Monograph: Indoor Scene Recognition Using Monocular Scene Graphs

Levi Lingsch[1,*] Rupal Saxena[2,*] Zuria Bauer[1] Mihai Dusmanu[3]
[1]ETH Zurich [2]UZH [3]Microsoft
levi.lingsch@sam.math.ethz.ch rsaxena@student.ethz.ch

Abstract

Scene matching and localization are crucial in applications like autonomous robotics, as these require specific knowledge about the position and spatial domain to plan future actions. Texture and feature-based methods work well outdoors due to the uniqueness of architectural works and landscapes. On the other hand, indoor scenes may feature similar furniture or layouts which cannot be distinguished based on textures. To account for this, graph-based approaches have been explored due to their ability to capture semantic information unique to an indoor scene. However, their reliance on 3D scans limits their performance without depth information. In this paper, we propose a novel method to construct scene graphs directly from RGB images, enabling a comprehensive understanding of semantic content. We conduct an ablation study to analyze the roles of graph connectivity, semantic, and depth predictions. Our approach offers practical advantages by eliminating the need for depth data, demonstrating the potential of graph-based techniques for indoor scene matching. The results contribute insights for developing robust localization systems in indoor environments.

1. Introduction

Visual localization and scene matching are fundamental tasks in computer vision that aim to establish correspondences between images or scenes. These tasks play a crucial role in various applications, including augmented reality [15], robot navigation [28], autonomous driving [5], and 3D reconstruction [18]. Systems such as these require detailed knowledge about their position and the world around them in order to carry out tasks, navigate, and plan for the future. For autonomous robots navigating a building, this means that the system must be able to recognize its current location from some input. In this work, we consider the input of our localization method to be a monocular RGB image.

In visual localization, the goal is to estimate the camera pose or position within a known environment based on image observations. Scene matching, on the other hand, focuses on finding correspondences between images taken from different viewpoints or at different times. Nonetheless, scene matching may be used as a means of localization if the locations of reference scenes are known. Likewise, both tasks may be accomplished by determining the similarity or proximity between images to establish meaningful connections.

Indoor scenes are challenging because different scenes may share similar features and furniture, as opposed to outdoor scenes where building exteriors, landmarks, and landscapes are often unique. Additionally, indoor scenes may change from day-to-day as people interact with the environment. Nonetheless, a situation such as two neighbouring offices with the same furniture, albeit in different layouts, should be distinguishable for an autonomous system to properly localize itself. Graphs find a natural use in this situation. The motivation behind graph-based approaches stems largely from *graph representation learning*, which focuses on representing graph components as low-dimensional feature vectors [3, 31]. Although simple, they efficiently capture meaningful semantic information of both the objects of indoor scenes and the relationships between them, and graph neural networks (GNNs) have been extensively investigated to capture contextual dependencies and exploit the rich relational structure present in the graph [10, 3, 31, 14].

One critical aspect of graph-based methods is the definition of connectivity criteria, parameterized by a threshold. A smaller threshold imposes a stricter criterion for considering two nodes as connected, while a larger threshold allows for more extensive connectivity between nodes. In this paper, we investigate the relationship between the threshold parameter, depth information, and semantic segmentation in the context of graph connectivity for visual localization and scene matching tasks. We treat localization as an image retrieval task, determining the position by matching the input to images of known location. Our goal is to gain insights into how different threshold values and depth sources affect

[*] indicates authors with equal contributions.

the quality of connectivity in the graph when the input is a monocular RGB image. These are quantitatively analyzed by their impact on recall and precision.

Related Work. Simultaneous localization and mapping (SLAM) has played a major role in the field of visual localization. This method relies on range measurement and data extraction [6], and it has been continuously improved over the last two decades. Notably, perceptually rich maps are crucial for accurate localization with a variety of methods aiming to improve this [17, 11, 21, 16, 29, 13]. Graph structures have also been incorporated into localization approaches [12, 8, 23]. Yet, these approaches focus on localization within a currently visible environment, not within a larger-scale structure such as a building.

