2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Inverse Perspective Mapping-Based Neural Occupancy Grid Map for Visual Parking

Xiangru Mu, Haoyang Ye, Daojun Zhu, Tongqing Chen and Tong Qin*

***Abstract*— Sensing environmental obstacles and establishing an occupancy map of surroundings are critical to achieving automated parking for autonomous vehicles. This paper presents a method to obtain surrounding occupancy information from inverse perspective mapping (IPM) images. This method uses the easily-accessed pseudo-labels from LiDAR to supervise a visual network, which can detect occupied boundaries of obstacles. Fusing this visual occupancy with ego-motion information, we develop a multi-frame fusion approach to build a local OGM to realize online environment mapping. Compared with other learning-based occupancy approaches, our method does not require time-consuming and labor-intensive labeling for the environment due to the ground truth of surrounding occupancy coming from LiDAR easily. The proposed method achieves LiDAR-like performance with pure visual inputs, which greatly decreases the cost of real products. Experiments on driving and parking environments prove that our method can accurately sense surrounding occupancy information and build a robust occupancy map of the environment.**

I. INTRODUCTION

Auto Parking Assist (APA) has become a hotspot in the development of self-driving vehicles, which can let humans get rid of complicated parking operations and improve parking success rates. And the key to realizing the APA is to sense the information of obstacles in the environment in real time and build a map of the environment. LiDARs and cameras are the most commonly used sensors to realize obstacle detection. LiDAR can actively detect the distance of objects in the environment, but the disadvantage is that the cost of LiDAR is high; The camera has the advantages of low cost and high resolution, but the disadvantages are that it lacks distance measurements and is heavily influenced by weather and lighting. For commercial production, low-cost sensors are preferred. So we want to use cameras to realize LiDAR-like performance for low-end cars without LiDAR.

Traditional vision methods [1, 2] cannot adapt to the texture-less parking lot, and are prone to unstable feature detection and matching. Based on deep learning, semantic information about the environment can be extracted, and adding semantic constraints will make feature matching more robust [3]. However, semantic-based methods require expensive manual annotation. The whole surrounding information can be obtained by using multiple cameras. More and more visual perception methods are conducted under bird's eye view (BEV). The BEV methods [4, 5] fuse the data of multiple sensors and uniformly express the data in the bird's eye view. Compared with the solution based on the image view, the perception under the BEV avoids scale and occlusion issues, which facilitates the development of a subsequent planning and control module.

All authors are with IAS BU, Huawei Technologies, Shanghai, China. Tong Qin is the corresponding author. muxiangru, zhudaojun, yehaoyang1, chentongqing, qintong@huawei.com

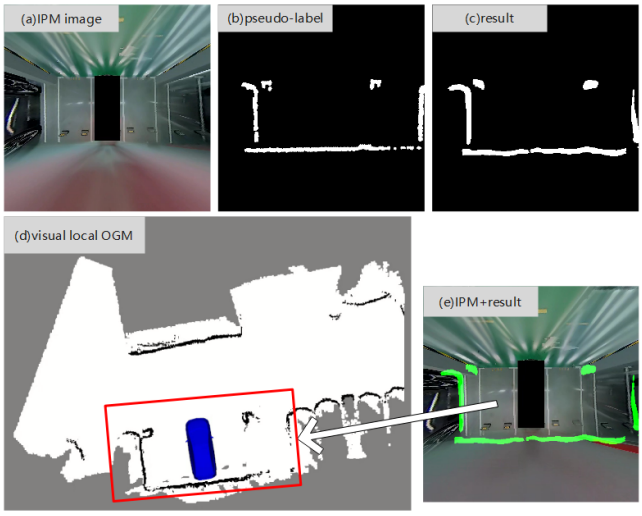


Fig. 1. The occupancy result from our neural network and local OGM. (a) IPM image from a vehicle in the parking plot. (b) The pseudo-label from LiDAR after the post process. (c) The occupancy segmentation results from our network. (d) The output of final visual local OGM. (e) Alignment of IPM image with occupancy detection from our network.

Occupancy grid maps (OGM) [6] are often used by self-driving cars to represent the environment and present whether the environment is occupied by obstacles. Based on the grid map format, we can build the occupancy map by stacking multi-frame visual information.

