Natural Language-Guided Semantic Navigation using Scene Graph

Dohyun Kim*, Jinwoo Kim*, Minwoo Cho, and Daehyung Park

School of Computing, Korea Advanced Institute of Science and Technology (KAIST),
Daejeon, Republic of Korea,
{dohyun141, kjwoo31, cmw9903, daehyung}@kaist.ac.kr,
http://rirolab.kaist.ac.kr

Abstract. We expect legged robots to perform complex navigation tasks by having higher mobility and sensing capability. However, the mobility does not easily lead to the ability of performing complex user tasks due to restricted interfaces such as a joystick. To perform tasks that users want, the robots require an expressive and accessible interface that can deliver human intention with lower mental demands, even in complex environments. In this work, we propose a novel natural language-guided robotic navigation framework that can effectively ground natural-language commands in large space. Our framework consists of three modules: a scene-graph generator, a grounding network, and a semantic navigation system. The scene-graph generator incrementally stores the semantic information of object instances, properties, and relationships. Then, the proposed scene graph-based grounding network (SGGNet) predicts the desired goal robustly by associating instances in a scene graph with a user command. Finally, the navigation system enables the robot to reach the goal location. Our evaluation result shows SGGNet achieves a grounding accuracy of 77.8% given $3,000$ scene graphs and $9,000$ natural language commands. The model also achieves a grounding accuracy of 52.4% given unforeseen objects. We demonstrate the robust performance of the proposed framework in three real-world scenarios with various speech commands.

Keywords: Natural language grounding, Scene graph, Semantic navigation

1 Introduction

Consider a problem of autonomous navigation for legged robots that can robustly navigate through challenging environments. Conventional control interfaces, such as a joystick or a touch panel, promise reliable operations but are hard to transfer our complex intentions or missions. Thus, we need an interface that requires low mental demand while being flexible to ground complex mission objectives.

Natural language grounding (NLG) is a candidate that has gained significant attention in recent years. NLG, particularly visual grounding (VG), has shown

[*] These authors contributed equally to this work

capabilities of locating objects [4], object identification [1], and relationships [11, 20] in an image or point clouds, given a natural language query. To expand grounding targets over pixels, researchers often use more structured sources such as scene graphs [16]. In Robotics, the use of NLG itself is not a new paradigm [9]. However, the observable/operation area is too large to be presented in an image. Therefore, researchers often perform language grounding high-level representations: a dictionary format of world model [2], and semantic map [15].

For example, conventional NLG approaches for navigation introduce methodologies for encoding the semantic information of a large navigation environment by using visual bag-of-words [14], topological maps from landmarks [3], or scene graphs [18]. The representation of scene graphs is particularly helpful for maintaining semantic information and geometric information about the world jointly considering topological relationships between entities. Further, the representation is also effective in incrementally adding/removing information based on new observations.

In this paper, we introduce a novel natural language-guided navigation framework that grounds navigational instructions given a scene graph. We particularly focus on the design of a unified framework, which we call a scene-graph grounding network (SGGNet), that encodes a linguistic input as well as a scene graph to return a grounded target with necessary action. We also show a completed semantic navigation framework combining the SGGNet and a simultaneous localization and mapping (SLAM) framework. Then, we show a legged robot can perform various spatial grounding and navigation tasks in the real world.

The proposed framework consists of three phases. First, the robot generates a scene graph under the observation and recognition of the object/location attributes as well as their relationships with others. Then, given a navigational phrase, the robot locates a referred goal from the scene graph. Finally, the robot navigates to the goal by planning a collision-free path. We verify the accuracy of the natural grounding model with the scene graph data from CLEVR [10]. In addition, we demonstrate three indoor language-guided navigation tasks based on the attribute, category, and relationship of goals with a real legged robot.

2 Natural Language-guided Navigation Framework

We describe the proposed natural language-guided navigation framework in detail. Fig. 1 illustrates the overall framework, where a robot follows a navigation task through three steps. The scene-graph generator first constructs a scene graph storing identified objects or locations. Then, given navigational instruction, the SGGNet module performs natural-language grounding to find a desired goal in the scene graph. Finally, the semantic navigation module operates path planning to reach the target location while running SLAM.

