IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium | 979-8-3503-2010-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IGARSS52108.2023.10282214

CloudLoc-NeRF: Point-cloud Assisted Volume Location for Neural Radiance Fields

Jingyi Cao[1], Yanan You[1], Songzhi Gao[2], Jun Liu[1]*
[1] Beijing University of Posts and Telecommunications, China
[2] The University of Sydney Business School, Australia

ABSTRACT

Realistic rendering results are available to be generated by volume-based neural rendering methods like NeRF. However, the existing schemes take the color vector as the unique supervision information, which leads to ambiguous prediction results of the volume density in the same spatial position through different rendering rays. This problem is common in large-scale scene reconstruction based on remote sensing images taken by UAVs and other equipment. To this end, we contrive to integrate spatial point cloud and multi-view image information, making the sparse point cloud the calibration for key features and regions. Therefore, the CloudLoc-NeRF is proposed. For volume information extraction, the multi-resolution hash coding and voxel are adopted to estimate the district for ray marching and extract volume features efficiently. For the point cloud, annular sampling and plane coding are used to combine image features of the training views and the point cloud. The regions with high feature response in multiple modal data should correspond to the regions with high volume density. In addition, an optimization method based on point cloud density is proposed. The weight parameter of volume density confidence is constructed to symbolize the correlation between density distribution and point cloud density. We verified the performance of our method on NVSF and the wide-area scene reconstruction dataset. Experiments showed that CloudLoc-NeRF accurately expresses the details of the rendered scene and produces better view synthesis results.

Index Terms— NeRF, point cloud, volume rendering, neural network

1. INTRODUCTION

Neural radiance and density field (NeRF) brings incredible rendering effects to 3D scenes. Not only small objects, but more studies have also promoted the application of NeRF to large-scale scenes recently. For example, MegaNeRF [1] and BlockNeRF [2] disassemble wide-area scenes in advance and integrate multiple independent rendering scenes based on position confidence and overlap. Thus, precision rendering of each predictable independent scene remains the crucial foundation for large scene reconstruction. Optimizing the expression of back- and fore-ground of outdoor scenes can ameliorate fuzzy rendering of large-scale scenes [3], because the ambiguity of predicted volume densities from different views is mitigated, as shown in Fig. 1 (a) and (b). In fact, the volume density of a point should have the only value, not affected by the view angle. It enlightens us on whether separating targets or recognizing the beingness of objects in the wide area can also achieve the effect.

*Corresponding to youyanan@bupt.edu.cn

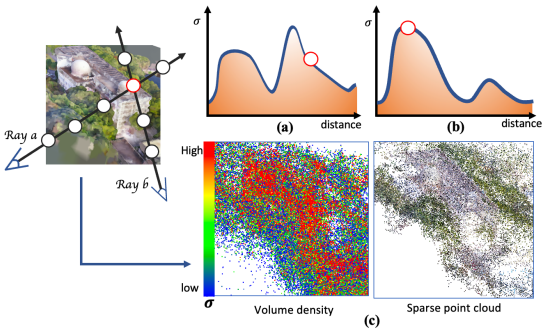

Fig. 1 (a)(b) Predicted density distribution curves along different rays; (c) Comparison of volume density and sparse point cloud

NeRF takes the camera poses and multi-view images as inputs. In engineering applications, in order to obtain camera pose, SfM should be executed in advance, incidentally producing sparse point clouds. It is worth noting that the sparse point cloud provides target existence and key point feature information, which is beneficial to the high-precision rendering of wide-area scenes. Ideally, the distribution of volume density should be similar to the signed distance functions (SDF) [4], that is, high values are on the object's surface and gradually decrease away from the surface. After visualization of the panoramic volume density rendered by the basic NeRF, the expression of sparse point cloud and volume density is similar. As shown in Fig. 1 (c), high volume density is distributed on the surface, corner points, and edges of the object. Unfortunately, the target surface shows a weak volume density. In fact, most current NeRF-related algorithms only use color vectors for supervision. As long as the integral of the volume density and color is consistent with the image pixels, without considering the correctness of each sample point, the model is considered not to need updating. That's not reasonable.

Therefore, our work takes the sparse point cloud as the global reference information to assist in the parameter learning of the volume rendering model, allowing for faster and higher-quality rendering. Specifically, for point sampling in ray marching, volume feature extraction and point cloud feature extraction are carried out synchronously. The probabilities of sampling points belonging to target surfaces and key points are calculated from the point clouds. Finally, the dual features are integrated to optimize volume density expression, so as to generate better view rendering results.