In conclusion, APA tasks require the use of low-cost sensors to detect obstacles in the environment in real time. To solve the above problem, this paper proposes a method to obtain surrounding occupancy information using cameras. The method uses a neural network to detect occupancy information from images. The network is supervised by LiDAR, from which pseudo occupancy labels are easily accessed. Compared with other methods, which detect freespace [7] and occupancy boundaries [8], our method doesn't require time-consuming and labor-intensive labeling. In addition, a fusion module incrementally builds the OGM and filters noise by stacking multiple frames with the ego motions. The proposed method can realize online environment mapping, using only visual information, which avoids using expensive LiDAR sensors during deployment.

The contributions of this paper are summarized as follows,

- A low-cost obstacle occupancy network based on IPM image is proposed.

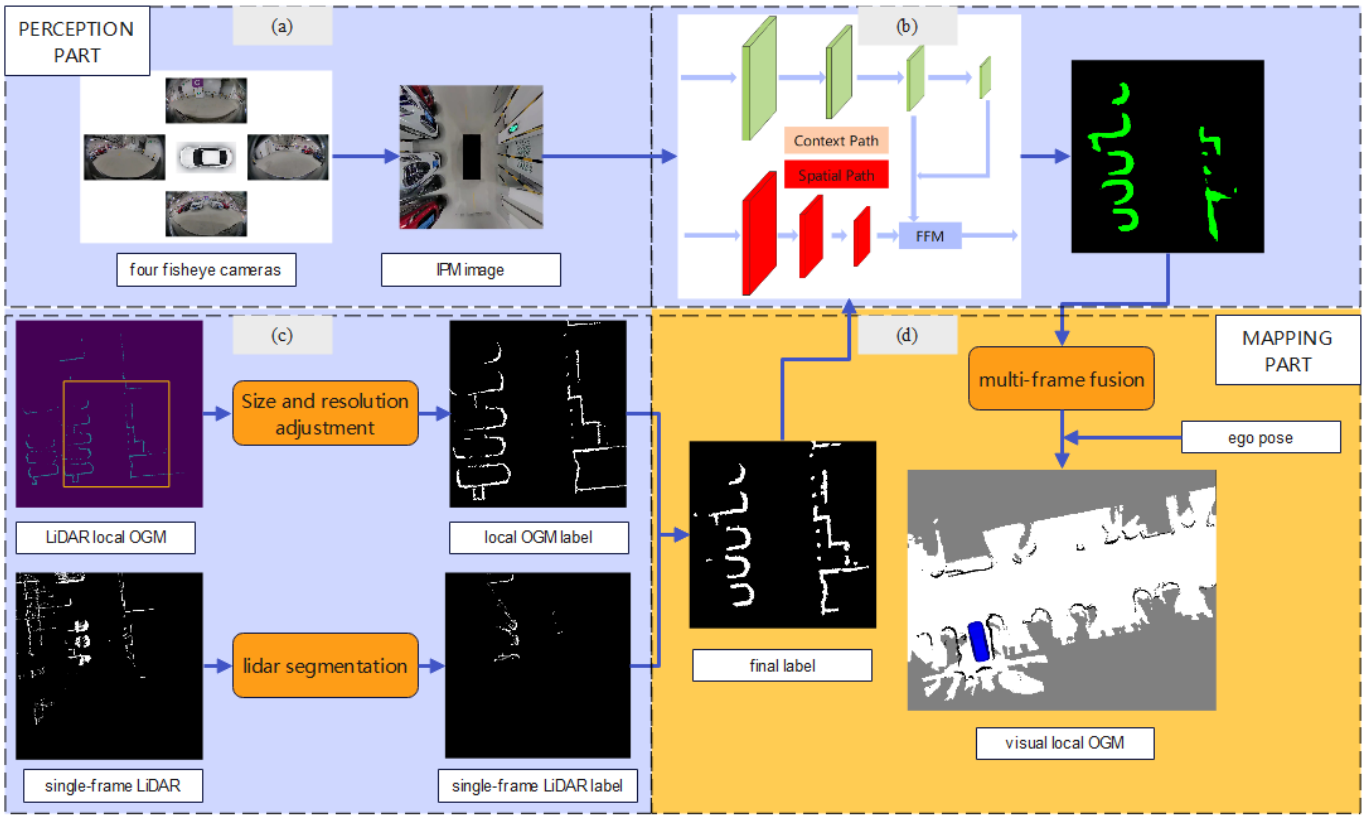


Fig. 2. The framework of the proposed system. The inputs of our network are IPM images and pseudo-labels. We obtain visual occupancy boundary information through the network and build visual local OGM by multi-frame fusion with vehicle poses.

