Level-S^{2}fM: Structure from Motion on Neural Level Set of Implicit Surfaces
Supplementary Material

Yuxi Xiao[1] Nan Xue[1*] Tianfu Wu[2] Gui-Song Xia[1]
[1] School of Computer Science, Wuhan University [2] Department of ECE, NC State University
Project Page: https://henry123-boy.github.io/level-s2fm/

A. Network Architecture Details

The architecture of our Level-S^{2}fM is shown in Fig. 1. We use dual fields to independently represent the radiance field and signed distance field (SDF), which have the same architecture. For each queried 3d point, we will first interpolate the feature of the queried points at multi-resolution grids, and then concatenate the multi-resolution features into the MLP to attain the density or the radiance. To accelerate the training, we adopt the multi-resolution hash table [9] in our implementation. In detail, we construct multiresolution grids of L levels, and the resolution of each level is:

$$N_l := \lfloor N_{min} \cdot b^l \rfloor \ , \tag{1}$$

$$b := \exp\left(\frac{\ln N_{max} - \ln N_{min}}{L-1}\right) \ , \tag{2}$$

where N_{min} and N_{max} are the coarsest and finest resolutions. In the multi-resolution hash table, to obtain the feature of point $\mathbf{x}$, we first scale and round $\mathbf{x}$ at each level l as $\lfloor \mathbf{x}_l \rfloor = \lfloor \mathbf{x} \cdot N_l \rfloor$, $\lceil \mathbf{x}_l \rceil = \lceil \mathbf{x} \cdot N_l \rceil$. Then we can obtain the voxel spanned by $\lfloor \mathbf{x}_l \rfloor$ and $\lceil \mathbf{x}_l \rceil$ and map each corner of the voxel to the hash table using the spatial hash function:

$$h(\mathbf{x}) = \left(\bigoplus_{i=1}^{3} x_i \pi_i\right) \mod T \, , \tag{3}$$

where $\oplus$ denotes the bit-wise XOR operation, and π_i are unique, large prime numbers. In our implementation, π_1, π_2, π_3 and T are set to $1, 2654435761, 805459861, 2^{19}$ respectively. After that, the feature vectors at each corner are interpolated at $\mathbf{x}$ by the interpolation weight $\mathbf{w}_l = \mathbf{x}_l - \lfloor \mathbf{x}_l \rfloor$. Lastly, we concatenate the feature vector of $\mathbf{x}$ at each level, as well as the encoded view direction $\mathbf{v}$ togehter, and send it into an MLP to predict the values. In our implementation, we leverage two different resolution hash tables to represent the sdf and radiance fields respectively. For sdf function, the configuration is $L = 8, N_{min} = 16, N_{max} = 2048$ and the number of features at each level is 4. While the configuration for radiance field is $L = 16, N_{min} = 16, N_{max} = 2048$ and the number of features at each level is 2.

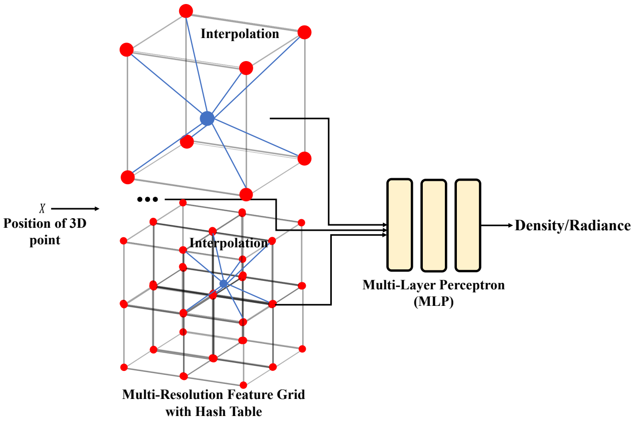
Interpolation
X
Position of 3D point
Interpolation
Multi-Layer Perceptron (MLP)
Density/Radiance
Multi-Resolution Feature Grid with Hash Table