2. METHOD

In this section, we introduce an overview of the proposed network in 2.1. The core technologies for key structures in the network are described in the following sections. Finally, the overall training loss is introduced in 2.5.

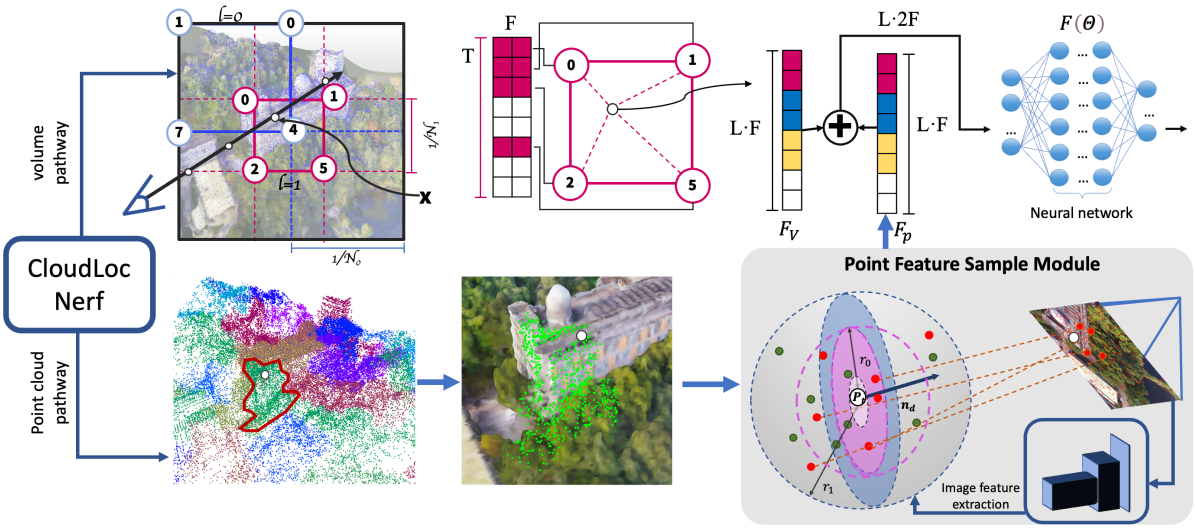

Fig. 2 Overall network of CloudLoc-NeRF

2.1. Network structure

Sparse point cloud provides absolute positioning information for scene construction. Therefore, CloudLoc-NeRF is built, as illustrated in Fig. 2. It combines neural radiance fields with the point cloud, and steers volume density regression based on the point cloud density and volume features. Two pathways and one integration module are designed to complete scene information synergistically.

Volume pathway is inherited from instant neural graphics primitives (instant NGP) [5], which replace normal position encoding of sine(cosine) with multi-resolution hash encoding providing significant speed boosts without accuracy drop.

Point cloud pathway aims to optimize key feature extraction and maintain consistency of volume density and point cloud distribution. Notably, the pre-clustering point cloud is used to reduce the computation burden. In order to obtain the multi-scale receptive field, the annular convolution is used to integrate the point cloud features around the sampling points. Finally, the features of the sample point in ray marching are generated by integrating image and point cloud distribution features.

Feature synergy and volume density optimization make use of the aligned information of volume space and point cloud to conduct original volume density with a neural network. Then, for ensuring the high-density area fits the object surface or key points, the point cloud distribution density is used to calculate density confidence and update volume density. At last, the color value of each sample point is predicted based on the updated volume density and observation direction, and then the integral is executed along the view ray to render the view color.

2.2. Volume pathway: Multi-resolution hash coding

Given the position of the point in the scene (x, y, z) and the direction of observation (θ, ϕ). For M sample points along an observed ray, $\{x_j | j = 1, \ldots, M\}$, neural network $F(\Theta)$ will output (c_j, σ_j) to represent the self-luminous color and volume density from the corresponding view. Then, rendering color C generated by discrete integration is calculated as,

$$C = \sum_{M} \tau_j [1 - exp(-\sigma_j \Delta_j)] c_j, \qquad where\ \tau_j = \exp\left(- \sum_{t=1}^{j-1} \sigma_j \Delta_j \right) \tag{1}$$

where, τ_j represents the probability that the ray does not meet other particles from t_1 to t_j, i.e., volume transmittance. Δ_t is the distance between adjacent samples. Instant NGP, which we inherited, stores features with the multi-resolution hash mapping. Consistent with [5], we use linear interpolation to express eigenvalues of any coordinates in 3D space, and keep the dimension of features (F) with 2 to maintain efficiency.