2.1 Scene-Graph Generator

We introduce the scene-graph generation module, which constructs a scene graph capturing environmental semantics, such as objects, attributes, or relations be-

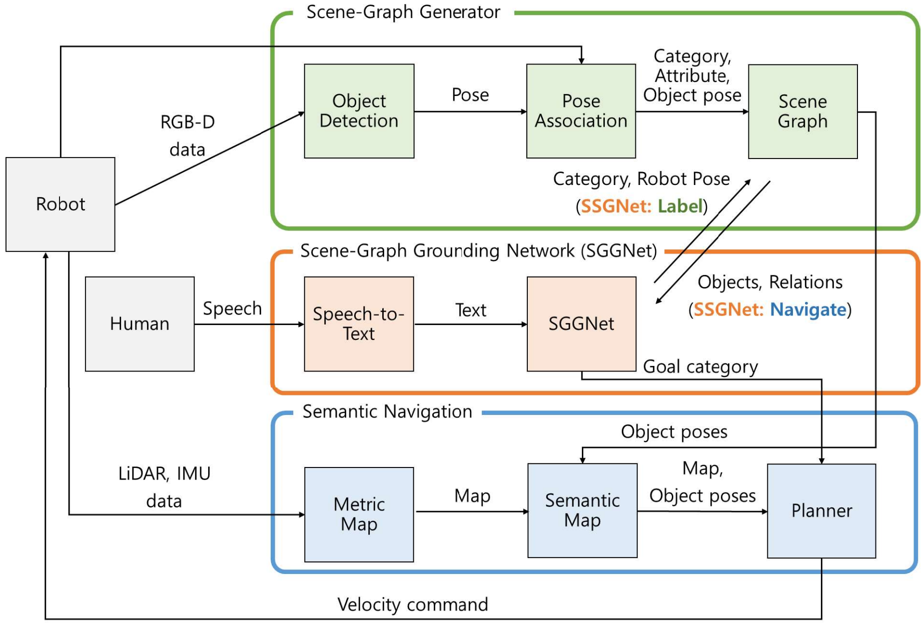
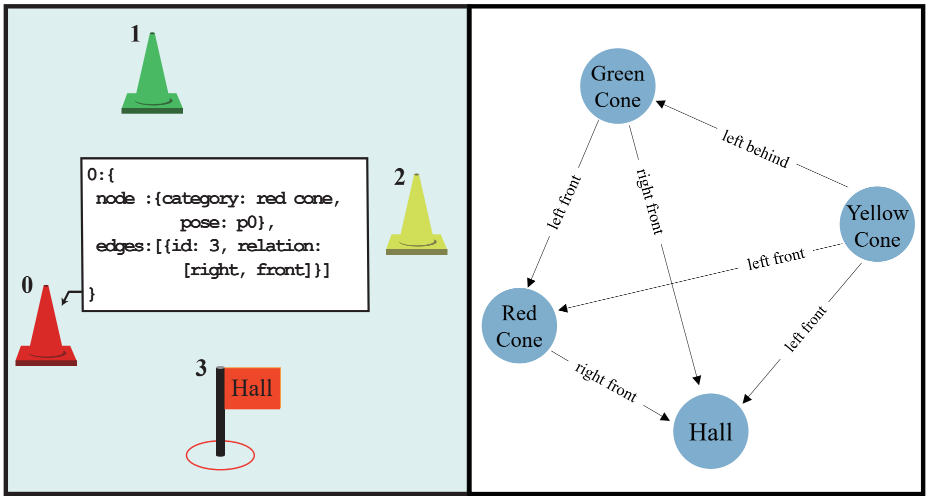


Fig. 1. Block diagram of natural language-guided navigation framework. Green, orange, and blue color boxes represent modules for the scene-graph generator, scene-graph grounding network, and semantic navigation system, respectively.



```
0:{
 node :{category: red cone,
          pose: p0},
 edges:[{id: 3, relation:
           [right, front]}]
}
```

Fig. 2. Illustration of a scene graph with three-cone environment. We represent identified objects (*Left*) as nodes in the scene graph (*Right*). In the scene graph, each node holds both node and edge features in a dictionary format. The node feature contains object category and pose information. The edge feature includes the target node information and a list of relationships between the source and the target node.

tween paired objects [22]. The module generates a dictionary format of a scene graph, $G = (V, E)$, where V and E are a set of nodes and directional edges, respectively. Fig. 2 shows an example.

