Proceedings of the 2025 SICE Festival with Annual Conference
September 9-12, 2025, Chiang Mai, Thailand

Mobile Robot Obstacle Avoidance Based on Optical Flow Images in Motion Planning

Satoshi Hoshino[1†] and Reo Sugata[1]

[1]Department of Mechanical and Intelligent Engineering Graduate School of Engineering Utsunomiya University, Japan
(Tel: +81-28-689-6053; E-mail: hosino@cc.utsunomiya-u.ac.jp)

Abstract: For autonomous navigation, mobile robots must avoid collisions with dynamic obstacles such as pedestrians. In previous work, we proposed a motion planner based on a CNN with RGB and depth image inputs. To enable the robot to plan avoidance motions while considering the moving direction of an obstacle, in this paper, we propose a motion planner that utilizes optical flow images as inputs. Through autonomous navigation experiments, we demonstrate that the proposed motion planner enables the robot to successfully avoid both dynamic and static obstacles.

Keywords: Mobile Robot, Motion Planning, Obstacle Avoidance, CNN, Imitation Learning, Optical Flow.

1. INTRODUCTION

For autonomous navigation of mobile robots, both global path planning from the current position to the destination and local motion planning to follow the planned path are essential capabilities. Robots can continue moving even when an obstacle is ahead by planning an avoidance path. However, obstacle avoidance based on path planning requires accurate maps obtained through environmental sensing, including obstacles.

On the other hand, in recent years, motion planners have been proposed in the field of autonomous driving that enable motor vehicles to sequentially determine control outputs, such as steering angle and acceleration/brake stroke, based on image inputs [1]. Similarly, a mobile robot equipped with a camera has been enabled to plan motions for obstacle avoidance based solely on image inputs of obstacles ahead, without relying on global path planning [2]. In addition to these related works, Convolutional Neural Networks (CNNs) have been utilized as motion planners for image-based inputs [3].

The authors have also proposed a CNN-based motion planner for obstacle avoidance in mobile robots in previous work [4]. In this paper, motion planning is defined as determining velocity and angular velocity as outputs based on sensor inputs, such as camera images. Dynamic obstacles, such as pedestrians, are the primary targets for avoidance. Therefore, Long Short-Term Memory (LSTM) [5] was integrated with the CNN to account for the time-series relationship between image inputs and their corresponding motion outputs. The CNN-based motion planner was developed using expert-provided motion instructions and imitation learning. As a result, the robot utilizing the motion planner successfully avoided dynamic obstacles at a greater distance than static obstacles, enhancing safety.

Dynamic obstacles can move at various speeds. Consequently, if a robot plans its avoidance motion based solely on relative position, it may fail to avoid obstacles in time and collide with them, depending on their speed. In our previous work [6][7], the robot successfully planned avoidance motions while considering obstacle speed by classifying into one of three states—fast, slow, or stationary—based on depth difference images. However, considering only the position and speed of obstacles may still result in collisions in certain situations, as shown in **Fig. 1**.

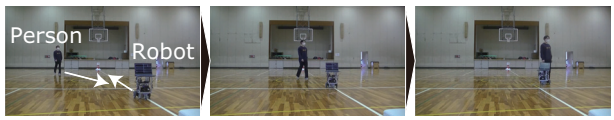


Fig. 1 Snapshots of robot planning avoidance motions for walking person

As indicated by the white arrow lines, the robot was moving straight ahead, while the person, acting as a dynamic obstacle, was walking toward the robot from the left front. As the two approached and reached the avoidance initiation distance, the pedestrian was still visible on the left side of the image from the robot's perspective. Consequently, the robot attempted to avoid the obstacle from the right, as it had been previously instructed. However, since the person was walking faster than the robot, the robot failed to avoid the obstacle and collided with it. This collision could have been prevented if the robot had avoided the obstacle to the left. In other words, the collision occurred because the robot planned its avoidance motion while ignoring the directional component of the obstacle's velocity.

