Eros: Real-Time Dense Mapping Made Easy on Mobile Devices

Yubin Dai, Bin Qian, Yangkun Liu, Yuxuan Yan, Yuanchao Shu
Zhejiang University
China

ABSTRACT

3D reconstruction is crucial for applications such as augmented reality (AR) and autonomous driving. However, state-of-the-art methods often rely on implicit neural representations (INRs), which require significant communication and computational resources, resulting in slow and expensive reconstruction. In this paper, we introduce *Eros*, a novel edge-assisted online 3D reconstruction framework tailored for latency-sensitive mobile applications. *Eros* reduces data sampling, transmission, and processing overhead, and combines explicit and implicit methods into an end-to-end system to minimize reconstruction latency. Experimental results demonstrate that *Eros* reduces reconstruction latency by 28% while consuming only 34% of the network bandwidth of low-latency systems, outperforming current approaches.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**.

KEYWORDS

Dense mapping, 3D reconstruction, Edge computing

ACM Reference Format:
Yubin Dai, Bin Qian, Yangkun Liu, Yuxuan Yan, Yuanchao Shu. 2025. Eros: Real-Time Dense Mapping Made Easy on Mobile Devices. In *The 26th International Workshop on Mobile Computing Systems and Applications (HOTMOBILE '25), February 26–27, 2025, La Quinta, CA, USA.* ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3708468.3711893

The work was partially supported by the National Science Foundation of China (NSFC) under Grants 92467301 and 62293511.

HOTMOBILE '25, February 26–27, 2025, La Quinta, CA, USA

ACM ISBN 979-8-4007-1403-0/25/02
https://doi.org/10.1145/3708468.3711893

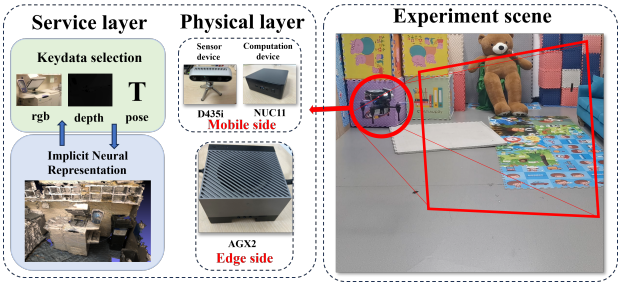


Figure 1: Illustration of our proposed NeRF system.

1 INTRODUCTION

Dense mapping, or 3D reconstruction, models the environment in real-time using camera data, which is essential for applications like augmented reality (AR)[14] and unmanned aerial vehicles (UAV)[3]. It enables accurate environmental reconstruction, supporting latency-sensitive tasks such as path planning and obstacle avoidance for autonomous driving in dynamic environments. In virtual reality (VR), it enhances human-computer interaction, offering more immersive experiences. Real-time 3D reconstruction is crucial for these applications, as it ensures effective and reliable system performance with minimal latency.

Traditional 3D reconstruction methods use either explicit [2, 5, 7, 11] or implicit [6, 8] techniques to represent surrounding environments. Explicit representations, such as point clouds [7], occupancy grids [5], and meshes [11], provide a more direct approach to capture scene information. However, explicit reconstruction necessitates a finer spatial partitioning to achieve high accuracy, which often results in increased memory requirements [17]. For instance, using an Octree for explicit representation on the ModelNet10 at a resolution of 256^3 results in a memory usage of 70GB [10], making it unsuitable for deployment on resource-constrained devices.

To mitigate memory usage while enhancing accuracy, implicit representation techniques [13, 16, 18] that leverage neural radiance fields (NeRF) [8] have been introduced. These methods typically yield more realistic reconstructions and high-quality images by processing new viewpoints through implicit neural networks. However, this approach has its downsides: training neural networks for implicit representations is computationally intensive. Consequently, it is not suitable for deployment on resource-constrained devices.