We define a dictionary format node as $v = (c, p) \in V$, where c indicates an object category and p is a pose. If there exists a relationship $r \in R$ between start node v_i and destination $v_j \in V$, the module connects them to an edge $e_{ij} \in E$, where R is a set of predicates:

- $\texttt{LEFT}(v_i, v_j)$ if $x_j < x_i$,
- $\texttt{RIGHT}(v_i, v_j)$ if $x_j \geq x_i$,
- $\texttt{FRONT}(v_i, v_j)$ if $y_j < y_i$,
- $\texttt{BEHIND}(v_i, v_j)$ if $y_j \geq y_i$,

where the x and y represent object coordinates in the global frame. Likewise, we also represent an edge as a dictionary that consists of the destination node *ID* number and the relationships between the current node and the destination node. Thus, the edge is directional and it belongs to the start node of the edge.

The proposed module automatically generates a scene graph through object detection and pose association steps. From a given image, an object detector, YOLO [17], recognizes objects with their bounding boxes. Then, the module estimates its poses by aligning the detection results with depth data. We consider the resulting pose is precise if there are clustered point clouds within a certain threshold distance. For association, we transform the estimated pose into the global map frame using the odometry of the robot. Finally, the module adds the name, attribute, and object pose to the scene graph.

Further, the module provides a human-in-the-loop update method that adds a new node labeled through natural language grounding. When a human wants to assign a name to the space where a robot is, our method registers the location and position of the robot as a new node in the graph. This enables an operator to manipulate the map and the robot robustly ground target spaces.

2.2 Scene-Graph Grounding Network (SGGNet)

We introduce SGGNet, which predicts the desired object from a scene graph and an action, such as `NAVIGATE` or `LABEL`, given a natural language command. Fig. 3 shows the entire architecture.

Given a natural language command, the SGGNet first encodes a scene graph into a graph feature vector using a text encoder, BERT [5], to embed object categories and relations. A graph neural network (GNN) then updates the vectors using a message passing scheme. To hold permutation invariance, we then pool the updated feature vectors to obtain the final graph feature vector.

Our model uses a pre-trained language model, BERT [5], to predict a target object from the graph and action specified in the command. The speech-to-text model, provided by Google Speech Recognition, first converts the voice command to text. We then employ Prefix-tuning method [12], in which a trainable neural network generates fake prefix vectors to combine with word embedding vectors

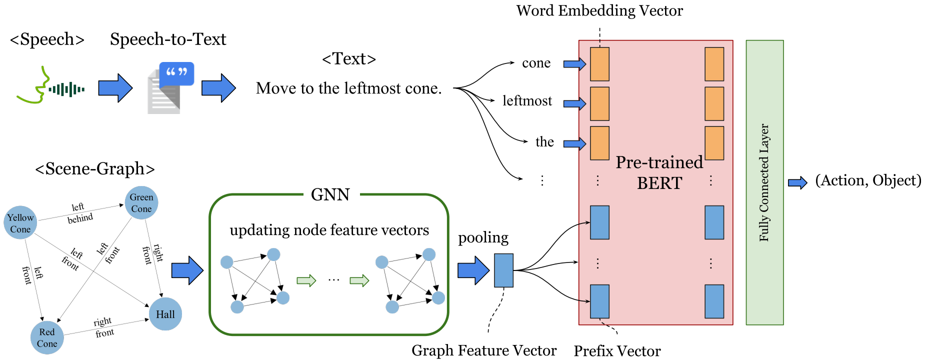


Fig. 3. Illustration of the scene-graph grounding-network architecture. Our model first uses a speech-to-text model to convert the voice command spoken by a person to text. The model also encodes a scene graph to a feature vector via a graph neural network. Then, a pre-trained large language model, BERT [5], encodes the input sentence and the graph-feature vector by concatenating the feature vector as a prefix of the input sentence. The model concatenates the prefix and the text to be one sentence and encodes them via the pre-trained language model. Finally, the last fully-connected layer returns a grounded object from the graph and an available action specified in the speech command.