Scene graph-based image retrieval was first introduced in [9], where the authors proposed a method for detailed image search by representing images as scene graphs, which capture objects and their relationships. In [14], the authors construct graphs directly from RGB images using color-based segmentation. Unlike our approach, this method neglects semantic information and does not maintain rotation and translation invariance through a 3D graph.

Zhang et al. [30] proposed a method, which focuses on 3D scene analysis. Similar to our approach, they leverage 3D point clouds to construct scene graphs that capture semantic information. Armeni et al. [1] present a similar method for scene matching using graphs constructed from a 3D mesh and a panoramic image. However, both of these works only consider inputs with true depth information, excluding monocular RGB inputs.

While these studies have made significant contributions to graph-based methods in indoor scene matching and localization, the performance of semantically rich graphs constructed directly from RGB images remains an open question [22]. Further exploration is needed to understand the effectiveness and limitations of such approaches.

This work makes three key contributions.

- We propose a robust indoor scene matching and localization technique that operates directly on monocular RGB images.
- We provide a comprehensive analysis of the impact of threshold values on graph connectivity in different scenarios for visual localization and scene matching.
- Finally, we investigate the interplay between depth information and semantic segmentation in the context of graph-based methods, shedding light on the performance and limitations of different depth sources.

2. Algorithm and Formulation

The overall pipeline can be divided into several distinct components, as illustrated in Figure 1. Beginning with the RGB image as input, we employ a series of predictive models to estimate depth and perform semantic segmentation. Using this information, we project the scene onto a 3D point cloud representation and construct a graph structure, where each object within the scene corresponds to a graph node. These graphs are then processed by a graph neural network with an appended linear layer that produces a feature vector.

To solve the problem at hand, we assume that a database of pre-generated features extracted from RGB images with known origins is available. Leveraging a proximity matching approach, we retrieve the k nearest neighbors from the database, which serve as potential matches to assist an autonomous system in determining its location. More detailed information regarding the dataset and the fundamental components of the pipeline is provided in the rest of this section.

Hypersim Dataset. The Hypersim dataset, developed by Apple Inc., consists of a collection of photorealistic synthetic images depicting indoor scenes, accompanied by per-pixel ground truth labels [19]. This dataset encompasses more than 77,000 images across 461 scenes. For our purposes, we utilize approximately 12,000 individual images, with 11,300 images used exclusively for training and 700 images reserved for evaluating the generalization capabilities during testing. In our final experiments, we rely solely on the RGB images from this dataset, while leveraging the ground truth depth and semantic labels for the ablation study.

There are a few key definitions that are important for our subsequent discussions. The term **Setting** refers to a specific room configuration. A **Scene** corresponds to a particular viewpoint within a given setting, and scenes can vary in terms of height and physical location. Some settings may consist of only one scene, while others may include multiple scenes. Lastly, a **Frame** denotes an individual image within a specific scene and setting, along with its associated depth and semantic information.

Semantic Segmentation. For the task of semantic segmentation, we employ the *DeepLabV3* model with the ResNet50 backbone [2]. This architecture has demonstrated state-of-the-art performance in semantic segmentation tasks by leveraging powerful feature extraction capabilities. The incorporation of dilated convolutions in the network allows for an expanded receptive field without sacrificing spatial resolution, which proves advantageous for capturing intricate details during the segmentation process.

To further enhance performance, the DeepLabV3 model incorporates the Atrous Spatial Pyramid Pooling (ASPP) module. This module consists of parallel atrous convolutional layers with varying dilation rates, enabling the model to capture multi-scale contextual information. By integrat-