- A fusion module which fuses multi-frame outputs from the obstacle occupancy network to generate an accurate OGM is proposed.
- The functionality of the proposed method is validated on real-world self-driving cars.

II. RELATED WORK

There is extensive research related to parking for self-driving vehicles. In this paper, we focus on obstacle occupancy detection and mapping.

A. Obstacle occupancy detection

By detecting obstacles, self-driving cars can build an obstacle map of the environment for navigation.

The most direct method is to use LiDARs to obtain the occupancy boundary of the obstacle directly, and then build the occupancy grid map [9, 10]. Other methods [11, 12] use the camera to calculate the depth information of environmental obstacles and extract the boundary of obstacles. The LiDAR-based solution needs to use expensive LiDAR sensors, while the computing depth of the vision-based solution is affected by factors such as lacking texture.

In addition to these two traditional approaches, more and more scholars are using deep neural networks to detect obstacles. Recently, Xiang et al. [7] proposed an obstacle boundary detection method based on an IPM image. This method uses the edge extracted by the IPM image and the free space obtained by semantic segmentation to obtain useful edges for mapping. Richter et al. [8] proposed a semantic grid map construction method based on semantic segmentation of monocular and multiple images. Hoermann et al. [13] trained a single-stage deep convolutional neural network to provide object hypotheses comprising of shape, position, orientation, and an existence score. Kumar et al. [14] explore an alternate approach of training using sparse LiDAR data as ground truth for depth estimation for fisheye camera, Maier et al. [15, 16] uses the knowledge about obstacles identified in the laser data to train visual classifiers based on color and texture information in a self-supervised way. Our methods uses the LiDAR information to supervise the network in bev view. In bev view, the network have larger range of perception and the network output avoids the inaccurate transform from focal view to bev view. In this paper, we directly use the IPM image as the network input to obtain environmental obstacle boundaries. LiDAR local OGMs are taken as pseudo-labels to supervise the network. This eliminates the need for complex and time-consuming manual labeling of data, as well as reduces the usage of expensive LiDAR sensors in the deployment.

B. Visual SLAM v.s. Semantic Visual SLAM v.s. LiDAR SLAM

APA tasks require an accurate representation of the environment. Common methods include visual SLAM, semantic visual SLAM, and LiDAR SLAM.

Advanced visual SLAM methods include SVO [17], VINS [18], OPEN-VINS [19], ORB-SLAM3 [2] and so on. These

methods extract environmental feature points and use descriptors to match feature points. Due to the lack of illumination and texture in underground garages, these methods are suffered from tracking loss, which limits their usage in parking tasks. Meanwhile, most visual maps are sparse and require further post-processing before they can be used by self-driving cars.

Semantic visual SLAM [20]–[24] compensates for shortcomings of traditional visual SLAM by leveraging additional semantics and geometric constraints, and achieves more stable performance. These methods extract more robust features, such as curbs, signs, and lane lines, to enhance the accuracy of feature extraction and matching. However, the semantic detection module often requires a lot of manually labeled data to train, which is time-consuming and labor-intensive.

LiDAR SLAM can create a 3D point cloud map or an occupancy grid map. The commonly used LiDAR SLAM solutions include LOAM [25], LeGO-LOAM [26], and so on. Compared with the visual SLAM, LiDAR SLAM needs to spend much time on processing a large amount of point cloud data. In practice, commercial vehicles are preferred to reduce the use of LiDAR for cost reasons. To this end, we seek to reduce dependency on LiDAR and pay more attention to visual SLAM solutions.

Considering the advantages and disadvantages of the above methods, we propose a method to build an environment map only using the images from cameras as inputs. By learning the occupancy information from LiDAR, the proposed system, with only cameras, can achieve nearly the same performance as the one with LiDARs.