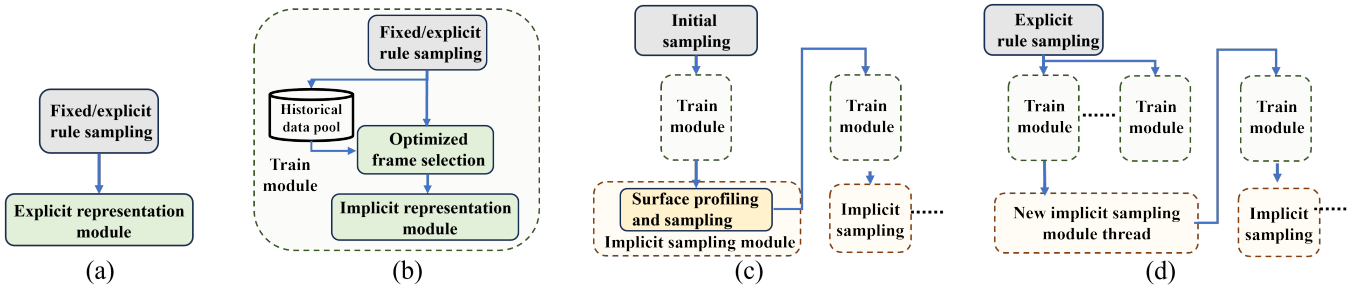


Figure 2: Existing workflow and our proposed; (a) Using explicit methods to guide explicit sampling; (b) Using explicit methods to sample for the implicit neural representation module; (c) Using implicit surface analysis methods to guide sampling for the implicit neural representation module(Active Mapping); (d) Our proposed system that combines explicit and implicit sampling approaches.

And it is also difficult to locate sparse data regions in the environment. Although *H2-Mapping* [6] enables real-time representation on edge devices, it still demands powerful edge devices and neglects the notable transmission overhead in typical mobile-edge setups, assuming data is both sampled and processed on the edge server. No existing systems have considered the long latency when deploying the whole processing pipelines of implicit 3D reconstruction across the mobile and the edge servers. To address this problem, we propose a new 3D reconstruction system named *Eros* (Figure 1), whereby several challenges must be addressed to alleviate the limitations of existing representation methods.

Firstly, from the data perspective, how to quickly identify and sample valuable data on the mobile side: 3D reconstruction involves iterating between sampling new areas from previously unobserved areas within the environment and scene representation. Explicit sampling such as periodic sampling or view field overlap-based sampling [6] are simple heuristic rules that struggle to accurately capture sparse data regions. Other sampling methods that rely on implicit surface analysis [9, 15] can accurately locate sparse regions but encounter issues with long latency. It is worth noting that, unlike active mapping approaches, we do not consider the motion planning of the device. Instead, we focus on how to perform reconstruction in real-time and with high bandwidth efficiency from a given data stream. Integrating explicit and implicit sampling can harness the advantages of each approach; however, it requires careful design for selecting methods, taking into account the trade-off between high-quality data sampling and additional computational overhead.

Secondly, from the computation resource perspective, how to effectively utilize the computing resources on the edge side for data processing: After sampling and transmitting the data to the edge, it is initially stored in a data buffer before being retrieved for further processing. The data cache is typically accessed multiple times to ensure consistent construction performance. However, prioritizing data that enhances overall reconstruction performance poses a significant challenge. To begin with, effectively optimizing performance by utilizing the order of data processing is a complex task. Moreover, since the NeRF network may lose prior scene representations, determining how to appropriately allocate the computational weight for each data item to accelerate performance and establish a robust scene representation becomes a challenging issue.