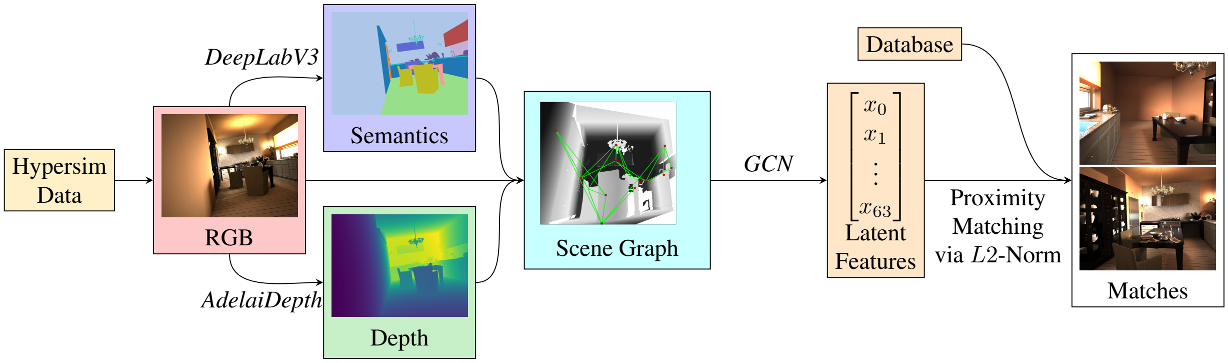


Figure 1: **Monograph Pipeline.** An RGB image is taken from the data set and a 3D semantic scene graph is constructed using predicted values of depth and semantic segmentation. The DeepLabV3 model is fine tuned using a categorical cross-entropy loss, while the AdelaiDepth LeRes model is used out of the box. This is fed to a Graph Convolutional Network (GCN) which has been trained using a triplet-loss to produce a set of latent features. Using a proximity matching method, matching scenes are procured from a database.

ing information from different scales, the model can effectively handle objects of different sizes within the scene.

We fine-tuned this model using a categorical cross-entropy loss on the training split of the Hypersim dataset to predict the NYU40 [20] semantic labels.

Depth Estimation. To generate depth predictions, we utilize the *AdelaiDepth* open-source toolbox for monocular depth estimation [26, 27, 25, 24]. This toolbox offers methods employing either ResNet50 or ResNet101 backbones to estimate depth from single monocular RGB images. Notably, the model has been trained on a large dataset consisting of over 6 million images, enabling it to achieve state-of-the-art performance in generating 3D point clouds of indoor scenes directly from RGB inputs and monocular depth estimation for the ScanNetV2 data set [27, 4].

The *AdelaiDepth* toolbox proves particularly valuable in our pipeline for several reasons. The extensive training data utilized by the toolbox facilitates its out-of-the-box application, eliminating the need for additional fine-tuning. Furthermore, the fact that this toolbox excels in producing 3D point clouds of indoor scenes aligns perfectly with the objectives of our pipeline.

Scene Graph Construction. The scene graph construction process begins by projecting the depth information onto a 3D point cloud. The semantic labels obtained from the previous step serve as a mask for the 3D projection. One key advantage of the 3D projection is its ability to provide translation and rotation invariance within the resulting scene graph. Leveraging the positional information of all points belonging to a specific semantic object, we compute the centroid, as well as the average RGB values across the object.

The graph is then constructed by creating a node at the positional median for each semantic object. Each node is associated with four attributes: the *NYU40* semantic label, and the median RGB values (with each color stored separately). Edges are established within the graph by connecting nodes that are in proximity to each other, where proximity is defined by a threshold distance in meters. This is visualized by the green lines in the scene graph of Figure 1. Additionally, the lengths of the edges are stored as edge attributes. In the ablation study, various edge distance thresholds are explored to examine their impact on the scene graph construction.

Graph Neural Network. For the machine learning components of our project, we utilized the *PyTorch* framework. Specifically, we made use of *PyTorch Geometric*, a library built on top of *PyTorch* that specializes in handling structured data [7]. The graphs constructed in the previous step were organized as `torch_geometric.data` objects, which facilitated the organization of nodes, edges, and their respective features, along with additional scene and setting information for evaluation purposes.

PyTorch Geometric provides a variety of pre-built neural network models that can be easily customized and trained for different applications. For this project, we selected the *Graph Convolutional Network* (GCN) introduced in [10]. This model utilizes a first-order approximation of spectral graph convolutions, allowing it to capture both local graph structures and node features. Furthermore, we appended a linear layer to the GCN output, producing a set of latent features.

In the ablation studies, the GCN structure was kept consistent. The graph was passed through five convolutional layers, each with 128 hidden channels. The resulting output was then fed into a linear layer to generate a set of 64 latent features for each node in the graph. Finally, a maximum-pooling operation was applied over all nodes to obtain a fixed-size latent feature vector of size 64, regardless of the graph size. The architecture of this model is visualized in Figure 2.