2.3. Point cloud pathway: Point Feature Sample Module

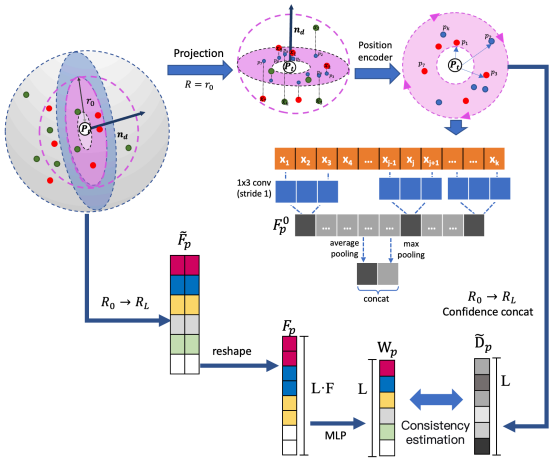

Fig. 3 The detailed structure of the point feature sample module

The wide-area coverage of remote sensing provides huge point cloud distribution, so it is inefficient to compute the whole point cloud. Therefore, Balanced Iterative Reducing and Clustering Using Hierarchies (BRICH), the unsupervised segmentation algorithm, is used to separate weakly correlated point cloud clusters. On the basis, for the sample points along the ray, annular sampling is adopted to obtain associated point clouds at different distances from the central point. In

Consistence with the multi-level hash coding, we apply hierarchical point cloud sampling. Inspired by [6], our work proposed to match L-level search grid scales in hash encoding with l annular sphere space with radius $r = \{\frac{1}{N_l}, \frac{1}{N_{l-1}}, \dots, \frac{1}{N_0}\}$. Within each annular, the contained points are sampled and projected to a uniform plane. Since the observation direction is known, instead of regressing the planar using multiple points, the observation direction of the camera represents the normal vector of the fitting plane, and the sample point p_t is the midpoint of the plane. The regression plane is as follows:

$$\overrightarrow{\boldsymbol{n_d}} = \overrightarrow{\boldsymbol{od}} = (a', b', c') \tag{2}$$

$$a'(x - p_t^{\mathrm{x}}) + \mathrm{b}'\left(\mathrm{y} - p_t^{\mathrm{y}}\right) + \mathrm{c}(\mathrm{z} - p_t^{\mathrm{z}}) = 0 \tag{3}$$

After projection, the point cloud feature vector F_p^i is constructed clockwise by the point cosine value, starting from the nearest point to the sampling point in the annulus. Thereinto, point cloud features include image feature $I(x_1^{od})$ and point cloud coordinate $h(x_1)$, where $I(x_1^{od})$ derived from features projected from point cloud to the observed image. And image features are obtained from the ResNet34 pre-training model generated by key point training. Then concatenation is conducted to integrate point cloud features in the annulus F_p^i.

$$\begin{aligned} &F_p^i = \mathcal{G}\left(\mathcal{H}\left(I\left(x_i^{od}\right), h(x_1)\right), \dots, \mathcal{H}\left(I\left(x_k^{od}\right), h(x_k)\right)\right), \\ &\mathbf{s.t.}\ \ \mathrm{x_i} \in \left\{r_{i-1} < \left\|h(x_i) - h(P_t\ )\right\|_2 < r_i \mid x_i \in C_s,\ P_t \in C_s\right\} \end{aligned} \tag{4}$$

where $\mathcal{H}$ represents feature concatenation in column, $\mathcal{G}$ represents feature concatenation in row, C_s is the s-class point cloud cluster generated by clustering.

Following, average pooling and max pooling are performed on each annular feature F_p^i. After annulus-by-annulus connection and feature flattening, point cloud structural feature F_p whose shape is consistent with the volume rendering feature F_V is obtained.

$$\tilde{F}_p = \mathcal{H}\left[\mathcal{G}\left(mp\left(F_p^0\right), ap\left(F_p^0\right)\right), \dots, \mathcal{G}\left(mp\left(F_p^L\right), ap\left(F_p^L\right)\right)\right] \tag{5}$$

$$F_p = flatten(\tilde{F}_p) \tag{6}$$

2.4. Feature synergy and volume density optimization

The concatenation of features from the volume pathway (F_V) and point cloud pathway (F_p) are inputted into the shallow neural network, which comes up with the available volume density σ_j of the sample point. Besides, the color vector c_j also produce through the neural network.