from the original sentence. They then feed the concatenated vectors to the frozen pre-trained model to perform natural language processing tasks. Our model uses the final graph feature vector produced by the GNN as a prefix vector. Then, our model concatenates the graph and word vectors and then feeds the vector to the pre-trained language model. Finally, the last fully-connected layer returns a target object and action pair.

2.3 Semantic Navigation

Lastly, we introduce a semantic navigation framework that consists of two steps: 1) semantic map construction and 2) planning and execution with grounded commands.

To build a 2D semantic map, we first adopt a state-of-the-art SLAM algorithm, called Fastlio2 [21], renowned for its fast computation performance. The algorithm fuses the measurements from LiDAR and IMU sensors to estimate the robot pose relative to the *odometry* frame. The fast computation of Fastlio2 enables it to quickly process dense point clouds, resulting in a robust performance. Through Fastlio2, we obtain the points registered in the global map frame. Then, we filter the registered points by the z-axis coordinates to remove the ground and ceiling. We then construct a 3D metric map by accumulating the filtered points using Octomap library [8] for efficient map management. Finally, we project the 3D metric map to get a 2D occupancy-grid map for navigation and add scene graph objects to the map.

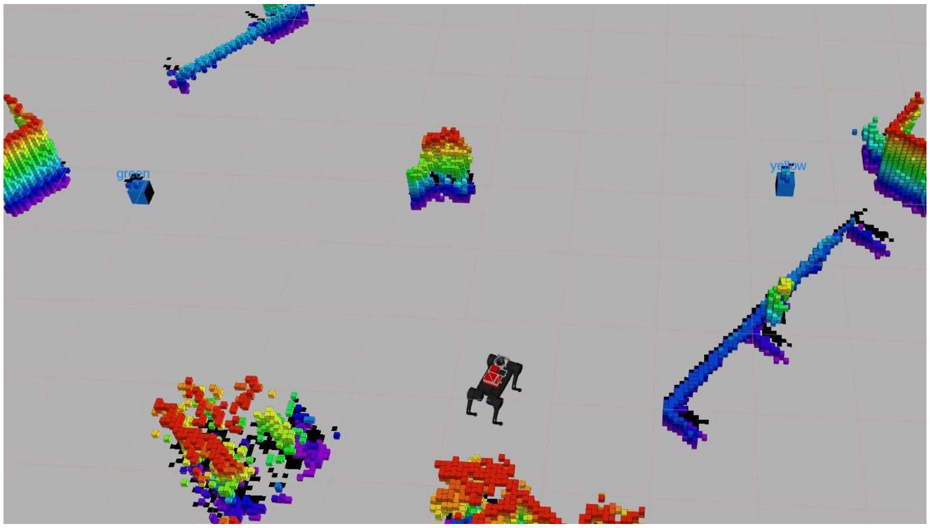


Fig. 4. Visualization of the semantic map with identified objects (i.e., cones). The rainbow-colored points represent the 3D metric map. The gray and black area represents the 2D occupancy map.

Our framework generates a collision-free path on a cost map using Dijkstra's algorithm [6] and DWA (Dynamic-Window Approach) local planner [7]. Based on the metric map, we build a cost map showing obstacle regions by inflating the occupied points as the robot size. After generating the cost map, using Dijkstra's algorithm, the robot plans the shortest path to reach the goal without collision. The robot follows the path using the DWA local planner that determines the robot velocity command.

3 Evaluation

We first evaluate the accuracy and robustness of our model, SGGNet. We also demonstrate our natural language-guided navigation framework with a real legged robot.