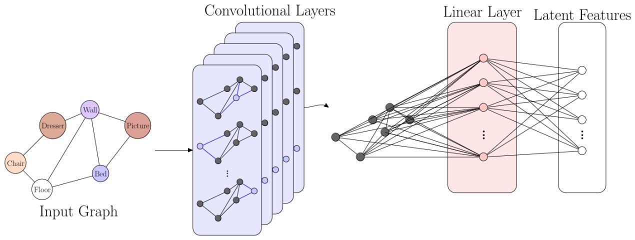


Figure 2: **Graph Convolutional Network.** A graph with semantic information may be passed to the GCN, through 5 convolutional layers of 128 hidden channels. A linear layer is then used to transform the graph to a vector of 64 latent features.

The training of the GCN is done using a triplet loss with an $L2$-Norm and a margin of 1. Mathematically, the loss function is formulated as:

$$\mathcal{L}(A,P,N) = \max\Big(||f(A)-f(P)||_2 - ||f(A)-f(N)||_2 + 1, 0\Big), \tag{1}$$

Here, A, P, and N represent an *anchor* graph, *positive* match, and *negative* match, respectively. The function $f()$ denotes the non-linear mapping performed by the GCN, which maps the graph space to the latent-feature space.

To create the (A, P, N) triplets, we generate them separately for each setting. For each graph within a setting, we randomly select it as an anchor. Then, we randomly choose a positive match from all other graphs within the same setting. Lastly, the negative match is randomly selected from all graphs in different settings.

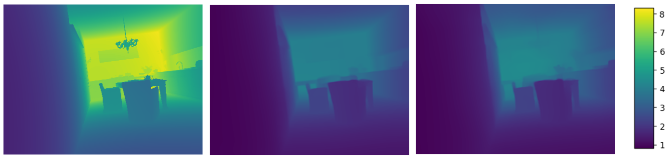
Proximity Matching. After obtaining the latent features from the GCN, we can interpret them as positions in a latent feature space. Since the GCN is trained using the $L2$-Norm, similar graphs should have smaller Euclidean distances between their corresponding latent-feature vectors. Therefore, to find the k best matches for a given graph, we can simply select the k nearest neighbors based on the Euclidean distance in the latent feature space.

3. Results

We first present results from intermediate components of the pipeline, namely depth and semantic predictions, followed by the results of the full pipeline and ablation.

Intermediate Results. The *AdelaiDepth* model was used without any fine-tuning. The model we employ predicts a relative depth, which is later corrected by another network which has been trained to predict the metric depth based on the focal length of the camera. As shown in Figure 3, the model tends to underpredict depth by a significant margin without the correction network. Despite this discrepancy, the qualitative accuracy of the predictions is still preserved. When the predicted depths are normalized to a range between 0 and 1, the differences between the ground truth and the predictions become minimal.

Comparing the results between the ResNet50 and ResNet101 backbones, we observe little difference. The model using the ResNet101 backbone slightly outperforms the ResNet50 backbone in resolving the edges of objects, but the overall predictions are quite similar. This comparison provides an interesting ablation study, as the edges of objects may play a crucial role in determining the medians of objects and subsequently affect the graph construction.



(a) Ground Truth (b) ResNet50 (c) ResNet101

Figure 3: **AdelaiDepth Predictions**. Compared to the ground truth, predictions are approximately half of what is expected, regardless of the chosen backbone. The ResNet101 backbone offers slight improvement resolving edges and distant objects.

The semantic segmentation proved to be a more challenging task. The quality of per-pixel labels in the *Hypersim* dataset could not be fully matched, particularly in scenarios with intricate details. For instance, as shown in Figure 4, the predictions struggle to accurately resolve the window shutters.

Despite this limitation, the semantic segmentation model based on *DeepLabV3* with the ResNet50 backbone still demonstrates reasonable performance in capturing the overall semantic structure of the scene. It effectively segments various objects and regions, even though finer details may be lost or incorrectly classified.