Figure 1. **The Multi-resolution Features Grid.**

B. Evaluation Metrics and Details

Because the world coordinate system varies for different SfM systems, we need to align the estimated poses to the ground truth poses first. We use the reconstruction alignment API from COLMAP [12], which first pre-aligns the two reconstructions with their poses and refine that by aligning the sparse point clouds of them. Here, the sparse point clouds are triangulated by the 2d matches of SIFT with the fixed poses, which can be also easily implemented with the existing COLMAP API. After the alignment, the rotation error is computed as follows:

$$\theta_i^{\text{error}} = \cos^{-1}\frac{\text{trace}(\mathbf{R}_i^{\text{gt}}\hat{\mathbf{R}}_i^T) - 1}{2}, i = \{1, ..., M\}, \tag{4}$$

where the M is the number of the cameras, and $\mathbf{R}_i^{\text{gt}}$, $\hat{\mathbf{R}}_i^{\ T}$ are the gt rotation matrix and aligned estimated rotation matrix respectively. We take the average error of the rotation in Equation (4) as the metric for rotation. As for the evaluation of translation, we use the ATE RMSE [13] to depict

[*] Corresponding Author

Metric	Definition
Acc	$\text{mean}_{p\in P}(\text{min}_{p^*\in P^*}\|\|p-p^*\|\|)$
Prec	$\text{mean}_{p\in P}(\text{min}_{p^*\in P^*}\|\|p-p^*\|\|\leq .035)$

Table 1. Caption

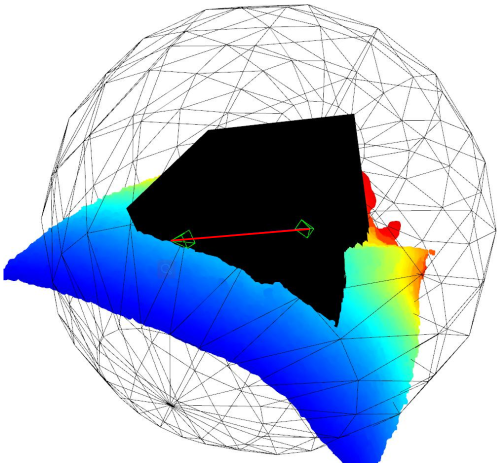

Figure 2. **The Visualization of Two View Initialization for Inside-forward Scenes.**

the distance between the ground truth trajectories and the estimated, specifically followed:

$$\text{RMSE}(\hat{\mathbf{T}}_i) = \left(\frac{1}{M} \sum_{i=0}^{M} ||\text{trans}(\mathbf{T}_i^{-1}\hat{\mathbf{T}}_i)||^2 \right)^{\frac{1}{2}}, \quad (5)$$

where the $\mathbf{T}_i, \hat{\mathbf{T}}_i$ are the gt and aligned transformation matrix respectively, and the trans means to take the translation part of the transformation matrix.

For the evaluation of reconstruction results, the definitions of metrics are shown in Table. 1. We use these two metrics to evaluate the accuracy of the reconstructed point cloud.

C. Two View Initialization

Because the learning of neural implicit fields was originally designed for bounded scenes, we have to carefully design the two-view initialization for our Level-S^{2}fM and ensure the incremental reconstruction process is within the bound. In our study, we mainly focus on two representative types of scenes. The first type is the inside-forward scene, where the cameras are surrounded by the target object and inside forwarding (Specifically seen in Figure. 2). The second type is the outside-forward scenes. In initialization details for inside-forward scenes, we put the first camera on a sphere with $r = 3$, and orient the camera toward the origin of the coordinate. The bound of the features grid is set to $[4, 4, 4]$. The initialized pose for the first camera is calculated by the following:

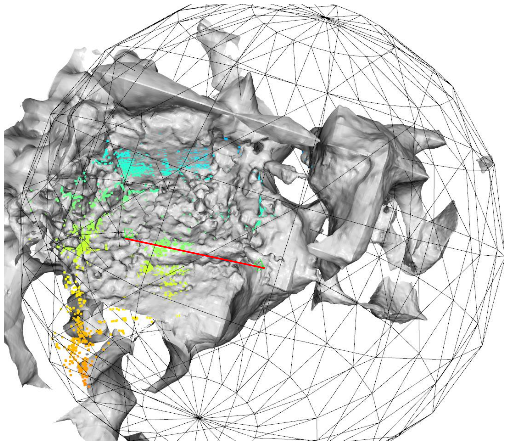

Figure 3. **The Visualization of Two View Initialization for Outside-forward Scenes.**

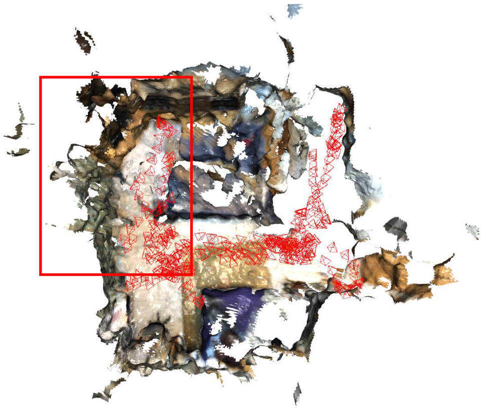

Figure 4. **Visualization of Mesh and Trajectory for Failure Cases.**This figure show the visualization of mesh and trajectory in scannet. As we can see the pose and geometry begin to be bad in the red box where the texture and matches are mostly less.

$$\begin{aligned}
\mathbf{t}_{\text{w2c}}^{\text{init}} &= \mathbf{R}_{\text{w2c}}^{\text{init}}\mathbf{t}_{\text{c2w}}^{\text{init}}, \\
\mathbf{t}_{\text{c2w}}^{\text{init}} &= \begin{bmatrix} -r\cos\theta_y\cos\theta_z \\ -r\cos\theta_y\sin\theta_z \\ -r\sin\theta_y \end{bmatrix}, \\
\mathbf{R}_{\text{w2c}}^{\text{init}} &= \mathbf{R}_x(\theta_x)^{-1}\mathbf{R}_{\mathbf{y}}(\theta_{\mathbf{y}})^{-1}\mathbf{R}_z(\theta_z)^{-1},
\end{aligned} \quad (6)$$

where the $\theta_x = 0, \theta_y = -\frac{1}{4}\pi, \theta_z = \frac{1}{4}\pi$. After that, the pose of the second camera is then initialized with the calculated relative pose by five points algorithm [10]. Meanwhile, the length of the translation of the relative pose would be the hyper-parameter for different scenes. As shown in Figure. 2, the red line is the baseline of the two-view camera, and the length of the baseline is empirically set. Moreover,

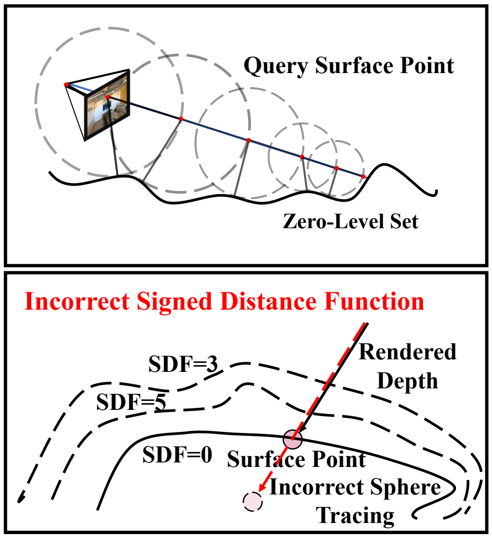

Figure 5. **Sphere Tracing and Depth Consistency.**

for the outside-forward scene, the θ_x is set to $\frac{1}{2}\pi$ to make the orientation of the camera outside. It needs to be noticed that the specific parameters of two-view initialization may be different for different scenes, which can be referred into our codes and configuration after it is released.