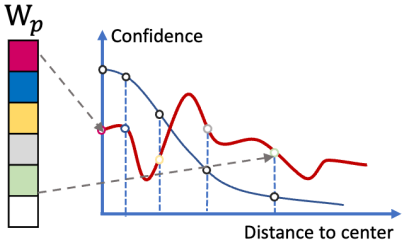

Fig. 4 Diagram of calculation process for weight parameter

In order to optimize the generation of volume density with the point cloud, we conduct to integrate features F_p through MLP, fitting weight density W_p including information on different grids to represent the importance of point clouds. It is deduced that the annular point cloud density is inversely proportional to the distance from the central point, as shown in Fig. 1(c). And the volume density should be proportional to the point cloud density, correspondingly. Therefore, we consider obtaining the similarity weight w_{P_t} by measuring the function relation between W_p (red line in Fig.4) and the symmetry unimodal function with 0 as the axis (blue line in Fig.4).

$$\boldsymbol{w_{P_t}} = \mathrm{Wasserstein}([sigmod]', W_p) \tag{7}$$

Then the weight is used to optimize the volume density. Coincidentally, the derivative of the sigmod function is a unimodal function. For the sake of simplicity, we measure the Wasserstein distance between the weight vector $\boldsymbol{w_{P_t}}$ and the derivative of the sigmod function. The final volume density used for scene modeling is the following,

$$\tilde{\sigma} = \boldsymbol{w_{P_t}} \cdot \sigma \tag{8}$$

2.5. Loss function

Multi-loss is adopted in the training model. Similarly, our method renders the observed color of the specified ray using the ray sampling and integration method, and calculates the color loss L_{volume} by Euclidean distance between the integral color value ($C(r)$) and the color of the view image on the specified pixel ($\tilde{C}(t)$).

$$L_{volume} = \sum_{r \in R} \left\|\tilde{C}(t) - C(r)\right\|_2^2 \tag{9}$$

In addition, for the features extracted from point clouds, the ratio of the number of sparse point clouds which are visible to the view image (red points in Fig. 3) and the number of all correlated point clouds (red and green points in Fig. 3) is taken as the supervision for training. After concatenation through radius, the density confidence vector $\widetilde{D}_p$ of the observation point p_t are generated. Then, we use cross entropy to construct consistency loss $L_{consistence}$, comparing otherness of W_p and $\widetilde{D}_p$.

$$L_{consistence} = \frac{1}{L}\sum_i -\left[\widetilde{D}_p^i \cdot log\left(W_p^i\right) + \left(1 - \widetilde{D}_p^i\right) \cdot log\left(1 - W_p^i\right)\right] \tag{10}$$

Therefore, the overall loss function is demonstrated below. In order to balance loss, λ is set as 0.3 in the experiment.

$$L_{consistence} = L_{volume} + \lambda L_{consistence} \tag{11}$$

3. EXPERIMENT

Experimental setup: We use three UAV data for verification, namely the countryside, academic building, and church. All images were taken by the drone carrying a visible-light camera in aerial perspective. In addition, in order to measure the effect on natural data, the benchmark dataset Tanks & Temple [7] is evaluated as well. Before experiments, ColMap was used to obtain sparse point clouds and estimated camera pose, and BRICH is applied to rapidly segment sparse point clouds. Specifically, the scene is transformed into 30 point cloud clusters.

Quantitative evaluation: To evaluate the performance of our methods, PSNR and SSIM are used to quantitatively estimate the rendered results. Besides, Learned Perceptual Image Patch Similarity (LPIPS) is also adopted, which is more in line with human visual perception standards.