Thirdly, from the communication perspective, how to minimize communication overhead in a mobile-edge environment while still maintaining reconstruction performance: Current 3D reconstruction methods often overlook transmission overhead, which limits their practical application in real-world settings. The concept of Region of Interest (ROI) has been utilized in video analytics [4] to lower transmission costs. However, obtaining the ROI region in the 2D image coordinate system based on 3D information in a lightweight manner is challenging, particularly when dealing with data such as point clouds or voxels and identifying these ROI regions based on feedback from NeRF networks. Therefore, a new mechanism must be developed to effectively select high-quality data patches while ensuring robust reconstruction performance. To address the aforementioned challenges, our key contributions are as follows:

- We propose *Eros*, to the best of our knowledge, the first system that enables edge-assisted implicit neural 3D reconstruction. *Eros* demonstrates the capability to achieve low-latency reconstruction under both normal and bandwidth-constrained conditions.
- We introduce an innovative hybrid data sampling approach that combines explicit and implicit methods, significantly improving the quality of the sampled data.
- We implement a value-based representation module that efficiently organizes computational resources to process data from the data cache to reconstruct the scene, significantly accelerating the reconstruction process.

- We design a novel ROI-based transmission mechanism for 3D information, thereby reducing unnecessary data training and transmission overhead.

Devices	Implicit representation	Implicit surface profiling	ratio
RTX 2080Ti	0.348	0.765	2.198
RTX 3090	0.344	1.076	3.138
Jetson AGX Orin	0.498	2.849	5.720
Jetson Orin NX	1.083	3.919	3.619

Table 1: Time cost (s) of the implicit representation and implicit surface profiling module.

2 BACKGROUND AND MOTIVATION

In this part, we will introduce the background of 3D reconstruction and analyze its workflow and characteristics.

2.1 3D Reconstruction

3D reconstruction is a technique that creates a 3D model of an environment from data collected within the environment. It involves two modules: 1) *Sampling module* that gathers RGB, depth, and pose information; 2) *Representation module* that reconstructs the scene using this data.

There are typically two types of representation techniques: *explicit* and *implicit* representation. Explicit representation refers to directly constructing 3D objects using various data representations, such as point clouds and voxels. Implicit reconstruction, on the other hand, represents 3D objects using continuous functions, allowing for smooth shapes without explicitly defining their surfaces. This approach leverages implicit neural representations (INRs), where neural networks are used to express these continuous functions. INRs train a neural network to predict light behavior from various angles and generate realistic images from novel viewpoints. It provides a way to represent both light (color) information and spatial (depth) information effectively, with low memory requirements and high reconstruction accuracy. However, it also has drawbacks, as the process requires network training and demands significant computational resources. Therefore, in practical implementations, data is often offloaded to edge servers and choosing which data to send is important.

Previous work on explicit representation methods has typically relied on heuristic sampling approaches, such as timing or view overlap. Similarly, implicit neural reconstruction methods have not specifically developed lightweight strategies for frame selection that can be deployed on mobile devices. Recent advances in implicit surface analysis techniques have enabled effective analysis of sparse regions within surface data [15]. However, as shown in Table 1, the computational complexity makes these methods suitable only for active neural mapping and unsuitable for assessing keyframe transmission under frame refresh rate constraints.

In summary, there are three types of commonly employed 3D reconstruction paradigms, as shown in Figure 2. In the next sections, we perform a detailed performance analysis of the existing methods and highlight their limitations.

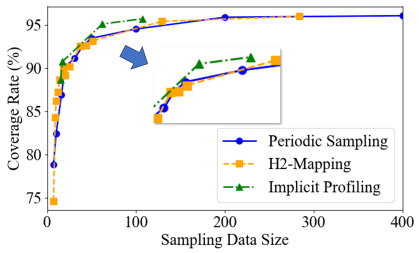


Figure 3: The relationship between sampling data volume and performance in different methods.

2.2 Performance and latency comparison between explicit and implicit sampling

In this section, we analyze the impact of the sampling module on the entire 3D reconstruction workflow. For scene representation, we use a 9-layer MLP [8] for scene representation on the Jetson Orin AGX2 and compare several different sampling methods based on their reconstruction coverage quality on the *Replica* dataset.