D. Sphere Tracing and Depth Consistency

As mentioned in our paper, the key component for attaining the 3D points from the 2D keypoints is sphere tracing. As shown in Figure. 5, sphere tracing algorithm [6] leverages the basic property of the signed distance function where the queried SDF value at each position is the closest distance from the point to the zero-level set of the surface. The depth of the queried 2d points can be calculated as follows:

$$\begin{aligned} \hat{t} &= t_0 + \sum_{j}^{max} sdf(\mathbf{X}_j), \\ X_{j+1} &= X_j + sdf(X_j)d, \end{aligned} \tag{7}$$

where the start point $X_0 = o + t_0 d$. By the sphere tracing algorithm, we can efficiently get the 3d points from 2d, and it can be a natural constraint for the learning of SDF. But sometimes, sphere tracing can not correctly trace the surface when the zero-level set is correct while non-zero-level sets are wrong. Therefore, it can not ensure the multi-view consistency of the sphere tracing algorithm (seen in the second row of Figure. 5). To overcome that, we use the depth calculated by volumetric rendering as a constraint to keep the consistency between these two sampling strategies as mentioned in our paper.

Scene		DTU [4]
Order 1st	rot°↓	0.74
	trans. (mm) ↓	5.81
Order 2nd	rot°↓	0.30
	trans. (mm) ↓	2.04

Table 2. **Ablation for Sequence Order.**

E. Sequence Order for Incremental Reconstruction

For incremental SfM, the sequence order for incremental reconstruction is a relatively important component of the final result. But this paper concentrates our attention into renew the SfM on the neural level sets, which show its promising future to make breakthroughs. To avoid being exhausted to be stuck in the discussion of various tricks and strategies, we simply implement the next best view selection according to the number of 3D-2D pairs in PnP which may have a better alternative discussed in [12]. In order completely discuss our framework, we also report the simple ablation result for the sequence order in Level-S^{2}fM, which can be seen in Table. 2. We conducted the ablation study for sequence order in DTU [4]. We report two different sequence orders by randomly selecting the start of two frames for the two-view initialization. We can see that different sequence orders will cause different results. Despite of this, we would like to emphasize again that because of the complexity of Structure-from-Motion, the problem of the next-best view is not the core of our current study, which will be left in our future work.

F. More Qualitative Results on Individual Datasets

We also report more qualitative results for our experiments. In Figure. 7, there are the rendered image results from our radiance field. While, in Figure. 6, the estimated pose, reconstructed 3d points, and the mesh are visualized.

G. Discussion on Failure Cases

In our paper, we take the indoor datasets, ScanNet [1] to discuss our failure cases, where there are a lot of textureless regions. As shown in Figure. 8, because of the textureless areas or the blurry problem of the captured images, the 3D-2D correspondences are badly distributed and limited in number. Therefore, the registration of these images is hard to solve and results in bad initialization for the pose estimation. As seen in Figure. 4, the trajectory of cameras becomes unsatisfactory due to the textureless wall. Meanwhile, due to the incremental reconstruction fashion, the subsequent pose is based on the former, so, the incorrect