Table I Performance comparison of different NeRF methods

		Tanks & Temple			Large scene	
		Barn	Caterpillar	Countryside	Academic Building	Church
instant NGP	$PSNR\uparrow$	26.6378	23.4297	23.2759	20.2736	20.2341
	$SSIM\uparrow$	0.8374	0.8841	0.5819	0.4443	0.6441
	$LPIPS_{vgg}\downarrow$	0.2992	0.2773	0.4158	0.5175	0.3077
CloudLoc NeRF	$PSNR\uparrow$	28.7457	24.3401	27.3926	23.8602	24.8957
	$SSIM\uparrow$	0.8237	0.8922	0.7680	0.6550	0.7836
	$LPIPS_{vgg}\downarrow$	0.2629	0.2839	0.3109	0.3606	0.2139

As shown in Tab.1, our algorithm achieves better SSIM and LPIPS in large scene reconstruction datasets. Notably, due to a lack of perception of key features and surroundings, the structural features are missed in instant NGP, which causes unsatisfactory indicators in PSNR and LPIPS. In the LPIPS, our model produces the best value 0.2139 for all datasets, which illustrates that the auxiliary point cloud features not only report the existence of the target, but also introduce local features into the model by our designed annular point feature sample module. Moreover, superiority is also evident in small-scale datasets, especially the Barn dataset, which presents a regular rigid body house shape, and target key points show highly consistent with the pre-trained ResNet model. More importantly, our neural system allows the model to focus on the actual scene surfaces while only operating the local point cloud, so that the rendering effect is guaranteed.

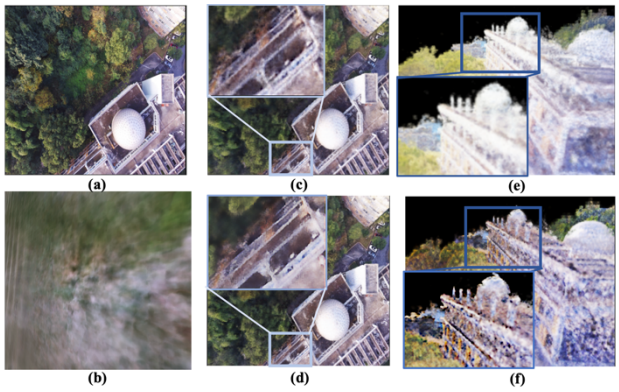

Fig. 5 Rendering effect visualization. (a) Original test view, (b) NeRF, (c) and (e) instant NGP, (d) and (f) CloudLoc-NeRF.

Visual evaluation: In addition, we visualized the rendering results in detail, as shown in Fig.5. (b)(c)(d) are rendered from the overhead view consistent with training data, and (e) and (f) are rendered from a random perspective. Taking the academic building data as an example, NeRF algorithm got an invisible rendering effect; instant NGP could realize the rendering, but the rendered surfaces are uneven. Relatively speaking, Our CloudLoc-NeRF has better scene rendering in general, and gets smoother surfaces and more distinct edges.

4. CONCLUSION

This paper proposes a novel approach named CloudLoc-NeRF for high-quality neural scene reconstruction and rendering. Addressing the lack of structural or detailed information in traditional NeRF rendering results, our work emphasizes synthesizing multi-resolution volume features and sparse point cloud features. The pre-created sparse point clouds are integrated into the scene rendering process through clustering, projection, and convolution operations. The scene representation ability of sample points along ray marching is symbolized through the spatial relation among point clouds and the critical degree of point cloud projection on the two-dimensional view plane. Finally, the volume features are optimized and globally located with the assistance of the point cloud. Experiments verified that CloudLoc-NeRF effectively integrates the point cloud information into the volume optimization process and obtains better render results.

5. ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant 62101060, the BUPT Excellent Ph.D. Students Foundation (CX2022151), and the BUPT innovation and entrepreneurship support program (2023-YC-S004).

6. REFERENCES

[1] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly- Throughs," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 12912–12921. doi: 10.1109/CVPR52688.2022.01258.

[2] M. Tancik *et al.*, "Block-NeRF: Scalable Large Scene Neural View Synthesis," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 8238–8248. doi: 10.1109/CVPR52688.2022.00807.

[3] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "NeRF++: Analyzing and Improving Neural Radiance Fields." arXiv, Oct. 21, 2020. Accessed: Jan. 09, 2023. [Online]. Available: http://arxiv.org/abs/2010.07492

[4] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[5] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–15, Jul. 2022, doi: 10.1145/3528223.3530127.

[6] A. Komarichev, Z. Zhong, and J. Hua, "A-CNN: Annularly Convolutional Neural Networks on Point Clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[7] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Aug. 2017, doi: 10.1145/3072959.3073599.