Online Metric-Semantic Mapping for Autonomous Robot Navigation

Jianhao Jiao[†], Ruoyu Geng[†], Yuanhang Li[†], Bowen Yang[†] and Ming Liu[†]
[†]Department of ECE, Hong Kong University of Science and Technology, China.
Email: jjiao@connect.ust.hk

***Abstract*—Metric-semantic mapping is the key capability for robots to improve the performance robustness as well as execute human instructions. This paper present an online metric-semantic mapping system that utilize multi-modal sensing to construct a lightweight metric-semantic mesh map around $10ms$. This fast performance is benefited from the GPU acceleration. The resulting map is then integrated into a navigation system, enabling the map-based localization and enhancing the traversability extraction of ground. Real-world experiments are done to validate the presented mapping and navigation system.**

I. Introduction

As the basis of localization and navigation, mapping is of growing importance in robotics. Mapping is the process to build up an internal representation of environments which can be operated by algorithms [11]. As the common type, metric maps (also referred to as "geometric maps") describe geometry of a scene and are usually defined by positions of landmarks, distance to obstacles, or binary values to indicate free and occupied space, which are critical for robots to optimize a smooth and collision-free trajectory. However, metric maps have difficulty in maintaining the long-term consistency since geometric features are sensitive to illumination and structural changes. Also, metric maps have limitation in encoding human-readable information. It is inconvenient for robots to execute abstract human instructions (*e.g.,* "navigate to the library" and "follow driving rules")

Metric-semantic mapping [9] is the capability to group semantic concepts into metric maps. The inclusion of human-labeled information facilitates many tasks such as scene abstraction [9] and exploration [1]. In this paper, we focus on the autonomous navigation task of ground robots in complicated environments with abundant semantic elements. A typical scenario is shown in Fig. 1, where many different objects such as trees and buildings appear. It is also composed of the sidewalk that is designed for pedestrains. By encoding human-prior knowledge, the semantic map can guide the vehicle to find a collision-free path along the road without intersecting with the sidewalk and grassland. But it is commonly challenging for geometry-based traversability extraction methods since the structures of roads, sidewalks, and grass are similar. Therefore, in this paper, we will study the online metric-semantic mapping method and explore its role in outdoor navigation systems.

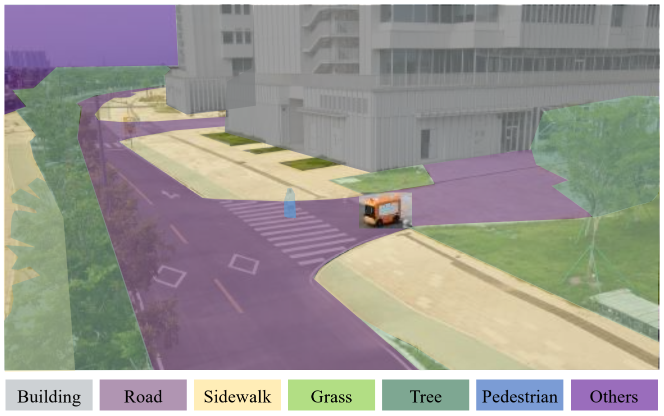


Fig. 1. To successfully navigate in the unstructured environment or conduct high-level or interactive tasks for a robot, semantic information that categorizes surrounding objects at a human-readable format is required.

A. Challenges

We consider that a desirable metric-semantic mapping approach should fulfill the following requirements:

1) **Accuracy:** The approach needs to construct the map that is close to real-world environments using onboard sensor data. But the quality of construction is affected by factors such as measurement noise, different view angles, and insufficient observations.
2) **Efficiency:** Mapping is a typical time-consuming task since thousands of map elements are queried and updated for new measurements. It is particularly important to guarantee the real-time performance for high-resolution or large-scale mapping applications.
3) **Versatility:** The resulting metric-semantic map should support a diverse array of functions, including but not limited to localization, motion planning, and visualization in a human-understandable format.

