Neural Implicit Representation based SLAM techniques: Comparative Study

Vineet Singh
University of Maryland
College Park
vsingh03@umd.edu

Shantanu Parab
University of Maryland
College Park
sparab@umd.edu

Abstract

This project explores the challenges of traditional Simultaneous Localization and Mapping (SLAM) techniques in robotics and investigates Neural-based SLAM approaches to address scalability, robustness, and adaptability issues. Five prominent neural SLAM methods—iMap, NICE-SLAM, Co-SLAM, Nerf SLAM, and vMAP—are comparatively evaluated for computational efficiency and suitability for resource-constrained robots. Our study aims to identify the most efficient and effective neural SLAM method, striking a balance between accuracy and robustness. Through extensive experiments, we demonstrate the practical feasibility of deploying neural SLAM on resource-limited platforms, contributing to the advancement of robotics in diverse and challenging environments.

1. Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental problem in robotics, where an agent concurrently builds a map of its surroundings while estimating its own pose (position and orientation) within the map. It plays a crucial role in enabling robots to navigate autonomously in various environments. As robots become increasingly sophisticated and operate in more complex scenarios, the need for efficient and robust SLAM techniques becomes paramount. Traditional SLAM approaches rely on various hand-crafted features and probabilistic estimation techniques. However, these methods often suffer from limitations in scalability, robustness, and accuracy, particularly in complex, dynamic environments. Reliance on hand-crafted features limits performance in diverse environments with varying textures and lighting conditions. High computational cost restricts real-time applications, especially for large-scale environments. It also faces difficulty incorporating various sensor data modalities (e.g., LiDAR, cameras) which hinders map accuracy and robustness. Also, the lack of adaptability to dynamic environments with moving objects and changing scene structure is a limitation of traditional SLAM techniques. To address these challenges, Neural networks (Neural Implicit techniques) are used and are classified as Neural-based SLAM.

Neural networks have emerged as a powerful tool for SLAM, demonstrating potential for improved accuracy and adaptability. These offer several advantages over traditional SLAM methods which are: 1) Feature learning: Ability to learn robust features directly from sensor data, adapting to diverse environments. 2) End-to-end training: Efficiently learn the entire SLAM pipeline from raw sensor data to pose and map predictions. 3) Data fusion: Integrate information from multiple sensors seamlessly, improving map accuracy and scene understanding. 4) Dynamic scene modeling: Ability to represent and reason about dynamic environments with moving objects.

However, the computational demands of many existing neural SLAM approaches hinder their practical implementation on resource-limited platforms. To address this challenge, a comprehensive evaluation of different neural SLAM methods is essential to identify those that are most efficient and effective for robots with limited computing resources.

This project presents a comparative study of five prominent neural SLAM approaches: iMap [1], NICE-SLAM [2], Co-SLAM [3], Nerf SLAM [4], and vMAP [5]. We analyze their strengths and weaknesses, focusing on their computational efficiency and suitability for resource-constrained robots. Our investigation aims to identify the neural SLAM method that offers the optimal balance of accuracy, robustness, and efficiency for real-world robotic applications. Through extensive experiments and comparisons, we demonstrate the practical feasibility of deploying neural SLAM on resource-limited platforms while maintaining acceptable performance.

2. Related Work

Neural networks have revolutionized the field of Simultaneous Localization and Mapping (SLAM) in recent years due to their ability to learn robust features directly from sensor data and handle complex environments. This section provides an overview of existing neural SLAM approaches, highlighting their strengths and weaknesses.

Neural Field SLAM Neural fields have gained significant popularity as versatile and precise representations of entire scenes [6, 7, 8, 9]. In the context of real-time Simultaneous Localization and Mapping (SLAM) systems, iMAP [1] pioneered the integration of these representations. It introduced a groundbreaking approach utilizing a basic Multilayer Perceptron (MLP) network, incrementally trained with depth data from RGB-D sensors. This method successfully achieves real-time representation of 3D scenes at room scale. It produced watertight reconstruction while completing even the hidden areas of the scene. While effective in small environments, iMap suffers from scalability limitations and limited data fusion capabilities. NICE-SLAM [2] proposes a real-time and scalable dense RGB-D SLAM system that leverages hierarchical feature grids and neural implicit representations to handle large-scale scenes. NICE-SLAM demonstrates predictive ability, robustness in challenging scenarios, and competitive performance in both mapping and tracking through extensive evaluations on diverse indoor RGB-D sequences. NICE-SLAM demonstrated good performance in large-scale environments and incorporates depth data but struggles with dynamic scenes and lacks real-time capabilities.