The objective of this paper is to enable the robot to plan avoidance motions based on the velocity of the obstacle, in addition to its position. For dynamic obstacles, the directional component of the velocity is crucial information. Therefore, we focus on the optical flow of the obstacle in RGB images captured by a camera and propose a motion planner based on CNN for optical flow image inputs. The motion planner is developed using expert-provided motion instructions and imitation learning, as described in Section 2. In Section 3, we describe how the optical flow images used as inputs to the motion planner are generated. In Section 4, we present three types of motion planners, depending on the image inputs. Through autonomous navigation experiments in Section

† Satoshi Hoshino is the presenter of this paper.

5, we demonstrate that the robot using the proposed motion planner can avoid both dynamic and static obstacles and reach its destination. Finally in Section 6, we conclude the paper.

2. IMITATION LEARNING THROUGH MOTION INSTRUCTION

As described in Section 1, the proposed motion planner is based on a CNN. Before the imitation learning phase, an expert provides the robot with the desired motion outputs corresponding to given sensor inputs. **Fig. 2** illustrates an overview of the motion instruction phase for autonomous navigation.

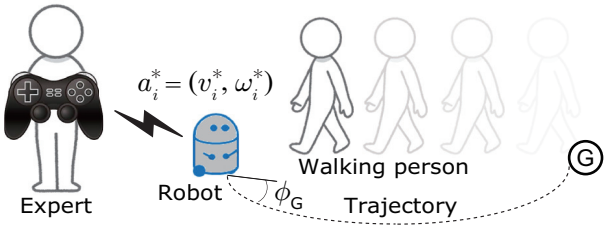


Fig. 2 Motion instruction for autonomous navigation including obstacle avoidance

The expert instructs the robot to move toward the goal destination if no obstacles are present. Conversely, if an obstacle is ahead, the expert directs the robot to avoid it. These desired motions are represented as a^*, consisting of velocity v^* and angular velocity ω^*. Thus, the instructed motion at time i is is given by $a_i^* = (v_i^*, \omega_i^*)$. While following the expert's instructions, the robot continuously records the sensor inputs—camera images and goal directions ϕ_{G}[1]—along with the corresponding motion outputs provided by the expert. In this phase, the following dataset, D, for imitation learning is collected as follows:

$$D = \{\langle o_i, a_i^* \rangle\}_{i=1}^{N}, \tag{1}$$

where o_i represents the input data, which consists of camera images and goal directions ϕ_{G}. a_i^* represents the output data, which consists of motion outputs—velocities v_i^* and angular velocities ω_i^*—instructed by the expert for the camera images. N denotes the size of the dataset.

In the motion instruction phase, the expert follows an optimal policy π^*. Under this assumption, the robot is provided with optimal motion outputs a_i^* for inputs o_i. Similar to the expert policy, the robot's policy is represented by π, which determines motion outputs a_i based on the inputs o_i, as follows:

$$a_i = \pi(o_i; \theta), \tag{2}$$

where θ represents the convolutional filters and connection weights in the CNN.

In the imitation learning phase, the network parameters, θ, in Eq. (2) are updated so as to associate the inputs with the corresponding outputs in the dataset in Eq. (1). To optimize θ, the following loss function L, based on mean squared error, is calculated by comparing the outputs of the training CNN a_i with those in the dataset a_i^*.

$$L = \frac{1}{N}\sum_{i=1}^{N}(a_i - a_i^*)^2 \tag{3}$$

Since a consists of velocity v and angular velocity ω, the total loss is given by $L = L_v + L_\omega$.

The network parameters θ in the CNN are updated to minimize the loss calculated in Eq. (3). As a result of parameter optimization through imitation learning, the robot's policy π ultimately approximates the expert policy, i.e., $\pi \approx \pi^*$. Therefore, after imitation learning, the policy π in Eq. (2) is used as the motion planner.

[1]The goal direction, ϕ_{G}, is calculated from the goal destination and robot position.

3. OPTICAL FLOW IMAGES

3.1. Visualization of Optical Flow

To enable the robot to plan avoidance motions based on the moving direction of dynamic obstacles, we focus on the optical flow in RGB images captured by a mounted camera. This is because optical flow represents moving direction on a two-dimensional image plane. For this purpose, the optical flow for all image pixels is first estimated using Gunnar Farneback's algorithm [8].