III. SYSTEM OVERVIEW

The framework of our system consists of two parts, a perception part, and a mapping part. As shown in Fig. 2, the perception part uses fisheye images as input. The four fisheye cameras are stitched into an IPM image before putting in into a detection network. The LiDAR is used to generate pseudo labels of occupation to supervise the network. We stack multiple LiDAR measurements to build the local OGM of static obstacles. Meanwhile, we add dynamic obstacles, such as pedestrians, and vehicles, from single LiDAR frame segmentation. The LiDAR local OGM and segmentation results are stacked to obtain the final pseudo-label. In the mapping part, the multi-frame fusion model stacks visual occupancy information with the vehicle's ego motion to establish the local OGM.

IV. PERCEPTION

A. IPM Image

We use four fisheye cameras, which are mounted on the vehicle at the front, rear, left, and right respectively, as shown in Fig. 2(a). These cameras are equipped with fisheye lens and look downwards. Images captured by these cameras are used to synthesize a bird's-eye view image. At first, images are transformed into a bird's-eye view by inverse projection. Based on the ground plane assumption, the inverse projection transformation is conducted as follows:

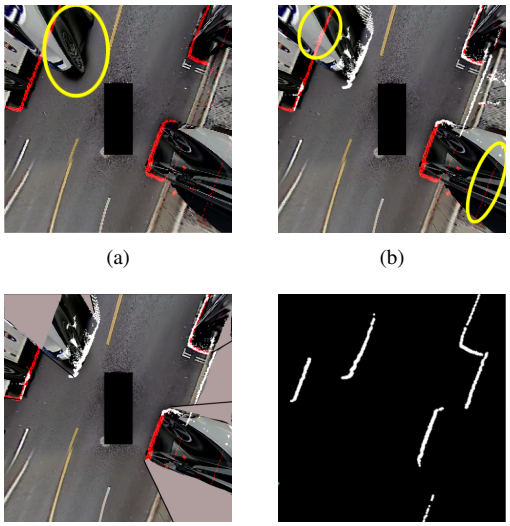


Fig. 3. The drawbacks of pseudo-labels from LiDAR local OGM. (a) shows the LiDAR local OGM (Red color represents the pseudo-label, and the yellow circle encircles unlabelled vehicles). (b) shows the stacking of the sing-frame label (white), LiDAR local OGM (red), and IPM image, and the yellow circle shows the occluded area. (c) shows the strategy to remove occluded areas. The gray area is occluded. (d) shows the input label of the neural network. Best viewed in color.

$$\frac{1}{\lambda}\begin{bmatrix} x^v \\ y^v \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \end{bmatrix}^{-1}_{col:1,2,4} \pi_c^{-1}\left(\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}\right), \tag{1}$$

where π_c^{-1} is the inverse projection, which lifts the pixel from the image coordinate into space. $\begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \end{bmatrix}$is the extrinsic matrix of each camera with respect to the vehicle frame. $\begin{bmatrix} u & v \end{bmatrix}$ is the pixel location in the image coordinate. $\begin{bmatrix} x^u & y^v \end{bmatrix}$is the position of the feature under the vehicle's coordinate. λ is the depth scalar, which can be solved in this equation. $()col : i$ means taking the ith column of this matrix. After the inverse perspective projection, we synthesize points from four images into one BEV image as shown in Fig .1(a).

B. Label Generation

To train a network to detect occupancy from IPM images, we need to prepare ground-truth labels first. We extract the ground-truth label from LiDAR observation automatically. By training a network supervised by LiDAR, the vision system can predict occupancy information from IPM images.

A 360°LiDAR is mounted on the top of the vehicle for the ground truth collection. Due to the limited scanning resolution of the LiDAR, a single LiDAR measurement is sparse, and cannot provides dense labels. To this end, we stack multiple LiDAR frames to generate a dense local OGM (Fig. 2(c)). When we fuse multiple LiDAR frames, the noisy measurement and dynamic objects are filtered out by occupancy probabilistic updating. Although the LiDAR local OGM is clean and presents stable occupancy in the surroundings, there are two drawbacks to supervising a single-frame IPM image.