Recent Neural SLAM Advancements include CO-SLAM [3] and Nerf SLAM [4]. Co-SLAM introduces a novel real-time SLAM framework by combining joint coordinate and sparse grid encoding for enhanced camera tracking and high-fidelity mapping, leveraging the advantages of smoothness from coordinate encodings and the optimization speed of sparse feature encodings. Notably, Co-SLAM achieves state-of-the-art performance in camera tracking and 3D reconstruction through global bundle adjustment, outperforming NICE-SLAM and iMAP while maintaining real-time efficiency across various datasets. Nerf SLAM pioneered scene reconstruction pipeline that integrates the strengths of dense monocular SLAM and hierarchical volumetric neural radiance fields. This approach enables the construction of precise radiance fields from sequential images, operating in real-time without the need for input poses or depths. It leverages neural radiance fields (NeRF) to represent the environment as a continuous function, enabling pose estimation and high-fidelity 3D reconstruction.

Our final paper for this comparative study is vMAP [5] which leverages depth information from RGB-D sensors to enhance the accuracy of neural field models for 3D object geometry. Depth-guided sampling optimizes point distribution along rays, focusing on points near surfaces for improved geometry representation. The network utilizes a binary indicator for object visibility and employs continuous occupancy fields for termination probability along rays. Training objectives are limited to pixels within object bounding boxes and masks. The disentangled representation facilitates versatile manipulations, including novel view synthesis and fine-grained object-level modifications such as shape and texture changes.

3. Methodology

This section provides a detailed description of each compared neural SLAM method, including its architecture, objective function, training process, datasets, and evaluation metrics.

iMAP, a groundbreaking real-time Simultaneous Localization and Mapping (SLAM) system utilizing an Implicit Neural Scene Representation (INSR) through a Multilayer Perceptron (MLP). Unlike traditional methods, iMAP dynamically trains the network in live operation without relying on prior training data. The system follows a keyframe-based structure reminiscent of PTAM and operates through tracking and mapping processes. The tracking process aligns live RGB-D observations with MLP predictions, while the mapping process optimizes keyframe poses and continuously improves the MLP using a set of historic keyframes. The dynamic sampling of informative RGB-D pixels reduces geometric uncertainty, achieving real-time speed with Python and PyTorch on a single desktop CPU/GPU system. iMAP efficiently represents scenes with continuous and adaptive resolution, demonstrating remarkable interpolation capabilities for complete, watertight reconstruction. With competitive tracking performance on the TUM RGB-D dataset, iMAP outperforms standard dense SLAM systems with a significantly smaller memory footprint, showcasing its efficacy in real-world scenarios.

Key contributions include iMAP being the first dense real-time SLAM system using an Implicit Neural Scene Representation, allowing joint optimization of a full 3D map and camera poses. The paper also highlights the ability to incrementally train the implicit scene network in real-time, facilitated by automated keyframe selection and loss-guided sparse active sampling. The system incrementally optimizes the network weights and camera poses based on actively sampled measurements. Two concurrent processes, tracking and mapping, run simultaneously. Tracking optimizes the pose from the current frame with respect to the fixed network, while mapping jointly optimizes the network and camera poses of selected keyframes, chosen based on information gain.

The implicit scene neural network utilizes an MLP with Gaussian positional embedding, allowing for efficient optimization of the embedding matrix. The rendering engine, inspired by NeRF and NodeSLAM, queries the scene network to obtain depth and color images. The joint optimization involves minimizing geometric and photometric errors for selected rendered pixels in keyframes. Active sampling is employed to render and optimize only a sparse set of random pixels, guided by loss statistics. Keyframes are selected based on information gain, and a bounded keyframe selection process is implemented to manage computational resources efficiently. The approach is demonstrated to achieve real-time camera tracking, continuous network optimization, and adaptive keyframe selection, showcasing its potential for robust and efficient scene mapping. The parallel implementation of the SLAM formulation in PyTorch, fully leveraging multi-processing, ensures online operation with hand-held RGB-D cameras.