B. Contributions

Our first contribution is to propose an online mapping system that uses LiDAR-visual-inertial sensing to estimate the robot's states and build a lightweight metric-semantic mesh map of environments. Fig. 2 shows the architecture of the mapping system. Building upon the NvBlox library[1], the mapping utilizes the signed distance field (SDF)-based

[1]https://github.com/nvidia-isaac/nvblox

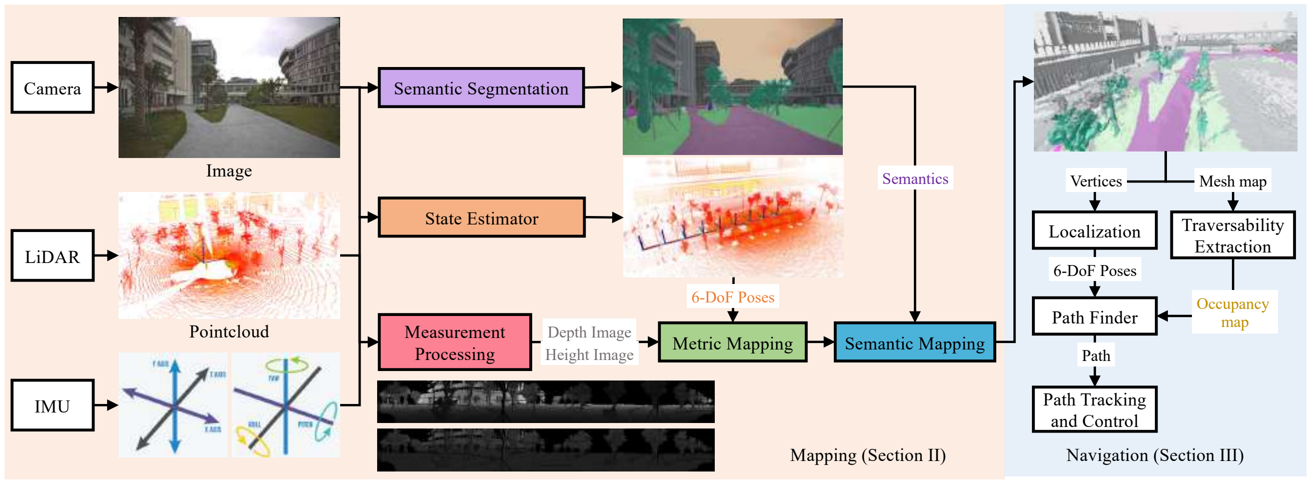


Fig. 2. The pipeline of the mapping and navigation system.

representation due to the advantage in constructing surfaces with sub-voxel resolution. It consists of four modules.

1) **State Estimator** is a LiDAR-visual-inertial odometry (LVIO) module implementing the EKF to simultaneously estimate real-time sensors' poses with a local and sparse color point cloud.
2) **Semantic Segmentation** is a pretrained convolutional neural network that assigns a class label to every single pixel of each input image.
3) **Metric-Semantic Mapping** takes sensors' measurements and poses as input, and constructs a global 3D mesh of environments using the implicit SDF-based volumetric representation with semantic annotations from the 2D pixel-wise segmentation. It is implemented in CUDAs and thus achieves the real-time performance. The original projective distance calculation is improved for more accuracy and complete (*e.g.,* less holes) mesh generation.
4) **Traversability Analysis** identifies drivable regions by jointly analyzing geometric and semantic properties of the resulting mesh map.

Our second contribution is to present an autonomous navigation system that utilizes the resulting metric-semantic map in localization and motion planning algorithms. We highlight that the introduced semantic information encodes human instructions, guiding a mobile robot to safely navigate through complex environments.

II. Mapping

A. State Estimator

The state estimator implements LiDAR-visual-inertial odometry (LVIO) that is presented in [6]. The LVIO consists of the LIO subsystem and the VIO subsystem, estimating sensors' poses and local maps in a coarse-to-find manner.