As shown in the Figure 3, when sampling 0-30 data samples, there is no clear difference in the coverage rate between the explicit and implicit methods. Subsequently, the implicit profiling module requires nearly 120 data samples to achieve coverage rates of 96%, while the explicit methods require over 200 data samples for similar performance. This demonstrates that implicit profiling can effectively identify high-value data, resulting in greater sample efficiency.

Despite its advantages, the surface profiling process following the representation stage is computationally expensive. During this period, the NeRF networks utilize the reconstructed mesh to identify sparse data regions, resulting in high computational demands. We present a benchmark in Table 1 showing the latency required for both representation and surface profiling across different devices. It is evident that on all devices, the time taken for one round of the surface profiling process exceeds that of one round of implicit representation by more than two times. This presents an opportunity to incorporate explicit sampling during the surface profiling process.

2.3 Communication and computation overhead for implicit representation

After sampling, valuable data is sent to the edge server for further NeRF training. We record the data buffer on both the mobile device and the edge end using different explicit sampling modes, including periodic sampling at rates of 5 FPS and 2 FPS, as well as view overlap rate-based sampling with overlap ratios of 0.8 and 0.9, as mentioned in subsection 2.1.

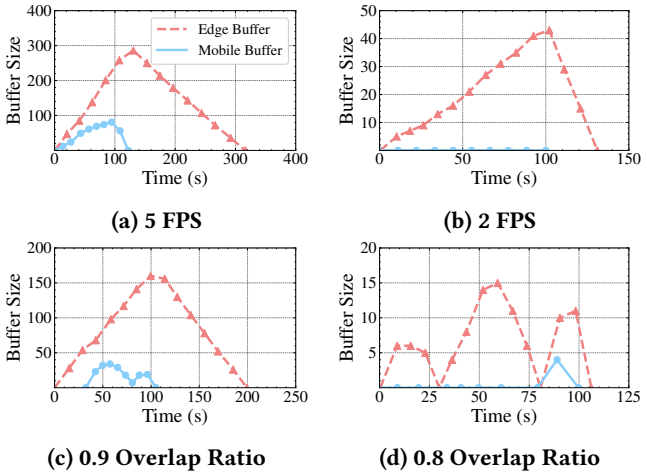


Figure 4: Communication and processing data flow under various sampling modes. (vertical axis unit:frames)

In Figure 4, we can find that in various sampling modes, there are often data processing buffers on both the sending side and the receiving side, especially when we need high FPS or view overlap rates. The data flow is mainly blocked on the edge. The data buffer at the edge side provides us with a cue to prioritize the transmission and processing of high-value data for reconstruction.

Opportunities for improvement: After determining the new data sent to the edge device in this round, due to the forgetfulness of implicit representation [15], some selected data from the historical dataset are combined with new data as input for training the implicit reconstruction module network. This makes it worthy to consider prioritizing which data to select for retraining to ensure effective utilization of computational resources.

3 DESIGN OF *EROS*

3.1 System overview

We design *Eros*, an edge-assisted NeRF system for mobiles, achieving low latency and bandwidth costs. *Eros* encompasses two modules as shown in Figure 5 and we consider system optimization attempts from two dimensions: keyframe selection, communication and computation optimization.

3.2 Keyframe selection

3.2.1 Local Mapping and Keyframe Selection. We first map the depth image into 3-dimensional space by applying a coordinate transformation of the camera's pose. We skip pixels that exceed our set maximum depth, as depth information beyond this distance often has significant noise. After this, we discretize the point cloud into a 3-dimensional voxel space based on a specific voxel grid size. We consider whether we can select a frame as a keyframe for processing based on whether the proportion of voxels in the current frame that are not present in the historical map exceeds a certain threshold.