NICE-SLAM, a novel dense RGB-D SLAM system, emerges as a sophisticated solution for navigating large-scale scenes while upholding predictive accuracy. The crux of this system lies in its hierarchical scene representation, ingeniously employing four feature grids and corresponding decoders to capture both the geometry and appearance of the environment. The hierarchical structure comprises coarse, mid, and fine-level features, facilitating a meticulous approach to scene reconstruction. The mid and fine-level grids play a pivotal role in this process, adopting a coarse-to-fine strategy that optimizes mid-level features first, followed by the nuanced refinement of fine-level details. The separate coarse-level grid predicts high-level scene geometry, providing approximate occupancy values even for unobserved regions. Pre-training the decoders, particularly for mid and fine-levels, ensures stable and consistent geometry learning. The color representation is encoded using an additional feature grid and decoder, enhancing tracking signals through the rendering of RGB images.

This advanced optimization scheme is orchestrated in stages, aiming to minimize geometric and photometric losses through an alternating fashion for selected keyframes. Geometric losses involve L1-norm differences between observations and predicted depths at coarse and fine levels, while photometric losses capture differences in rendered and observed color values. The optimization unfolds sequentially, with mid-level features refined first, followed by the inclusion of fine-level details. A local bundle adjustment then harmonizes the optimization of feature grids at all levels, color decoders, and camera extrinsic parameters for selected keyframes. Importantly, this multi-stage optimization framework ensures efficient convergence, with the higher-resolution appearance and fine-level features building upon the already refined mid-level geometry.

The system's execution is streamlined through a parallelized approach, employing different threads for specific tasks. Threads dedicated to coarse-level mapping, mid & fine-level geometric and color optimization, and camera tracking operate concurrently, expediting the optimization process. Camera tracking is formulated to optimize camera poses in tandem with scene representation optimization, enhancing the system's adaptability to sudden frame loss or rapid camera movement. Notably, a modified geometric loss is introduced during camera tracking, down-weighting uncertain regions in the reconstructed geometry, further fortifying the system against potential challenges.

Beyond the optimization intricacies, NICE-SLAM implements a judicious keyframe selection strategy, reminiscent of other SLAM systems but tailored for its unique hierarchical representation. The selection process incorporates only keyframes with visual overlap with the current frame, ensuring that the optimization of scene geometry focuses on relevant, static features outside the current view. This strategy not only refines the system's efficiency but also contributes to its resilience in navigating dynamic environments, as demonstrated by the exclusion of pixels with large depth/color re-rendering losses attributed to dynamic objects during tracking. The system's resilience to dynamic elements, robust keyframe selection strategy, and meticulous optimization process underscore its potential as a cutting-edge solution for real-time, large-scale spatial understanding.

CO-SLAM tackles the challenge of real-time SLAM by jointly optimizing camera poses and a neural scene representation. The neural representation employs a combination of coordinate and parametric encoding to balance the advantages of coherence provided by coordinate encodings and the optimization speed of sparse parametric encodings. The system initializes with training iterations on the first frame, optimizing camera poses using a constant-speed motion model. Subsequent frames involve joint optimization, where a small fraction of pixels is sampled and globally adjusted for both scene representation and camera poses.

To enhance the efficiency of the system, a joint coordinate and parametric encoding approach is introduced. This method integrates One-blob encoding for spatial coordinates with a hash-based multiresolution feature grid. The multiresolution grid enables fast convergence, efficient memory use, and effective hole filling, critical for online SLAM. Depth and color rendering are performed by integrating predicted values along sampled rays, with a depth-guided sampling strategy to balance accuracy and computational speed.