3.1 Experimental Setup

We trained and evaluated the SGGNet with the scene-graph data from CLEVR dataset [10], where we only used the relationship in edges but not the node information since the dataset has limited object categories, such as cube, cylinder, and sphere. We collected 20 different words as object categories from WordNet [13] and paired them with action words, such as `NAVIGATE` and `LABEL`. For each scene graph in the dataset, we generated three typical natural-language commands for robotic navigation or labeling. To train it, we used $12,000$ samples of the dataset which contains a pair of $4,000$ scene graphs and four commands per scene graph and we used the same form of the $3,000$ dataset for evaluation.

Also, to evaluate the robustness of the unseen objects, we generated an additional dataset with no common object category between the train and evaluation datasets. As a baseline model, we used a simple multi-layer perceptron (MLP) based network. The MLP-based network first encodes a graph and a text via embedding layers to produce feature vectors. Then, the network concatenates each feature vector and applies MLP to classify the object.

We also demonstrated natural language-guided navigation tasks using a real legged robot. In a hall environment with three colored cones, we gave three types of commands referring to the attribute, category, and relationship of the objects.

Fig. 5 shows our sensor system of the robot. We obtained image and point clouds from Ouster OS-1 32 channel LiDAR and Realsense D455 RGB-D camera. To perform navigation in a large space, we use sensors with a relatively long range. From the sensor data, we process the natural language-guided navigation framework with an NVIDIA Jetson AGX Orin machine, a single-board computer with high GPU performance.

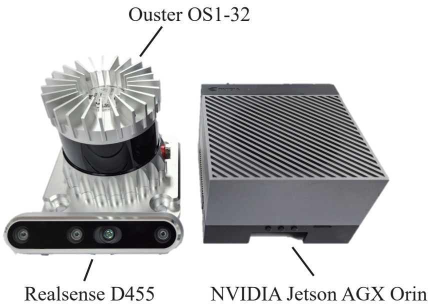
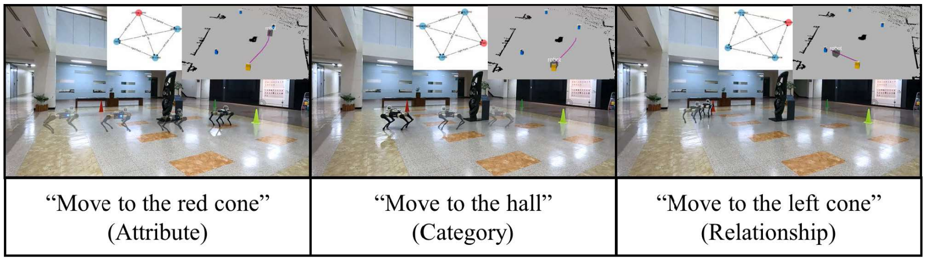


Fig. 5. Sensor-computation system. The Ouster OS1-32 LiDAR sensor produces point cloud for metric map generation. The Intel Realsense D455 camera takes RGB-D images for object detection. Then, the NVIDIA Jetson AGX Orin machine fuses the sensor data for generating a scene map for semantic navigation.

3.2 Experimental Results

Given known object categories, we evaluated our model, SGGNet, on a $3,000$ evaluation dataset and achieved 77.78% of accuracy. This means that our model is able to understand the language instructions in various environments. Given unknown object categories, our model achieved 52.4% of accuracy while the MLP-based model achieved 0.93%. The result shows that our model is robust to unseen objects successfully deploying the pre-trained language model from a large dataset.

In addition, we demonstrated our natural language-guided navigation framework with a real legged robot. Fig. 6 shows the navigation scenarios, where the robot finds a goal in the indoor environment. Given the natural language command, SGGNet successfully found the expected destination from the scene graph. Overall, the demonstrations show our robot understands human commands and moves as intended in real-world navigation scenarios.



"Move to the red cone" (Attribute)

"Move to the hall" (Category)

"Move to the left cone" (Relationship)

Fig. 6. Captures of natural language-guided robotic navigation experiments.

