaussian splatting for sample ScanNet and Replica scenes. (a) Ground-truth textured meshes, (b) 3D Gaussian point cloud, colorized by the corresponding plane descriptor.