3.2.2 Quick Implicit Surface Profiling. Based on Active Neural Mapping [15], we explore whether surface profiling methods can be used to identify sparse regions in small areas of the data, while still ensuring that explicit methods are employed to sample and process larger regions effectively. After identifying low-quality surface areas through analysis, we then return their position, direction, and area size to the mobile end. Upon receiving the evaluation results from the network, the mobile device retrieves the corresponding matching data from the local unsent data cache and retransmits it.

3.3 Communication and computation optimization

3.3.1 Value-based Ranking. Unlike the traditional first-sample, first-transmit, first-train approaches, we try to prioritize sending data groups with higher reconstruction value and evaluate each data group's value.

3.3.2 Optimized-frames Organization. Similar to H_2−*Mapping*, all voxels will be repeatedly covered multiple times to select the optimized frames for each training iteration. But we struggle to choose loss-high voxels for one additional training round. This approach increases the training iterations for regions with high loss, allowing more computational resources to be used efficiently.

3.3.3 ROI-based Transmission. After keyframe selection and low-quality region filling, we obtain a series of data sets to be sent to the edge, including depth, color images, and the corresponding poses with timestamps. However, there is still a lot of redundant information among these data sets, such as noise regions in the depth maps beyond a certain threshold, as discussed before in *3.2.1*, and unavoidable overlap between frames, among other issues. Therefore, we use voxel information and sparse regions to mask the selected keyframes for ROI-based transmission.

4 EVALUATION

4.1 Evaluation settings

We use the Intel Next Unit of Computing (NUC) with 16GB of memory and Ubuntu 18.04 as the mobile side, and use the Jetson Orin AGX2 (64GB) as the edge side. The mobile side and the edge side are connected in Robot Operating System(ROS). We utilize implicit representation for the 3D reconstruction task, employing the Replica dataset [12], a popular virtual dataset, and the ScanNet dataset [1], a widely adopted real-world dataset. We use the base representation algorithm same as *H2-Mapping* [6]. To simulate scenarios in low bandwidth conditions, we limit the bandwidth to 2 MB/s by applying a router port bandwidth constraint.

Baselines: 1) Periodical sampling and representation: a sampling method that samples data with a fixed time interval

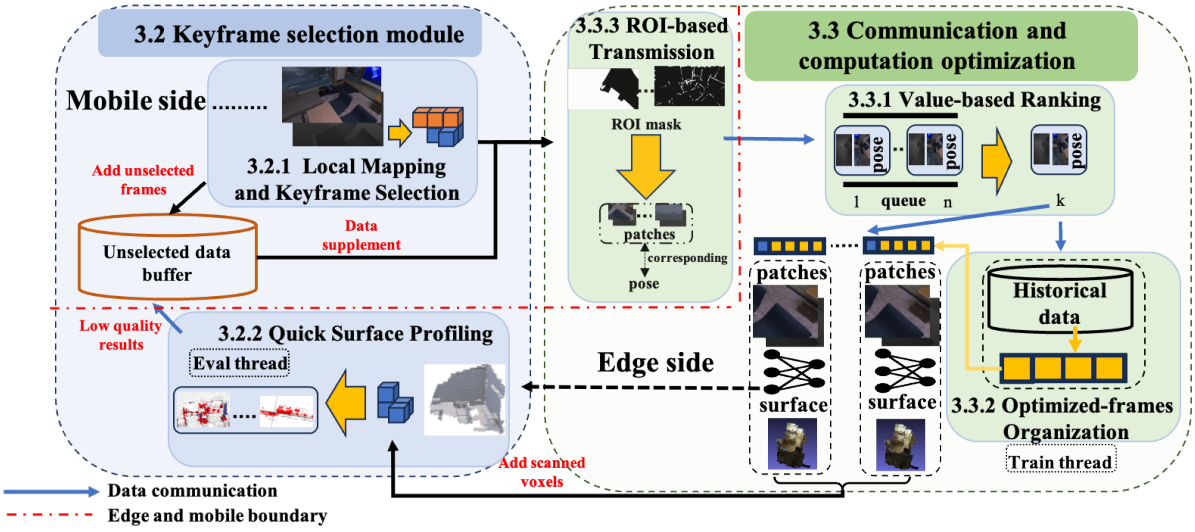


Figure 5: System Overview of *Eros*

for scene representation. 2) *H2-Mapping* [6]: we use the same keyframe sampling strategy that samples data based on the field overlap ratio between two consecutive keyframes. 3) *Implicit Profiling*: In particular, we introduce a sampling method guided solely by implicit surface analysis as a baseline.