B. Semantic Segmentation

Confidence-aware semantic segmentation combined with prototype learning is used, composed of a off-the-shelf segmentation backbone [12] with a customized segmentation head and a confidence head. Specifically, in the segmentation head, each class is represented by a non-learnble prototype which is updated by online clustering of the data points using Optimal Transport [10]. Inspired by [7], to guide the network to pay more attention to the areas where the predictions are uncertain, the confidence head predicts pixel-wise aleatoric uncertainty [5] using images, semantic predictions as inputs, supervised by semantic ground truth. To train a neural network that has promissing performance, we prepare 3000 images from the public CityScapes urban dataset [2] and 1000 images from the self-collected campus dataset.

C. TSDF-Based Volumetric Mapping

After obtaining the undistorted 3D scans (e.g., LiDARs, RGB-D depth maps) and labeled images with associated poses, our metric-semantic mapping module incrementally builds a dense map for 3D reconstruction.

1) Measurement Preprocessing: This paper utilizes the LiDAR as the range sensor due to its active and accurate advantage in measuring depth. With knowing the specifications of a LiDAR (*i.e.,* horizontal and vertical angular resolution $\Delta\phi$ and $\Delta\theta$, the starting vertical angle θ_0), we can project the undistorted point cloud onto a depth image D and height image H without much information loss. Such an image-type representation is both lightweight (100KB *v.s.* 10MB) and easily processed in parallel.

2) Distance Calculation in Metric Mapping: Our mapping divides the space into voxels that are ordered according to their coordinates. To enable the dynamic insertion and deletion, voxels are managed and queried using the hashing approach. The original NvBlox implementation calculates the projective distance (*i.e.,* the distance along the sensor ray to the measured surface) of each voxel. For each incoming depth and height image, it iterates through each pixel and then raycats the pixel until it finds the voxel which intersects with the ray. However, this method is prone to overestimate the actual Euclidean distance to the nearest surface in cases where the sensor ray is not perpendicular to the surface locally. Therefore, we implement the non-projective method presented in [8], where the local planar information of the surface is used, to better approximate the true distance of voxels.

3) Semantic Mapping: Given the labeled image and the associated pose, we can retrieve all visible voxels within the camera frustum by raycasting. We only preserve voxels that stay within the TSDF truncation distance (i.e., near the surface) and are initialized (i.e., non-zero weight). Each voxel is projected onto the image plane to obtain the semantic label. Each semantic voxel stores a vector of label probabilities. We iteratively update probabilities of each voxel to improve the semantic consistency using an iterative Bayesian filter. The global mesh is finally extracted using the marching cubes algorithm, where labels of each vertex are propagated from the semantic voxel.

D. Traversability Extraction

The definition of traversability should consider both robots' mobility and human instructions. Robots' mobility is typically formulated according to robots' kinodynamic property. For robots that have off-road capability such as quadruped robots, grassland should be traversable. Regarding the latter factor, the introduced semantic information of maps is beneficial. This section proposes a traversability extraction method that jointly consider both geometric and semantic properties from the resulting metric-semantic mesh map.

1) Analysis of Geometric Properties: We analyze geometric properties of the map to determine whether a vertex is traversable or not: height difference v_{hd}, steepness v_s, and roughness v_r. The "height difference" and "steepness" are used to indicate the risk of collision, while the "steepness" indicates the changing height of terrain. A vertex is traversable and selected if its v_{hd}, v_s, and v_r are larger than thresholds.

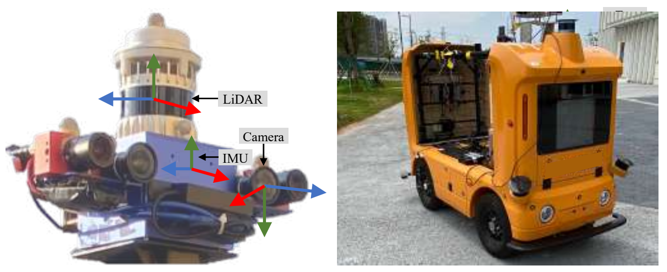
- **Height Difference** refers to the maximum difference in elevation between two points within a local region (*i.e.,* a ball $\mathcal{B}$ with radius r): $v_{hd} = \max \|\boldsymbol{v}_i - \boldsymbol{v}_j\|$, where $\boldsymbol{v}_i, \boldsymbol{v}_j \in \mathcal{B}$.
- **Steepness** refers to the degree of incline of a surface: $v_s = \arccos(\boldsymbol{n}_{\boldsymbol{v}_i})$.
- **Roughness** refers to the irregularities and unevenness of a surface: $v_r = \frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{v} \in \mathcal{B}} \boldsymbol{n}_{\boldsymbol{v}}$.