The first drawback is missing dynamic occupancy. We expect the ground truth consists of both static occupancy and dynamic occupancy (e.g. moving pedestrians, cyclists, vehicles) at a certain moment, but the local OGM cannot show the dynamic occupancy as shown in Fig. 3(a). To this end, we add dynamic occupancy information from a single LiDAR frame into local OGM at a certain moment. We only need the observation of moving objects, so with the help of LiDAR segmentation directly from the vehicle's onboard processing, we extract vehicles and pedestrians to compensate for moving objects. As shown in Fig. 3(d), by combining LiDAR local OGM with single-frame LiDAR observation, we generate the ground-truth label, which consists of both stable and dynamic occupancy information.

Another drawback is that some areas which should be occluded at a certain moment are visible in LiDAR local OGM. As shown in Fig. 3(b), by stacking multiple LiDAR frames, the left corner which should be occluded in the image view, is visible. The ghost label will confuse the neural network. Therefore, the occluded part must be eliminated. As is shown in Fig .3(c), our strategy is to emit a ray from the vehicle center. Once the ray reaches the nearest occupancy boundary, the subsequent occupancy points on the ray are removed. In this way, we get the occupancy information containing dynamic and static objects. Then we intercept occupancy information in a 16×16m area around the ego-body to generate the ground-truth label, which is suited to a single-frame IPM image.

C. Network Training

The occupancy detection task is similar to semantic segmentation in the computer vision area. Therefore, we can adopt semantic segmentation networks, such as pspnet [27], hrnet [28], and regnet [29]. To ensure low-cost onboard computation, we choose BiSeNet [30] as our backbone, which consists of two paths, spatial paths, and contextual paths, as shown in Fig. 2(b). The spatial path preserves the spatial scale of the original input image and encodes rich spatial information. The context path can quickly downsample the feature map to obtain a large receptive field and encode high-level semantic context information. Finally, a Feature Fusion Module is used for two-path fusion to output the occupancy edges. We use focal loss to resolve the imbalance between positive and negative samples.

We build a multi-task training framework based on this framework, with other segmentation heads, such as lane, maker, stop line, and so on.

V. MAPPING

In the parking task, an accurate OGM with a large perception range is necessary. However, the obstacle's occupancy detected from a single IPM image is noisy due to distortion, and the perception range is limited. Therefore, we need to stack multiple frames in a fixed local frame to reduce the impact of detection inaccuracies and enlarge the perception range. We use the method similar to the traditional two-dimensional LiDAR occupancy grid mapping [31] to update the occupied area. Firstly, the detected occupied points are transformed from the image coordinate into the vehicle's frame using IPM intrinsic and extrinsic parameters. Secondly, these points are transformed into a local frame by the vehicle's ego pose. Before updating, we filter noisy points by the statistical method. The space is divided into small grids. Then the detected occupied points are used to update corresponding grids. Within a sliding window of N frames, if one grid is hit by over k points, we treat these points as inliers, otherwise, outliers, which are thrown out. In practice, we set $N = 10, k = 6$. A comparison before and after the filter is shown in Fig. 4 (a) and (b), where (a) shows the result without noisy points filtering and (b) shows the result with noisy points filtering. It can be seen that excessive noisy points are removed by the sliding-window filter.

(a) (b)

Fig. 4. The black points are the occupancy points, while the white area shows the free space. (a) shows the OGM without noisy points filtering. (b) shows the OGM with noisy points filtering. It can be seen that excessive noisy points are removed by the sliding-window filter.

The filtered obstacle points are used to update the local OGM. The map is divided into small grids, and the value in each grid indicates the occupancy probability. We use the log-odds representation of occupancy:

$$l_i^t = l(p_i^t) = \log \frac{p_i^t}{1 - p_i^t}, \tag{2}$$

where p_i^t indicates occupied probability of grid i at time t, whose value is between $[0, 1]$. l_i^t is its log-odds form. We follow the Bresenham line algorithm [32] to update the occupancy of the grid. When a grid is hit by an occupied point, the occupancy update formula is as follows:

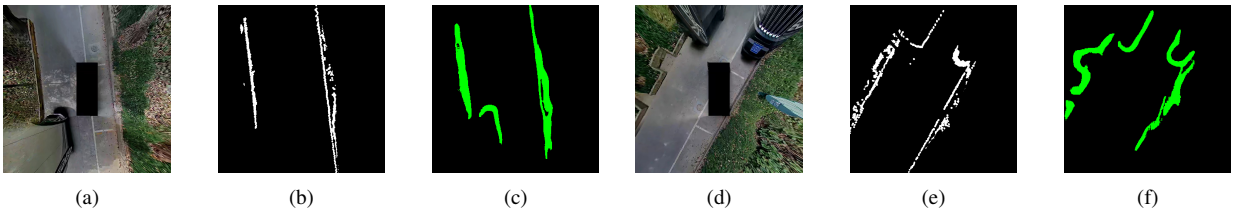
$$l_i^{t+1} = l_i^t + l_{\text{occu}} - l_0. \tag{3}$$