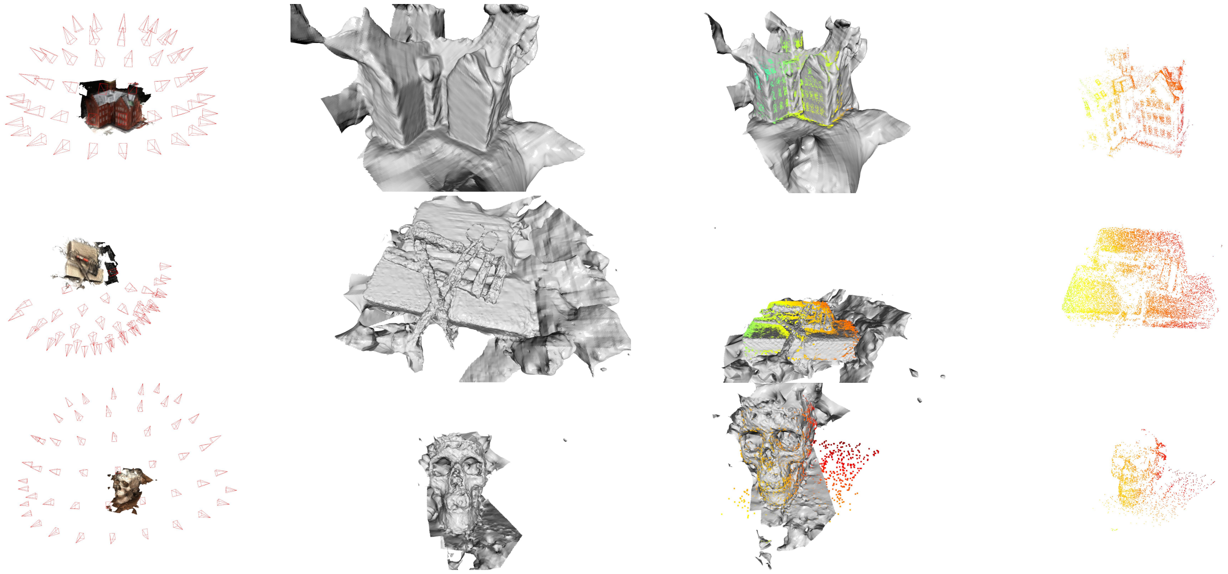

Figure 6. **Qualitative Results for Reconstruction and Pose Estimation.** The first column is the refused mesh and the visualization of camera poses. While, the second and third are the mesh and mesh shown with the points respectively. We can observe that our reconstructed points sticking on the surface of mesh. In the last column is the point cloud reconstructed.

pose estimation results by the textureless region will influence the whole process. To alleviate the problem, the recent robust deep learning-based 2d matches method may play a core role [2, 11], which will be our future works to explore the solution to this problem.

Meanwhile, for those NeRF [7] based SLAMs Framework [14, 15], they usually need the depth as an extra input. With the assistance of the depth map, the coarse pose is easily attained by aligning the depth of two consecutive frames, and they are not easily influenced by the issue of textureless. Therefore, in our paper, we did not compare our method with the depth-aware SLAM methods. Besides, we find that the optimizer for the networks is another core for getting stable and accurate pose results. The optimizer, Adam [5], used in our paper is may not the best choice for our problem as it is not easy to judge whether the optimization is converged. Therefore, in our future work, we are going to explore using the second-order optimizer like Levenberg-Marquardt (LM) [8] or Gauss-Newton (GN) [3] algorithm to solve our problem.

References

[1] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 2432–2443, 2017. 3

[2] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, pages 224–236, 2018. 4

[3] Herman O Hartley. The modified gauss-newton method for the fitting of non-linear regression functions by least squares. *Technometrics*, 3(2):269–280, 1961. 4

[4] Rasmus Ramsbøl Jensen, Anders Lindbjerg Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, pages 406–413. IEEE Computer Society, 2014. 3

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 4

[6] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. DIST: rendering deep implicit signed distance function with differentiable sphere tracing. In *CVPR*, pages 2016–2025, 2020. 3

[7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, volume 12346, pages 405–421, 2020. 4

[8] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978. 4

[9] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 1

[10] David Nistér. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE TPAMI*, 26(6):756–777, 2004. 2