2) Analysis of Semantic Properties: Selected vertices are checked with their labels, indicating objects' categories. Several vertices are unlabeled since they are not visible by cameras. For a specific robot, we define a traversability rule human knowledge. Taking the our vehicle platform (shown in Fig.3(b)) as an example, only roads are drivable. Vertices that are not labeled as "Road" are discarded.

III. Navigation System

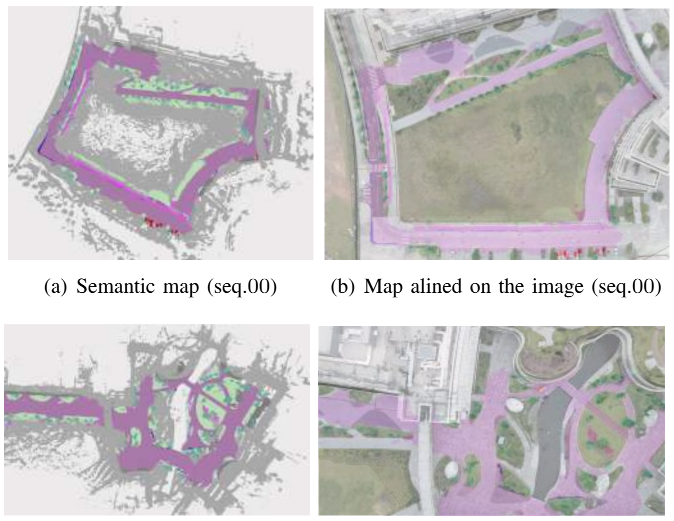
The navigation system consists of these modules:

1) **localization**: We design a coupled LiDAR-GPS-encoder localization method to estimate the robot's poses at real time. The mesh map is use to enforce the localization accuracy by being registered with the current scan.



(a) Mapping device (b) Autonomous vehicle

Fig. 3. The mapping device (a) that consists of a high-resolution LiDAR and camera is used to collect data for the environmental mapping. The real-world vehicle (b) provides a platform for testing the navigation system.



(a) Semantic map (seq.00) (b) Map alined on the image (seq.00)

(c) Semantic map (seq.01) (d) Map alined on the image (seq.01)

Fig. 4. Semantic maps are created from data of the sequence00 (top) and sequence01 (bottom), respectively. Maps are manually aligned with images to show the specific meaning of labels.

2) **Path Finder**: We project vertices of the traversable region onto the 2D occupancy grid map. During navigation, we implement the hybrid A* algorithm [3], one of the search-based planners, to calculate a trajectory from the start point to the end point by considering the kinodynamic constraints of our vehicle.
3) **Path Tracking**: The path tracking module is composed of a lateral trajectory tracking and a longitudinal speed controller. The tracking controller outputs the desired steering rate based on the cross-track and heading error. The cross-track error is defined as the distance between the point on the path closest point on the path with the front axle of the vehicle. The speed controller utilizes the PID controller to adjust the vehicle's speed. The input of the controller is the desired speed of the closest reference point to a speed control chassis.

IV. Experiment

A. Implementation Details

The mapping system is mainly implemented with C++, while the semantic segmentation is implemented with the