Meanwhile, other grids along this ray in front of the hitting point are updated as free space:

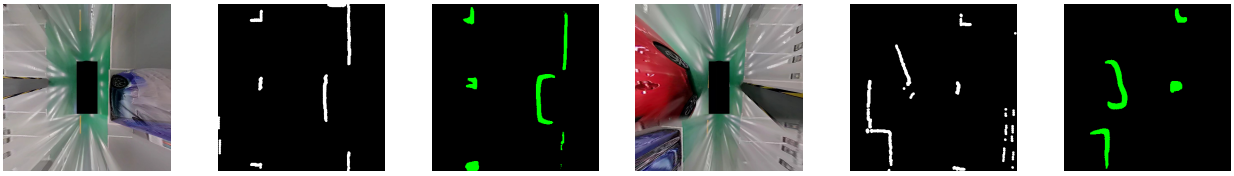
$$l_i^{t+1} = l_i^t + l_{\text{free}} - l_0, \tag{4}$$

where l_0 is the prior occupancy of grids, l_{occu} and l_{free} is probability of occupancy and free space in log-odds form. In practice, we use a local OGM with a size of 30×30 m and a grid resolution of 0.05×0.05 m. The initial prior l_0 equals to $l(p = 0.5)$. The occupancy probability l_{occu} and free space probability l_{free} are the log-odds of 0.8 and 0.2 respectively.

In this way, we build a local OGM for the visual parking scenario. The following path planning module will take this OGM to generate safe trajectories avoiding collision.



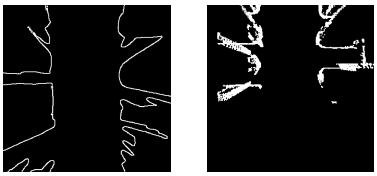
(a) (b) (c) (d) (e) (f)



(g) (h) (i) (j) (k) (l)

Fig. 5. The output of our network in different scenes. (a)(d)(g)(j) show the surrounding view image. (b)(e)(h)(k) show the pseudo label and (c)(f)(i)(l) shows the output of our network.

(a) (b) (c)



(d) (e)

Fig. 6. The results of different occupancy detection methods: (b) the pseudo LiDAR label, (c) the output by the proposed network, (d) the free-space edge extracted by the canny algorithm, (e) the single-frame LiDAR.

TABLE I

THE OUTPUT METRICS IN DIFFERENT SCENES

metrics	Underground Parking Scene	Aboveground Parking Scene
IOU ↑	0.366	0.243
ASD ↓	5.284 pixel (22.01 cm)	7.08 pixel (29.5 cm)
Surface Dice ↑	0.523	0.532

*The ↑ means the value of this metric is the higher the better, while ↓ means the value of this metric is the lower the better.

VI. EXPERIMENT

A. Data Collection

We conduct qualitative and quantitative experiments on the occupancy segmentation network and local OGM on self-driving vehicles. We use four fisheye cameras mounted on the vehicle which capture images at the frequency of 30 Hz and the resolution of 1280×720. The raw images are converted to a bird's eye view with a resolution of 800×800, which covers a 16×16 m area surrounding the vehicle. A LiDAR sensor is mounted on the top of the vehicle to generate pseudo-labels. We collected 100,000 images in more than 100 parking scenarios for training the network.

B. Results of Occupancy Segmentation Network

We collected datasets for more than ten parking scenarios to build a test set, as shown in Fig. 5. In Fig. 5(a), (d), the vehicle drives in an outdoor parking lot, while in Fig. 5(g), (j), the vehicle drives in the underground parking lot.