4.2 Overall performance

In this experiment, we validated the performance of our *Eros* under different performance requirements and environmental conditions. For Figure 6a, Figure 6c and Figure 6d, we used the ScanNet dataset to simulate a data stream, and the total playback time of the data stream is set to 40 seconds. For Figure 6b, we used the real-world testbed to collect data. We recorded the performance changes from the start of reconstruction until the end of sampling. In the first two experiments, we set high bandwidth between the mobile side and the edge server, then performed reconstruction tasks using two different mechanisms: a) prioritizing high coverage rate, b) prioritizing low latency. The results are shown in Figure 6a and Figure 6b. We also show the performance of *Eros* under low bandwidth conditions in Figure 6c.

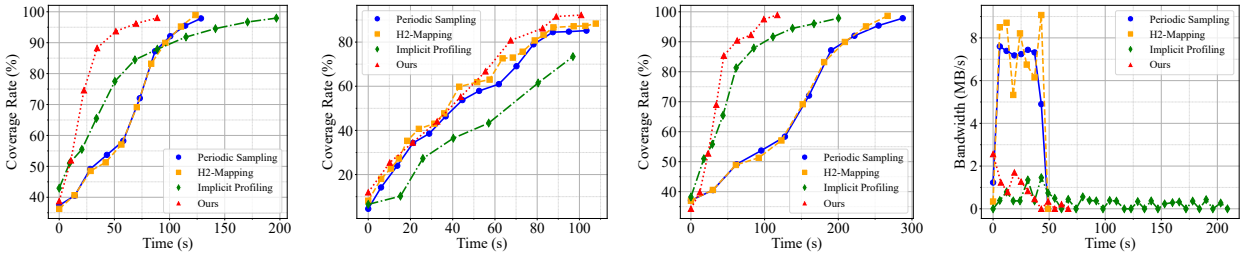
For the first experiment, we use the 5 FPS periodic sampling method and *H2-Mapping* with a 0.9 view overlap rate. A more detailed bandwidth consumption is shown in Figure 6d. We can know that the time required by our system to reach 98% coverage is near 72% that of all the baselines from Figure 6a. Our system can achieve low latency with high coverage requirements. Compared to methods that rely solely on implicit profiling, although there is some bandwidth pressure, it clearly offers the advantage in terms of latency. It can also be observed from Figure 6d that our system can only consume 34% of the maximum bandwidth compared to the other two baselines. This is because we use an explicit and implicit sampling framework to ensure that only valuable frames are selected, and we use ROI-based data transmission to remove intra-frame redundant pixels.

In Figure 6b, we conduct our study on a real-world test bed to demonstrate the performance of our low-latency reconstruction system. In this experiment, we set the periodic sampling to 1.25 FPS and *H2-Mapping* to a view overlap rate of 0.35. Correspondingly, we reduced the explicit sampling rate in our system to 0.6 to meet real-time requirements. By the end, our system achieves a coverage rate of 92.256%, while the baselines only reach below 89%. It shows that our system can prioritize the usage of valuable data for transmission and processing to enable real-time reconstruction.

Finally, we demonstrated that our system is capable of maintaining good performance under low bandwidth conditions. We can see in Figure 6c that the latency of our system is near 58% of the baselines. Our system is almost unaffected by the reduction in bandwidth, while the latencies of the other two baselines increase significantly. This is mainly because our hybrid sampling module and ROI-based data transmission module greatly reduce bandwidth consumption, invariant of the network conditions within the environment.