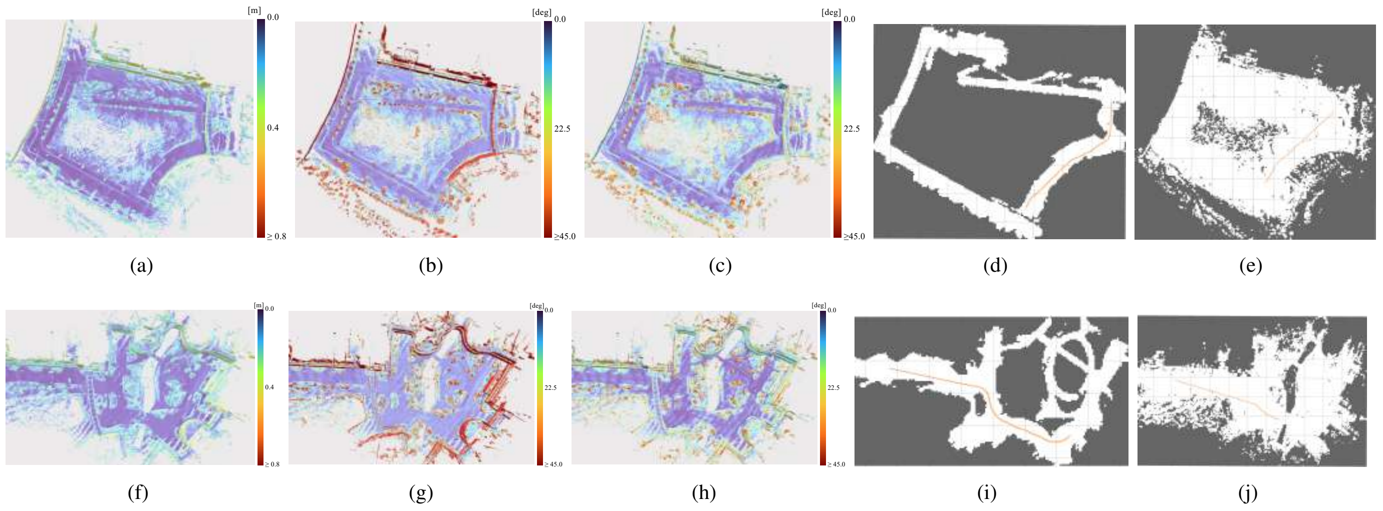


Fig. 5. Visualization of geometric properties of the resulting metric-semantic mesh map and projected 2D occupancy maps for navigation on sequence00 (top) and sequence 01 (bottom): (a)(f) height difference, (b)(g) steepness, (c)(h) roughness, (d)(i) occupancy map using semantic information, and (e)(j) occupancy map without using semantic information. The yellow lines in (d)(i) and (e)(j) indicate the found navigation paths, where the path in (d)(i) does not intersect with untraversable regions.

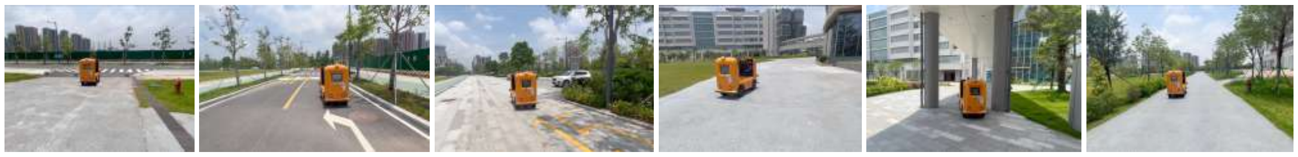


Fig. 6. Autonomous navigation in the scenario whose metric-semantic map is built on the sequence00.

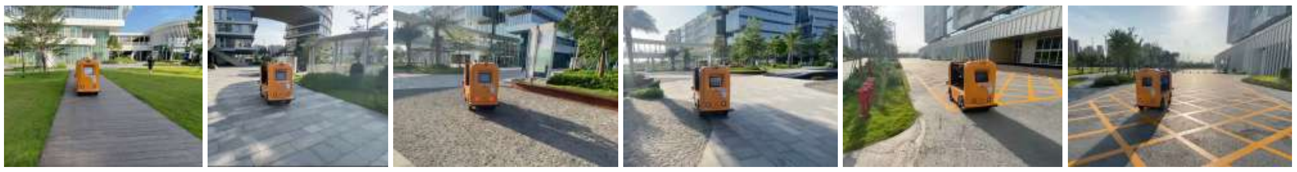


Fig. 7. Autonomous navigation in the scenario whose metric-semantic map is built on the sequence01.