Full Pipeline Results. The pipeline was evaluated using the following experimental setup. Latent-feature vectors from 16 settings, each containing 41 frames, were extracted from the test split. For each setting, 35 images were randomly selected as queries, while the remaining 6 images

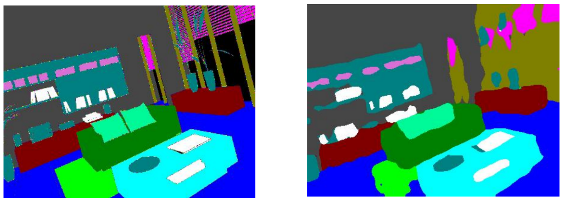

(a) Ground Truth (b) Semantic Predictions

Figure 4: **DeepLabV3 Predictions.** Using the ResNet50 backbone, this model is able to obtain a complete semantic segmentation with reasonable accuracy, but some errors are present around objects which must be finely resolved, e.g. the window shutters (top right corner).

served as positive matches. The database was constructed using the 6 positive images and all frames from the other 15 settings, resulting in a database of 600 images. The chance of randomly drawing a positive match from this database is approximately 1%.

To assess the impact of different configurations, the pipeline was evaluated on four different *Scenarios*, each with a specific data set:

1. Ground truth depth and semantics
2. Ground truth depth and *DeepLabV3* semantics
3. *AdelaiDepth* (ResNet50) and *DeepLabV3* semantics
4. *AdelaiDepth* (ResNet101) and *DeepLabV3* semantics

In addition, we analyze the role of the threshold distance which decides whether two nodes are connected. The results are summarized in Table 1 for recall and Table 2 for precision.

For recall, we computed the values at $k = 1$, 5, and 10 to capture a wide range. As for precision, we considered $k =$ 2, 4, and 6. Since recall and precision at $k = 1$ are identical, we started from $k = 2$, and since the maximum number of positives is 6, we stopped at $k = 6$. We also included $k = 4$ as an intermediate value.

Among the different scenarios, the best overall recall and precision results were obtained when using ground truth depth and semantic segmentation (*Scenario 1*) with a threshold of 1 meter. In *Scenario 2*, a threshold of 1 meter also resulted in the best precision values, but there was little correlation between recall and threshold. For *Scenarios 3* and *4*, a threshold of 5 meters offered the best performance in most cases.

4. Discussion

Quantitative and Qualitative Analysis. The threshold parameter is crucial in determining the connectivity of the graph and affects the results in the tables. A smaller threshold imposes stricter connectivity criteria, while a larger threshold allows for more extensive connectivity.

Table 1: **Recall at *k*.** We compute the recall as a percentage of instances where a true positive is obtained. The best Scenario results are bold, while best overall results are in blue.

Data	Distance	k		
	Threshold (m)	1	5	10
Scenario 1	1	**94.3**	**94.7**	**96.2**
	2	90.9	92.8	96.0
	5	80.2	88.0	92.4
	10	91.4	91.4	91.4
	∞	93.5	93.8	96.0
Scenario 2	1	**83.3**	87.8	89.2
	2	81.4	87.8	88.5
	5	**83.3**	**89.2**	89.3
	10	83.1	88.3	**90.0**
	∞	78.9	85.6	87.1
Scenario 3	1	71.7	81.3	86.4
	2	68.8	74.4	82.7
	5	**82.5**	**87.7**	**89.2**
	10	79.8	84.2	84.6
	∞	78.3	84.6	85.6
Scenario 4	1	73.1	84.2	86.7
	2	66.3	75.6	79.2
	5	**79.4**	**87.5**	86.3
	10	75.0	86.3	87.7
	∞	79.2	85.8	**91.0**

In Scenario 1, high recall and precision values are consistently obtained across different values of k, indicating that any chosen threshold effectively captures the desired connectivity.

Comparing Scenario 2 to Scenarios 3 and 4, we observe lower recall and precision values. Scenario 2 uses true depth, while Scenarios 3 and 4 use predicted depth that is approximately half of the true value. Thus, we would expect similar effects in Scenarios 3 and 4 with lower thresholds. While this holds true for precision, it is not the case for recall. The lower threshold in Scenario 2 improves precision by allowing finer-grained connectivity, but this improvement does not necessarily translate to recall.