4 Conclusion

We introduced a natural language-based navigation framework that utilizes a scene graph in NLG. This framework has an advantage in grounding navigation instructions in a large environment. Given natural-language commands, the evaluation result shows that the SGGNet can robustly find the referred goal. The framework can successfully conduct natural language-based navigation with attribute, category, and relationship of goals with a real legged robot.

In future work, we will improve the accuracy of SGGNet by replacing the BERT with a better pre-trained model, such as the DistilBERT [19]. Since the DistilBERT has 40% fewer parameters than the BERT, it can also help to increase the inference speed, which has an impact on real-time navigation system. We also have a plan to improve the framework by combining a neural SLAM method. The current framework can only perform navigation commands with the shortest path. However, by combining the neural SLAM and scene graph, we can plan a complex path, such as a constrained path that passes between two obstacles.

Acknowledgement. This work was supported by the National Research Foundation of Korea(NRF) grants funded by the Korea government(MSIT) (No. 2021R1C1C1004368 and 2021R1A4A3032834).

References

1. Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 422–440. Springer (2020)
2. Arkin, J., Park, D., Roy, S., Walter, M.R., Roy, N., Howard, T.M., Paul, R.: Multimodal estimation and communication of latent semantic knowledge for robust execution of robot instructions. International Journal of Robotics Research **39**(10-11), 1279–1304 (2020)
3. Blochliger, F., Fehr, M., Dymczyk, M., Schneider, T., Siegwart, R.: Topomap: Topological mapping and navigation based on visual slam maps. In: Proceedings

of the International Conference on Robotics and Automation (ICRA). pp. 3818–3825. IEEE (2018)

4. Chen, D.Z., Chang, A.X., Nießner, M.: Scanrefer: 3d object localization in rgb-d scans using natural language. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 202–221. Springer (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) (2019)
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**, 269–271 (1959)
7. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. Robotics & Automation Magazine **4**(1), 23–33 (1997)
8. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: an efficient probabilistic 3d mapping framework based on octrees. Autonomous Robots **34**, 189–206 (2013)
9. Howard, T., Stump, E., Fink, J., Arkin, J., Paul, R., Park, D., Roy, S., Barber, D., Bendell, R., Schmeckpeper, K., et al.: An intelligence architecture for grounded language communication with field robots. Field Robotics (2022)
10. Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2901–2910. IEEE (2017)
11. Krishnamurthy, J., Kollar, T.: Jointly learning to parse and perceive: Connecting natural language to the physical world. Transactions of the Association for Computational Linguistics **1**, 193–206 (2013)
12. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4582–4597 (2021)
13. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
14. Nicosevici, T., Garcia, R.: Automatic visual bag-of-words for online robot navigation and mapping. Transactions on Robotics **28**(4), 886–898 (2012)
15. Patki, S., Fahnestock, E., Howard, T.M., Walter, M.R.: Language-guided semantic mapping and mobile manipulation in partially observable environments. In: Conference on Robot Learning (CoRL). pp. 1201–1210. PMLR (2020)
16. Qiao, Y., Deng, C., Wu, Q.: Referring expression comprehension: A survey of methods and datasets. In: Transactions on Multimedia. vol. 23, pp. 4426–4440. IEEE (2020)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 779–788. IEEE (2016)
18. Rosinol, A., Violette, A., Abate, M., Hughes, N., Chang, Y., Shi, J., Gupta, A., Carlone, L.: Kimera: From slam to spatial perception with 3d dynamic scene graphs. International Journal of Robotics Research **40**(12-14), 1510–1546 (2021)
19. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In: The 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS) (2019)

20. Wang, P., Wu, Q., Cao, J., Shen, C., Gao, L., Hengel, A.v.d.: Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1960–1968. IEEE (2019)
21. Xu, W., Cai, Y., He, D., Lin, J., Zhang, F.: Fast-lio2: Fast direct lidar-inertial odometry. Transactions on Robotics **38**, 2053–2073 (2022)
22. Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., Shen, P., Feng, M., Zhao, X., Miao, Q., Shah, S.A.A., et al.: Scene graph generation: A comprehensive survey. arXiv preprint arXiv:2201.00443 (2022)