Qualitative Results. In Fig. 5(d), another moving car passed by, and our network accurately captured the dynamic obstacle as shown in Fig. 5(f). It's worth noting that when the LiDAR is unable to capture a vehicle from the rear blind zone, in Fig. 5(b); our network can detect the vehicle from the rear as shown in Fig. 5(c). When driving in an underground garage, both static and dynamic obstacles can be sensed using the proposed network, as shown in Fig. 5(l).

Quantitative Results. We use three metrics to measure the precision of occupancy segmentation: IOU (Intersection over Union), ASD (average surface distance), and surface dice [33], averaged over all test scenarios. We compare the proposed method with the ground truth of LiDAR. The result is shown in the table I.

We compare the proposed method with some other occupancy detection methods (Fig. 6) that are available in autonomous vehicles: the free space boundary and the single-frame LiDAR boundary. The free-space boundary is obtained by the canny edges extracted from the free space of the IPM image, as shown in Fig. 6(d). The single-frame LiDAR boundary is from the points with obstacle labels, provided by LiDAR segmentation, as shown in Fig. 6(e). The network output is shown in Fig. 6(c). Compared with other methods quantitatively in table II, our method achieves the best occupancy estimation evaluated in the above-mentioned metrics among other common methods.

C. Results of Local OGM

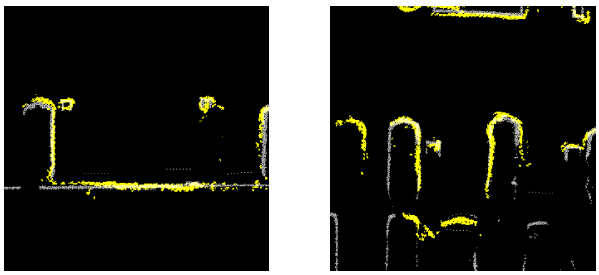
We evaluated the efficiency of the local OGM in several parking lot scenarios. The size of local OGM is 30×30 m,

TABLE II

COMPARISION OF DIFFERENT OCCUPANCY DETECTION METHODS

metrics	Our Method	FreeSpace+Canny Method	Single-frame LiDAR Method
IOU ↑	**0.237**	0.0736	0.0936
ASD ↓	**18.98 (37.96 cm)**	42.62 (85.24 cm)	21.88 (43.76 cm)
Surface Dice ↑	**0.31**	0.27	0.21

*The ↑ means the value of this metric is the higher the better, while ↓ means the value of this metric is the lower the better.



(a) (b)

Fig. 7. The result of stacking of LiDAR local OGM and visual local OGM. The gray line represents LiDAR local OGM, while the yellow line represents visual local OGM (best viewed in color).

TABLE III

THE OUTPUT VISUAL MAP BY DIFFERENT METHODS

metrics	Our Method	ORB-based OGM [34]
IOU ↑	**0.130**	0.128
ASD ↓	**20.88 cm**	33.82 cm
Surface Dice ↑	**0.46**	0.45

and the grid resolution is 0.05×0.05 m. The local OGM output at a frequency of 10 Hz.

First, we compared our visual OGM with LiDAR OGM. The registration of the visual and the LiDAR OGM is shown in Fig. 7. Qualitatively, the visual and the LiDAR OGM can be accurately matched. It's reasonable that LiDAR has a bigger perception range than an IPM image.

Quantitatively, we compared our method with an ORB-based method [34]. This method obtains the key-frames' poses and map points as occupied space to build the visual OGM. We implement that method and construct the occupied local OGM, shown in Fig. 8(b). The result of our method is shown in Fig. 8(c). We take the LiDAR OGM as the baseline, and conduct the metric comparison of these two methods in table III. It can be seen that the average distance of our method is smaller than the ORB-based method [34]. In addition, the visual mapping procedure in [34] is easily lost in underground scenes due to a lack of feature points, while our method is robust to various scenes.

VII. CONCLUSIONS

In this paper, we proposed a method using IPM images to obtain the obstacle occupancy. Pure vision is used to build an environmental occupancy map to assist in automatic parking. A neural network is trained for occupancy detection, which is supervised by LiDAR pseudo labels, avoiding intensive manual labeling. The predicted occupancy from IPM images is further processed by a multi-frame fusion module to build a local OGM of the surroundings. Experiments on multiple parking scenarios demonstrated the accuracy of our method, which has the ability to enable automated parking tasks.

In the future, we will further extend our method to 3D space to establish a 3D occupancy map.