[11] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature

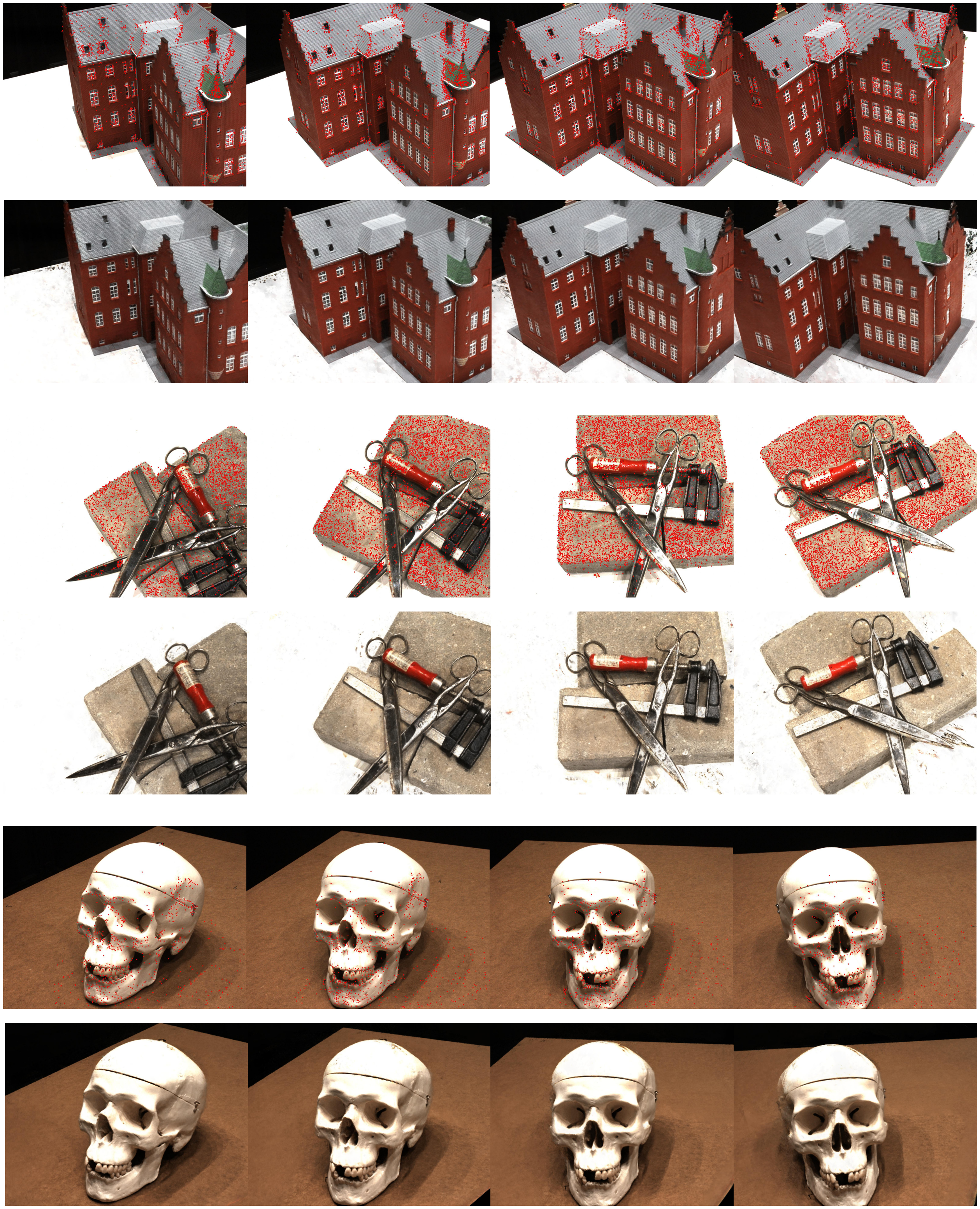

Figure 7. **Qualitative Results for Rendering.**This figure show the visualization of three scenes. At each group, the first row is the ground truth rgb images and their corresponded 3d observation projected on them. And the second row is the rendered images.