The tracking and bundle adjustment phases involve minimizing objective functions with respect to learnable parameters and camera parameters. To ensure accurate, smooth reconstructions, approximate Signed Distance Function (SDF) and feature smoothness losses are applied.

Feature	vMAP	NeRF SLAM	Co-SLAM	NICE-SLAM	iMAP
Tracking	Pose graph optimization with volumetric updates	Pose estimation, NeRF update	Collaborative pose and NeRF optimization	Factor graph optimization of poses and NeRFs	Occupancy grid update based on NeRF predictions
Optimization algorithm	Gauss-Newton	Levenberg-Marquardt (LM)	Differentiable objective, Adam optimizer	Alternating optimization with marginalization	Iterative update based on prediction error
Mapping approach	Update the volumetric occupancy grid and appearance map	Integrate new poses and NeRF updates into a global map	Incremental fusion of poses and NeRF updates from each view	Integrate optimized poses and NeRFs into a factor graph map	Build a spatial occupancy map over time
Key strengths	Efficient memory usage, good for large-scale scenes	Efficient pose refinement, accurate NeRF updates	Improved tracking and NeRF learning, reduced ambiguity	Robust pose estimation, efficient inference, handles missing data	Fast mapping, memory-efficient, suitable for resource-constrained devices
Weaknesses	Limited detal compared to NeRF, less accurate for complex scenes	Computational cost, memory requirements	Increased complexity, coordination challenges	Less flexibility compared to pure NeRF approaches	Lower accuracy compared to dense NeRF reconstruction

Table 1: Qualitative Comparative Study of methods under consideration

Depth-guided sampling is favored over importance sampling for efficiency.

In camera tracking, the system optimizes the camera-to-world transformation matrix, and for bundle adjustment, it goes beyond traditional keyframe selection. Co-SLAM avoids storing full keyframe images, relying on sparse representations (around 5% of pixels) for each keyframe. This innovative approach allows for more frequent insertion of new keyframes, maintaining a larger keyframe database. Joint optimization involves random sampling of rays from the global keyframe list for alternating optimization of scene representation and camera poses. This eliminates the need for keyframe selection, achieving improved robustness in camera pose optimization with minimal additional computational cost. The entire system demonstrates state-of-the-art performance in camera tracking and 3D reconstruction across various datasets while maintaining real-time efficiency.

NeRF SLAM integrates dense monocular Simultaneous Localization and Mapping (SLAM) outputs into the supervision of a neural radiance field, aiming for real-time performance. Utilizing Droid-SLAM as the tracking module, the system computes dense depth maps and poses, incorporating uncertainty estimates. The tracking loop involves solving a linear least-squares problem, optimizing camera poses using the Schur complement of the Hessian, and updating depths iteratively. Marginal covariances for depths and poses are calculated efficiently. The tracking module continuously refines these estimates and feeds induced optical flow back into the system. In the mapping backend, a probabilistic volumetric NeRF is employed. However, supervising the neural volume with noisy depth maps from SLAM can lead to biased results. Inspired by previous work, the system leverages depth uncertainty to weight the depth loss for more robust supervision. The mapping loss includes terms for both depth and color, optimizing neural parameters and poses simultaneously. The architecture consists of parallel tracking and mapping threads, with communication occurring only when generating new keyframes. The mapping thread renders reconstructions for interactive visualization.

In the tracking module, Droid-SLAM extracts dense optical flow between frames, facilitating efficient dense bundle adjustment for 3D geometry. The system optimizes camera poses and per-pixel inverse depths using a linear least-squares approach. Covariances of depth maps and poses are efficiently computed from the Hessian structure. The mapping backend employs a probabilistic volumetric NeRF, addressing the noise in SLAM depth maps by weighting depth loss with uncertainty. The mapping loss combines depth and color terms, with simultaneous optimization of neural parameters and poses. The architecture maintains real-time performance by running tracking and mapping threads in parallel, with minimal communication between them. The tracking thread continuously refines estimates, generating new keyframes based on optical flow thresholds. The mapping thread ensures accurate and interactive visualization of the reconstruction. The system's overall design aims for a robust fusion of dense SLAM and neural radiance field training, demonstrating the feasibility of real-time 3D scene reconstruction with enhanced accuracy and efficiency.