Incorporating predicted depth enhances some results when using predicted semantic segmentation. Surprisingly, the AdelaiDepth model with the ResNet50 backbone performs better than the ResNet101 backbone in most scenarios, despite less accurate depth predictions. The ResNet50 backbone's smoother transition across object edges may avoid overemphasis on edges when calculating the object centroid, leading to better predictions of the true centroid.

Qualitative results shown in Figure 5 reveal that well-lit scenes with rich semantic information are easily matched by

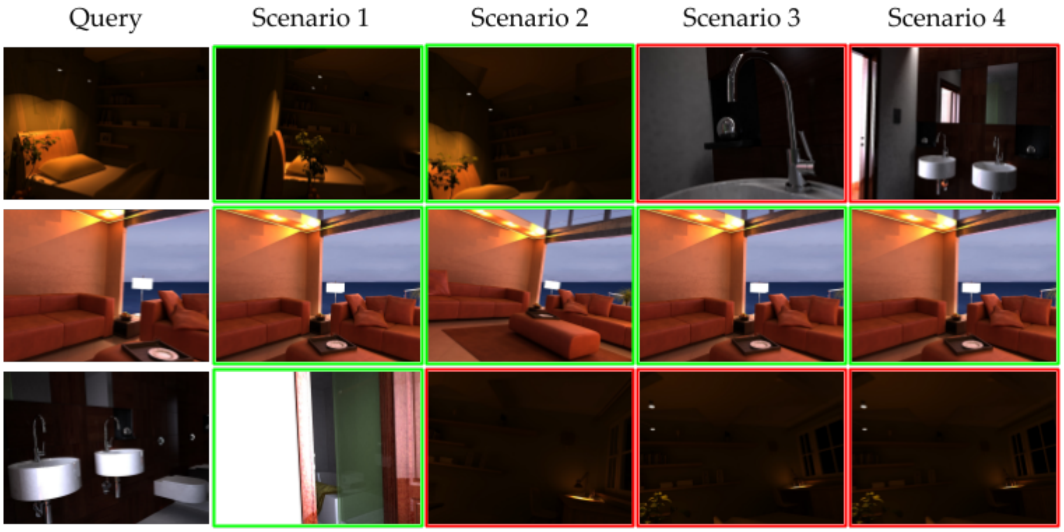


Figure 5: **Matching Results.** The first returned result for each scenario is displayed. Two difficult scenes (top and bottom row) produce features which are often confused for one another, while a clear setting allows correct results to easily be found. A green border represents a true positive, and a red border a false positive.

Table 2: **Precision at *k*.** We compute the precision as the aggregate percentage of true positives retrieved. The best Scenario results are bold, while best overall results are in blue.

Data	**Distance**	**k**		
	Threshold (m)	2	4	6
Scenario 1	1	85.7	**85.8**	**77.2**
	2	81.5	79.6	72.7
	5	83.7	69.4	62.9
	10	83.8	76.4	66.9
	∞	**90.7**	77.2	71.7
Scenario 2	1	**79.4**	**72.2**	**63.5**
	2	78.3	65.1	57.5
	5	75.9	69.5	61.3
	10	75.8	69.9	61.9
	∞	73.4	65.1	53.0
Scenario 3	1	61.9	58.5	46.0
	2	62.2	50.1	43.1
	5	**76.4**	**68.9**	59.2
	10	74.5	64.7	53.9
	∞	71.2	66.8	**60.1**
Scenario 4	1	67.4	59.9	50.6
	2	62.5	47.2	39.5
	5	**77.0**	63.7	**59.9**
	10	68.1	64.9	54.4
	∞	75.2	**65.2**	59.2

all models. In contrast, low-light scenes with little semantic information pose challenges, and confusion is more likely, especially when semantic and depth predictions are less reliable. This emphasizes the importance of capturing a minimum level of semantic information for constructing meaningful graphs. In dark settings, a sufficiently rich graph cannot be constructed.