For a given pixel $\boldsymbol{x}$, the horizontal and vertical components of the estimated optical flow are denoted as $u_{\boldsymbol{x}}$ and $v_{\boldsymbol{x}}$, respectively. The direction of the optical flow $\theta_{\boldsymbol{x}}$ is then calculated as follows:

$$\theta_{\boldsymbol{x}} = \begin{cases} \arctan 2(v_{\boldsymbol{x}}, u_{\boldsymbol{x}}) & (v_{\boldsymbol{x}} \geq 0) \\ \arctan 2(v_{\boldsymbol{x}}, u_{\boldsymbol{x}}) + 2\pi & (v_{\boldsymbol{x}} < 0). \end{cases} \tag{4}$$

Moreover, the magnitude $d_{\boldsymbol{x}}$ is calculated as follows:

$$d_{\boldsymbol{x}} = \sqrt{u_{\boldsymbol{x}}^2 + v_{\boldsymbol{x}}^2}. \tag{5}$$

Based on the direction and magnitude calculated from Eq. (4) and Eq. (5), the optical flow is visualized by converting it into the HSV color space. Thus, the hue $H_{\boldsymbol{x}}$ of pixel $\boldsymbol{x}$ is derived from $\theta_{\boldsymbol{x}}$ as follows:

$$H_{\boldsymbol{x}} = \frac{\theta_{\boldsymbol{x}}}{2\pi} \times 360, \tag{6}$$

and the value $V_{\boldsymbol{x}}$ of pixel $\boldsymbol{x}$ is derived from $d_{\boldsymbol{x}}$ as follows:

$$V_{\boldsymbol{x}} = \frac{d_{\boldsymbol{x}}}{d_{max}} \times 100, \tag{7}$$

where d_{max} is the maximum magnitude of the optical flow in the image.

Fig. 3 shows a visualized optical flow in the HSV color space. For the HSV conversion of the estimated optical flow, the saturation S was fixed at 100 [%].

(a)RGB images at two time steps (b)Visualized optical flow image: I_{OF}

Fig. 3 Visualized optical flow in HSV color space

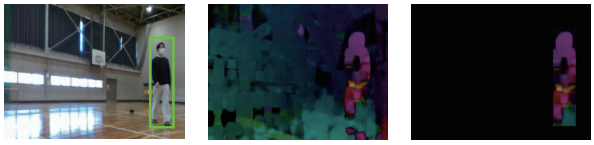
In Fig. 3(a), a person walking from the front left to the back right is visible in two RGB images captured at times $t-1$ and t. In Fig. 3(b), the optical flow of the dynamic obstacle, estimated from these two images, is visualized in the HSV color space using the calculations from Eq. (4) to Eq. (7).

In this paper, the visualized image shown in Fig. 3(b) is hereafter referred to as an optical flow image, I_{OF}. In the optical flow image, an obstacle moving from left to right is colored red, while one moving from right to left is colored blue. In other words, different moving directions are represented by different colors. Therefore, the motion planner, which takes the optical flow image as input, is expected to enable the robot to avoid dynamic obstacles by considering the moving direction based on the colors.

3.2. Optical Flow Cropped Images of Obstacles

In Section 3.1, given a fixed camera, the optical flow is estimated only from dynamic obstacles in the image. Thus, as shown in in Fig. 3(b), only the dynamic obstacle is visualized in the optical flow image.

However, when the robot equipped with the camera moves, the static background also undergoes relative motion in the image. As a result, optical flow is estimated not only from the dynamic obstacles but also from the background. Therefore, in this paper, the optical flow of obstacles is cropped from the optical flow image using object detection, as shown in **Fig. 4**. YOLO [9] is employed for object detection.