vMAP proposed an approach for object initialization and association and involves associating frames with densely labeled object masks, either provided or predicted using a 2D instance segmentation network. Object association is based on semantic and spatial consistency criteria between consecutive frames. If satisfied, objects are considered the same instance; otherwise, a new model is initialized. 3D object bounds are estimated using the object's 3D point cloud, depth map, and camera pose. Object supervision is

Methods	Reconstruction Comparison			
	Depth L1[cm]⬇	Acc.[cm] ⬇	Comp.[cm]⬇	Comp. Ratio (%) ⬆
iMAP	5.9	4.2	5.3	80.4
NICE-SLAM	3.1	2.7	3.2	89.7
Co SLAM	2.83	2.1	2.9	92.9
NeRF-SLAM	2.58			
vMAP	2.46	3.2	2.4	93.3

Table 2: Reconstruction Comparison on the average of 8 scenes of the Replica Dataset

applied within 2D bounding boxes, optimizing training efficiency. Vectorized training is employed, utilizing small network designs and PyTorch's optimized operations for simultaneous batch training of multiple object models, enhancing GPU resource utilization.

The method incorporates depth information from RGB-D sensors to enhance the accuracy of neural field models for 3D object geometry. Depth-guided sampling involves stratified sampling along rays, with points near the surface aiding in accurate geometry representation. The network excludes viewing direction, employing a binary indicator for object visibility. Occupancy probability is parameterized as a continuous field, guiding termination probability along rays. Training objectives are confined to pixels within an object's 2D bounding box and its mask. The disentangled representation allows manipulation within estimated 3D object bounds, enabling novel view synthesis and fine-grained object-level modifications like shape and texture changes.

Datasets and Evaluation Metrics

Datasets used for comparison include Replica [1] and TUM RGB-D [10]. Evaluation metrics include:

- Absolute Trajectory Error Root Mean Squared Error (ATE_RMSE): Measures the average Euclidean distance between the predicted and ground truth robot poses, with lower values indicating better performance in SLAM systems.
- Accuracy (cm): the average distance between sampled points from the reconstructed mesh and the nearest ground-truth point [1]
- Completion (cm): the average distance between sampled points from the ground-truth mesh and the nearest reconstructed [1]
- Completion Ratio (<5cm %): the percentage of points in the reconstructed mesh with Completion under 5 cm
- L1 depth error between the estimated and the ground-truth depth maps as a proxy for geometric accuracy (Depth L1). [1]
- Computational Efficiency: Measures the execution time and memory consumption of each method on a specific hardware platform.

4. Results and Analysis

For comparing the models, we decided to use the Replica dataset [1] for analyzing the scene reconstruction and the TUM-RGBD dataset [10] for camera trajectory. Below we explain both the datasets briefly and why we chose them for our comparison. The models were first compared on a qualitative basis, comparing their mapping, planning, and optimization strategies. Further to that, we did a quantitative study for the scene reconstruction and the localization of the camera. The localization study is done on the TUM-RGBD dataset on its 3 classes and Table 3 shows the results of the analysis. The reconstruction quality analysis is done on the Replica dataset on all its 8 classes and Table 2 presents the average error of across the 8 classes, and the parameters used are accuracy, completion, completion ratio, and L1 depth. Table 4 compares the memory requirements and runtime of each model (some runtime numbers were referred from [5], [2])

Replica dataset The Replica dataset is a large-scale benchmark for evaluating visual odometry and SLAM algorithms in photorealistic simulated environments. It provides diverse and challenging scenarios with various lighting conditions, textures, and moving objects. The dataset includes synchronized RGB-D images and ground truth poses, allowing for a comprehensive evaluation of the algorithms' accuracy and robustness in realistic settings.

TUM RGBD dataset The TUM RGBD dataset is a widely used benchmark for evaluating SLAM algorithms in real indoor environments. It consists of several sequences captured with a Microsoft Kinect sensor, including RGB images, depth images, and ground truth poses. The dataset offers a variety of challenges, including dynamic objects, cluttered scenes, and varying lighting conditions.