Limitations and Future Work Several limitations should be considered in this project. For the system to operate autonomously, it needs to be fast. While the GCN model is small and lightweight, the models relying on large CNNs used to generate graphs are relatively slow and resource-intensive. Additionally, the projection to a 3D point cloud is computationally expensive and time-consuming. Exploring alternatives, such as direct prediction of bounding boxes around objects for median estimation, could help mitigate these limitations. Generalizing the model's capabilities to real-world scenarios beyond the Hypersim environment, especially those which exhibit the dynamic nature of object-person interactions, would be a valuable avenue for future investigation.

Work Distribution Levi contributed to monocular depth estimation, construction and training of the GCN, latent feature matching, threshold comparision, and the report. Rupal contributed on training of semantic segmentation model on hypersim data, 3D scene graph generation, project infrastructure, pipeline integration, and the midterm presentation. Levi and Rupal both contributed on dataloaders and final poster presentation. We agree that we contributed equally to the project.

References

[1] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera, 2019.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[3] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE transactions on knowledge and data engineering*, 31(5):833–852, 2018.

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[5] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Thanh-Toan Do, and Ian Reid. Scalable place recognition under appearance change for autonomous driving, 2019.

[6] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics and Automation Magazine*, 13(2):99–110, 2006.

[7] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with pytorch geometric. *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 2019–2026, 2019.

[8] Xiyue Guo, Junjie Hu, Junfeng Chen, Fuqin Deng, and Tin Lun Lam. Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment. *IEEE Robotics and Automation Letters*, 6(4):8349–8356, oct 2021.

[9] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number 2, pages 3668–3678, 2015.

[10] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[11] Dongjiang Li, Xuesong Shi, Qiwei Long, Shenghui Liu, Wei Yang, Fangshi Wang, Qi Wei, and Fei Qiao. Dxslam: A robust and efficient visual slam system with deep features, 2020.

[12] Shiqi Lin, Jikai Wang, Meng Xu, Hao Zhao, and Zonghai Chen. Topology aware object-level semantic mapping towards more robust loop closure. *IEEE Robotics and Automation Letters*, 6(4):7041–7048, 2021.

[13] David G. Lowe. distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[14] Mario Manzo. Graph-based image matching for indoor localization. *Machine Learning and Knowledge Extraction*, 1(3):785–804, 2019.

[15] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. 09 2014.

[16] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[17] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011.

[18] Onur Ozyesil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion, 2017.

[19] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

[20] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[21] Lukas von Stumberg, Patrick Wenzel, Nan Yang, and Daniel Cremers. Lm-reloc: Levenberg-marquardt based direct visual relocalization, 2020.

[22] Yuanyan Xie, Yu Guo, Zhenqiang Mi, Xiaokun Wang, Yang Yang, and Mohammad S. Obaidat. Indoor visual re-localization for long-term autonomous robots based on object-level features and semantic relationships. *IEEE Robotics and Automation Letters*, 8:840–847, 2023.

[23] Kuan Xu, Chen Wang, Chao Chen, Wei Wu, and Sebastian Scherer. AirCode: A robust object encoding method. *IEEE Robotics and Automation Letters*, 7(2):1816–1823, apr 2022.

[24] Wei Yin, Yifan Liu, and Chunhua Shen. Virtual normal: Enforcing geometric constraints for accurate and robust depth prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[25] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[26] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Simon Chen, Yifan Liu, and Chunhua Shen. Towards accurate reconstruction of 3d scene shape from a single monocular image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[27] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR)*, 2021.

[28] Safdar Zaman, Wolfgang Slany, and Gerald Steinbauer. Ros-based mapping, localization and autonomous navigation using a pioneer 3-dx robot and their relevant issues. In *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*, pages 1–5, 2011.

[29] Chao Zhang, Ignas Budvytis, Stephan Liwicki, and Roberto Cipolla. Lifted semantic graph embedding for omnidirectional place recognition. In *2021 International Conference on 3D Vision (3DV)*, pages 1401–1410, 2021.

[30] Chaoyi Zhang, Jianhui Yu, Yang Song, and Weidong Cai. Exploiting edge-oriented reasoning for 3d point-based scene

graph analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9705–9715, June 2021.

[31] Detian Zhang, Chi-Yin Chow, An Liu, Xiangliang Zhang, Qingzhu Ding, and Qing Li. Efficient evaluation of shortest travel-time path queries through spatial mashups. 22:3–28, 2018.