(a) (b) (c)

Fig. 8. The visual local OGM in different methods. (b) shows the visual local OGM from [34] and (c) shows the visual local OGM from our method.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[3] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, 11 2019.

[4] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers," *arXiv e-prints*, p. arXiv:2203.17270, Mar. 2022.

[5] L. Peng, Z. Chen, Z. Fu, P. Liang, and E. Cheng, "BEVSegFormer: Bird's Eye View Semantic Segmentation From Arbitrary Camera Rigs," *arXiv e-prints*, p. arXiv:2203.04050, Mar. 2022.

[6] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, A. Koubaa, Ed. Springer, 2016, ch. 5.

[7] Z. Xiang, A. Bao, and J. Su, "Hybrid bird's-eye edge based semantic visual slam for automated valet parking," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 546–11 552.

[8] S. Richter, Y. Wang, J. Beck, S. Wirges, and C. Stiller, "Semantic evidential grid mapping using monocular and stereo cameras," 05 2021.

[9] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[10] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.

[11] A. Santana, K. Aires, R. Veras, and A. Medeiros, "An approach for 2d visual occupancy grid map using monocular vision," *Electronic Notes in Theoretical Computer Science*, vol. 281, p. 175–191, 12 2011.

[12] C. Häne, T. Sattler, and M. Pollefeys, "Obstacle detection for self-driving cars using only monocular cameras and wheel odometry," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5101–5108.
[13] S. Hoermann, P. Henzler, M. Bach, and K. Dietmayer, "Object Detection on Dynamic Occupancy Grid Maps Using Deep Learning and Automatic Label Generation," *arXiv e-prints*, p. arXiv:1802.02202, Jan. 2018.
[14] V. R. Kumar, S. Milz, M. Simon, C. Witt, K. Amende, J. Petzold, S. Yogamani, and T. Pech, "Monocular Fisheye Camera Depth Estimation Using Sparse LiDAR Supervision," *arXiv e-prints*, p. arXiv:1803.06192, Mar. 2018.
[15] D. Maier, C. Stachniss, and M. Bennewitz, "Vision-based humanoid navigation using self-supervised obstacle detection," *International Journal of Humanoid Robotics*, vol. 10, 07 2013.
[16] D. Maier, M. Bennewitz, and C. Stachniss, "Self-supervised obstacle detection for humanoid navigation using monocular vision and sparse laser data," 06 2011, pp. 1263 – 1269.
[17] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 15–22.
[18] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
[19] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4666–4672.
[20] K. Doherty, D. Fourie, and J. Leonard, "Multimodal semantic slam with probabilistic data association," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2419–2425.
[21] M. Grinvald, F. Furrer, T. Novkovic, J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3d object discovery," 03 2019.
[22] Y. Bao, Z. Yang, Y. Pan, and R. Huan, "Semantic-direct visual odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6718–6725, 2022.
[23] Z. Zhao, Y. Mao, Y. Ding, P. Ren, and N. Zheng, "Visual-based semantic slam with landmarks for large-scale outdoor environment," in *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, 2019, pp. 149–154.
[24] T. Qin, T. Chen, Y. Chen, and Q. Su, "Avp-slam: Semantic visual mapping and localization for autonomous vehicles in the parking lot," 10 2020, pp. 5939–5945.
[25] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," 07 2014.
[26] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.
[27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239.
[28] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation," *arXiv e-prints*, p. arXiv:1902.09212, Feb. 2019.
[29] I. Radosavovic, R. Prateek Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing Network Design Spaces," *arXiv e-prints*, p. arXiv:2003.13678, Mar. 2020.
[30] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation," *arXiv e-prints*, p. arXiv:1808.00897, Aug. 2018.
[31] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
[32] E. Angel and D. Morrison, "Speeding up bresenham's algorithm," *IEEE Computer Graphics and Applications*, vol. 11, no. 6, pp. 16–17, 1991.
[33] S. Nikolov *et al.*, "Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy," *arXiv e-prints*, p. arXiv:1809.04430, Sep. 2018.
[34] A. Yusefi, A. Durdu, and C. Sungur, "Orb-slam-based 2d reconstruction of environment for indoor autonomous navigation of uavs," *European Journal of Science and Technology*, pp. 466–472, 09 2020.