Why Replica and TUM RGBD datasets are used for this study:

- Representativeness: Both datasets offer realistic and diverse environments, encompassing a wide range of scenarios that neural SLAM methods might encounter in real-world applications.
- Benchmarking: The availability of ground truth data allows for quantitative comparison of the performance

of different neural SLAM methods in terms of accuracy, robustness, and efficiency.

- Complementary characteristics: Combining simulated and real-world data provides a more comprehensive assessment of the algorithms' strengths and weaknesses, highlighting their generalizability to diverse environments.

Experiment Setup

For our experiments, we ran all the models on our laptop with a NVIDIA GeForce RTX 3060 GPU with 6GB GDDR6 and an AMD Ryzen 9 5900HS Processor, 3301 Mhz, 8 Core(s), 16 Logical Processor(s). For the models of iMAP and NICE-SLAM, the number of sampling pixels are selected as 5000. Because of the high GPU requirement for running the NeRF-SLAM, which is close to 11GB, we couldn't run the model at our end. So, we skipped the quantitative comparison for NeRF SLAM.

ATE RMSE (cm) ↓	iMAP*	NICE-SLAM	Co SLAM	vMAP
fr1/desk	5.1	2.8	2.4	2.6
fr2/xyz	2.1	1.9	1.7	1.6
fr3/office	5.9	3.0	2.4	3.0

Table 3: ATE_RMSE Comparison (Localization results) on TUM-RGBD dataset.

5. Conclusion

This comparative study evaluated the performance of five prominent neural SLAM methods: iMap, Nice-SLAM, Co-SLAM, NeRF-SLAM, vMAP. Our findings demonstrate the potential of neural networks to address the limitations of traditional SLAM approaches, achieving improved accuracy, robustness, and adaptability in various environments. However, each method possesses unique strengths and weaknesses, making them suitable for different applications and resource constraints. Co-SLAM is the technique to use for applications with limited resources (both computational and memory constraints). In terms of performance, NeRF-SLAM generalizes much better as compared to other methods, so we can say that it is the best choice if computational resources are not a constraint. However, if accurate reconstruction of the objects in scene and the completion is the primary objective of the task then vMAP is the best method to use.

	iMAP*	Nice-SLAM	Co SLAM	vMAP
Model Param ↓	0.2 M	10.3 M	0.45 M	0.6 M
Runtime ↓	18.4 mins	45.2 mins	2.1 mins	10.3 mins
Mapping Time ↓	430 ms	895 ms	270 ms	245 ms

Table 4: Co-SLAM is most memory efficient and runs faster w.r.t. other models.

Potential Impact of Neural-based SLAM:

- Robotics: Enhance the autonomy and navigation capabilities of robots in complex and dynamic environments.
- Autonomous Vehicles: Enable self-driving cars to navigate safely and efficiently in urban environments.
- Virtual Reality and Augmented Reality: Create realistic and immersive experiences by accurately mapping real-world environments.
- Search and Rescue: Assist robots in locating individuals in disaster areas or hazardous environments.

References

[1] Sucar, E., Liu, S., Ortiz, J., & Davison, A. J. (2021). iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6229-6238).

[2] Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., ... & Pollefeys, M. (2022). Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12786-12796).

[3] Wang, H., Wang, J., & Agapito, L. (2023). Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13293-13302).

[4] Rosinol, A., Leonard, J. J., & Carlone, L. (2022). Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*.

[5] Kong, X., Liu, S., Taher, M., & Davison, A. J. (2023). vmap: Vectorised object mapping for neural field slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 952-961).

[6] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., & Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4460-4470).

[7] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, *65*(1), 99-106.

[8] Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, *41*(4), 1-15.

[9] Park, J. J., Florence, P., Straub, J., Newcombe, R., & Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 165-174).

[10] Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012, October). A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 573-580). IEEE.

Individual Contribution (Group 10)

Shantanu: Ran the models of Nice-SLAM, NeRF-SLAM, presentation, and final report (related work and quantitative).

Vineet: Ran the models of iMAP*, Co-SLAM and, vMAP, presentation and final report (experiments, qualitative and quantitative comparison).