5 FUTURE DIRECTIONS

Our system demonstrates the feasibility of edge-assisted dense mapping and reveals several promising avenues for future research. First, robustness remains a key challenge, prompting the need for adaptive modules that can dynamically respond to varying network and resource conditions while maintaining optimal performance. Second, task offloading warrants deeper exploration: since dense mapping depends on high-quality localization, developing synergistic offloading strategies for both tasks within a mobile-edge-assisted framework is essential. Finally, extending 3D reconstruction for richer scene understanding stands out as a



(a) Prioritize high coverage rate, high bandwidth **(b) Prioritize low latency, high bandwidth(Real-world)** **(c) Prioritize high coverage rate, low bandwidth(2MB/s)** **(d) Bandwidth consumption for Figure 6 (a)**

Figure 6: *Eros*'s performance with different bandwidth and optimization goals.

particularly exciting direction, opening up new possibilities for advanced applications in robotics, augmented reality, and beyond.

6 CONCLUSION

In this paper, we introduce *Eros*, an edge-assisted 3D reconstruction system that features fast reconstruction, high coverage rate, and low bandwidth usage. In *Eros*'s design, we first explore the opportunity to efficiently sample data from the data stream on the mobile side. Then, we prioritize the reconstruction process with carefully selected high-value data as well as historical data. In order to reduce the bandwidth consumption, we further develop ROI-based data transmission specially designed for 3D data samples. We prototype *Eros* with a mobile-edge system and validate its real-time performance in various scenarios with different requirements.

REFERENCES

[1] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.

[2] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1.

[3] Chen Feng, Haojia Li, Fei Gao, Boyu Zhou, and Shaojie Shen. 2023. PredRecon: A Prediction-boosted Planning Framework for Fast and High-quality Autonomous Aerial Reconstruction. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*.

[4] Ali Hojjat, Janek Haberer, Tayyaba Zainab, and Olaf Landsiedel. 2024. LimitNet: Progressive, Content-Aware Image Offloading for Extremely Weak Devices & Networks. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services (Mobisys)*.

[5] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. 2013. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous robots* 34 (2013), 189–206.

[6] Chenxing Jiang, Hanwen Zhang, Peize Liu, Zehuan Yu, Hui Cheng, Boyu Zhou, and Shaojie Shen. 2023. H _{2}-Mapping: Real-time Dense Mapping Using Hierarchical Hybrid Representation. *IEEE Robotics and Automation Letters* (2023).

[7] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. 2023. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*.

[8] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[9] Yunlong Ran, Jing Zeng, Shibo He, Jiming Chen, Lincheng Li, Yingfeng Chen, Gimhee Lee, and Qi Ye. 2023. Neurar: Neural uncertainty for autonomous 3d reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters* 8, 2 (2023), 1125–1132.

[10] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. 2017. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

[11] Fabio Ruetz, Emili Hernández, Mark Pfeiffer, Helen Oleynikova, Mark Cox, Thomas Lowe, and Paulo Borges. 2019. Ovpc mesh: 3d free-space representation for local ground vehicle navigation. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*.

[12] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* (2019).

[13] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. 2021. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*.

[14] Jamie Watson, Sara Vicente, Oisin Mac Aodha, Clément Godard, Gabriel Brostow, and Michael Firman. 2023. Heightfields for Efficient Scene Reconstruction for AR. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

[15] Zike Yan, Haoxiang Yang, and Hongbin Zha. 2023. Active Neural Mapping. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*.

[16] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. 2022. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*.

[17] Xingbin Yang, Liyang Zhou, Hanqing Jiang, Zhongliang Tang, Yuanbo Wang, Hujun Bao, and Guofeng Zhang. 2020. Mobile3DRecon: Real-time monocular 3D reconstruction on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics* 26, 12 (2020), 3446–3456.

[18] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. 2022. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.