Pytorch library. We use a handheld multi-sensor device [4] (see Fig. 3(a)) to collect mapping data. This device contains OS1-128 LiDAR, two FILR BFS-U3-31S4C global-shutter color cameras, and one STIM300 IMU. We also deploy the navigation system on a ground robot (see Fig. 3(b)). The size of voxels in mapping is set as $0.25m$.

B. Mapping Experiments

1) Qualitative Results: We collected two typical sequences (00–01) that contain objects including roads, sidewalks, terrain, vegetation, vehicles, and buildings. Fig. 4 visualizes the resulting metric-semantic map.

2) Timing: We take the the sequence 00 as an example. Most of computations of mapping are done in GPUs (NVIDIA GeForce RTX 3080Ti) and very fast. The metric mapping module that processes each new frame takes an average of $2.0ms$, including gathering visible voxels by ray tracing as well as updating their distance and weight. The semantic mapping module needs to find visible voxels and update their class probabilities via the Bayesian filter, costing around $12.6ms$. The mesh generation takes an average of $22.5ms$ to update the global metric-semantic mesh at a constant frequency, which is affected by the scale of scenarios.

C. Navigation Experiments

1) Results of Traversability Extraction: We visualize the geometry-based traversability in Fig. 5 on resulting maps. Objects that are not traversable such as cars, buildings, and trees are easily distinguished and filtered out by setting threshold from geometric information, i.e., "height difference", "steepness", and "roughness". But the sidewalk and grassland which are not traversable for vehicles have to be distinguished by semantic information. By combining all geometric and semantic information, we obtain the traversable map, and then generate occupancy maps for the subsequent motion planning, as shown in Fig. 5(d) and Fig. 5(i).

2) Results of Real-World Navigation: The vehicle is given a set of start point and end point to perform missions. It uses the

3D metric-semantic map for localization and the occupancy map to find a safe and collision-free path. Fig. 6 and Fig. 7 record the testing process. The average speed is $3m/s$.

V. Conclusion

We introduced a online metric-semantic mapping system for autonomous robot navigation. The system consists of several key modules, including state estimator, semantic segmentation, TSDF-based metric-semantic mapping, and extraction of traversable regions. We further utilized the resulting map in a navigation system for localization and identifying the traversbility of ground. We believe that this paper should provide a new framework and insight of the high-level representation of outdoor environments.

References

[1] Arash Asgharivaskasi and Nikolay Atanasov. Semantic octree mapping and shannon mutual information computation for robot exploration. *IEEE Transactions on Robotics*, 2023.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[3] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Practical search techniques in path planning for autonomous driving. *Ann Arbor*, 1001 (48105):18–80, 2008.

[4] Jianhao Jiao, Hexiang Wei, Tianshuai Hu, Xiangcheng Hu, Yilong Zhu, Zhijian He, Jin Wu, Jingwen Yu, Xupeng Xie, Huaiyang Huang, et al. Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3851–3856. IEEE, 2022.

[5] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[6] Jiarong Lin and Fu Zhang. R3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10672–10678. IEEE, 2022.

[7] Jiawei Liu, Jing Zhang, and Nick Barnes. Modeling aleatoric uncertainty for camouflaged object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1445–1454, 2022.

[8] Yue Pan, Yves Kompis, Luca Bartolomei, Ruben Mascaro, Cyrill Stachniss, and Margarita Chli. Voxfield: Non-projective signed distance fields for online planning and 3d reconstruction. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5331–5338. IEEE, 2022.

[9] Antoni Rosinol, Andrew Violette, Marcus Abate, Nathan Hughes, Yun Chang, Jingnan Shi, Arjun Gupta, and Luca Carlone. Kimera: From slam to spatial perception with 3d dynamic scene graphs. *The International Journal of Robotics Research*, 40(12-14):1510–1546, 2021.

[10] Mikołaj Sacha, Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protoseg: Interpretable semantic segmentation with prototypical parts. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1481–1492, 2023.

[11] Sebastian Thrun et al. Robotic mapping: A survey. 2002.

[12] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.