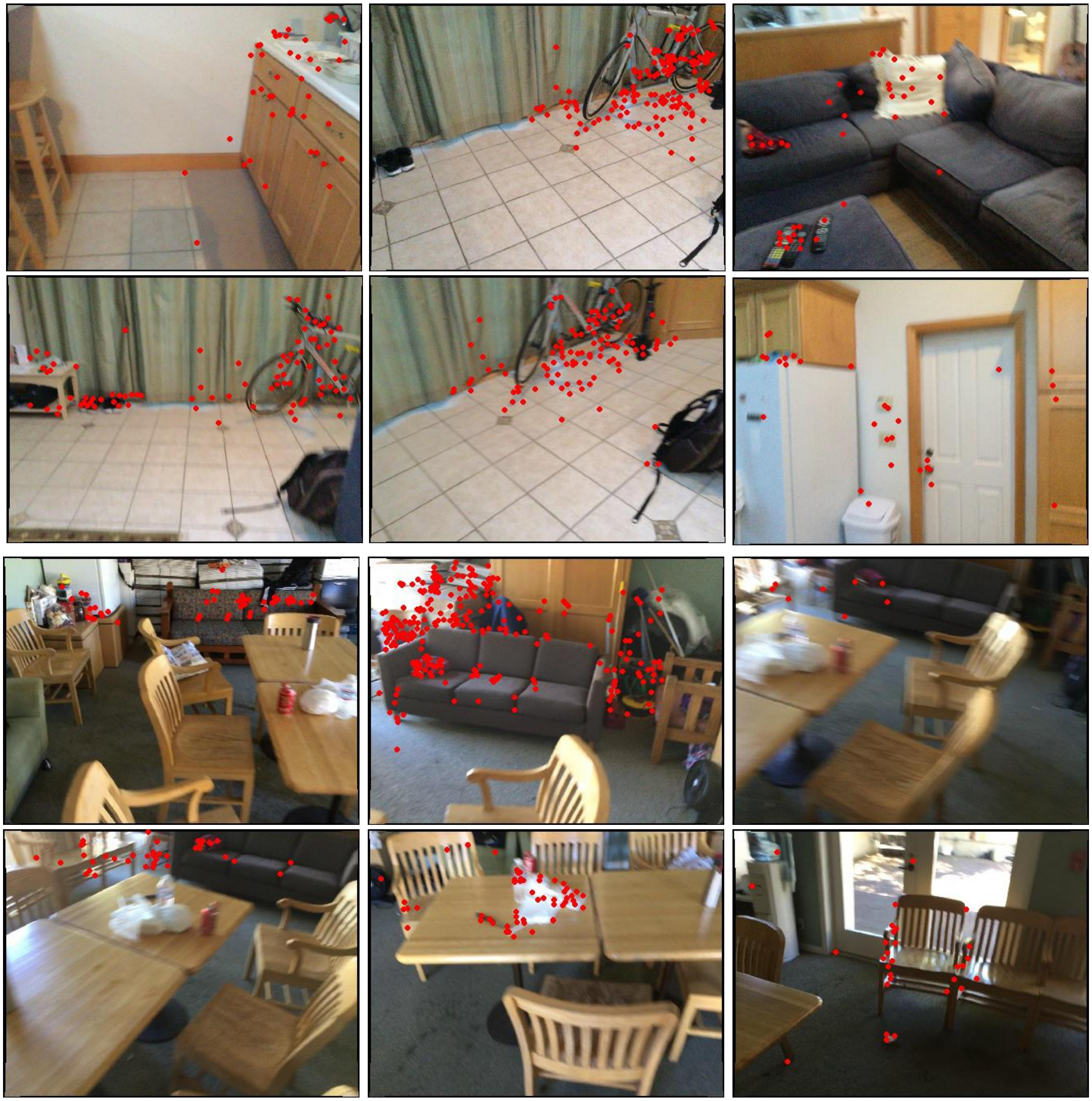

Figure 8. **Visualization for Images with Its 3D observations for PnP in Scannet.**This figure show that because of the blurry problem and textureless region, the 3D-2D correspondences are badly distributed and limited number.

matching with graph neural networks. In *CVPR*, pages 4937–4946. Computer Vision Foundation / IEEE, 2020. 4

[12] Johannes L Schönberger and Jan-Michael Frahm. Structure-from-motion Revisited. In *CVPR*, pages 4104–4113, 2016. 1, 3

[13] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012. 1

[14] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, pages 6209–6218. IEEE, 2021. 4

[15] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: neural implicit scalable encoding for SLAM. In *CVPR*, pages 12776–12786. IEEE, 2022. 4