(a)Obstacle detection in RGB image (b)I_{OF} (c)Optical flow cropped image: I_{OFC}

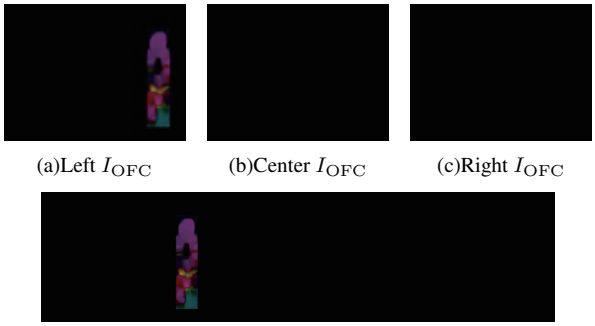
Fig. 4 Optical flow images captured by moving robot

In Fig. 4(a), a bounding box indicating a detected walking person in an RGB image is displayed. Since the robot moved along with the camera, the optical flow of the entire background was visualized in the optical flow image I_{OF}, as shown in Fig. 4(b). For this image, the pixels within the bounding box were cropped, while the remaining pixels were filled with black, as shown in Fig. 4(c). Although the robot was moving, the optical flow of only the obstacle—the target for avoidance—was visualized in Fig. 4(c), in contrast to Fig. 4(b). Thus, the optical flow cropped image I_{OFC}, as shown in Fig. 4(c), is used as an input to the motion planner.

3.3. Image Concatenation for Panoramic View

In this paper, three cameras are used to expand the robot's field of view. **Fig. 5** shows the optical flow cropped images I_{OFC} obtained from three viewpoints: left, center, and right. As an example, a walking person is visible in the left image as a dynamic obstacle.

Unlike Fig. 5(a), no obstacles appear in Fig. 5(b) and Fig. 5(c), resulting in entirely black optical flow cropped images. If these images are fed independently into the CNN, it is difficult to optimize the convolutional filters for extracting image features crucial for determining motion outputs during imitation learning.



(a)Left I_{OFC} (b)Center I_{OFC} (c)Right I_{OFC}

(d)Concatenation of three-view images: I_{OFCP}

Fig. 5 Concatenation of three optical flow cropped images I_{OFC}

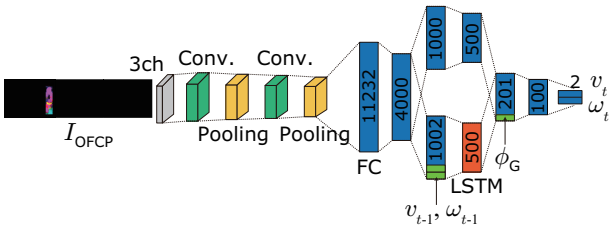
Fig. 5(d) shows a panoramic image obtained by horizontally concatenating the optical flow cropped images from the three viewpoints shown in Fig. 5(a) to Fig. 5(c). By combining these images into a single panoramic view, the optical flow of an obstacle appears more often within the inputted image. This ensures that convolutional filters can be correctly optimized through imitation learning in the CNN. Therefore, the panoramic optical flow cropped images I_{OFCP} are used as inputs for the motion planner.

4. MOTION PLANNERS USING OPTICAL FLOW IMAGES

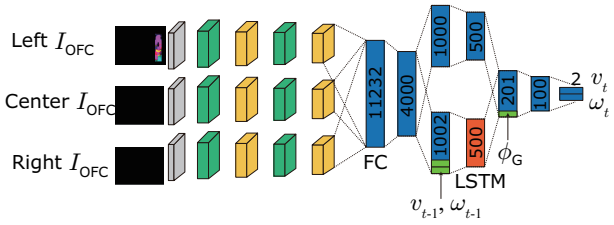
Referring to our previous work [4], the proposed motion planner is structured based on a CNN with LSTM. To enable the robot to plan avoidance motions considering the moving velocity—both speed and direction—of dynamic obstacles, optical flow images, as described in Section 3, are used as inputs to the motion planner. **Fig. 6** illustrates the structures of three types of motion planners based on different image inputs. While the convolutional layers vary, all motion planners have the same fully connected layers.

In Fig. 6(a), panoramic optical flow cropped images I_{OFCP}, as generated in Section 3.3, are used as inputs. These images undergo repeated convolution and pooling processes. In Fig. 6(b), each optical flow cropped image I_{OFC} from the three viewpoints, as generated in Section 3.2, is used as an input to separate convolution and pooling layers. In Fig. 6(c), optical flow images from the three viewpoints, as generated in Section 3.1, are simply concatenated horizontally and used as panoramic optical flow image inputs I_{OFP}.

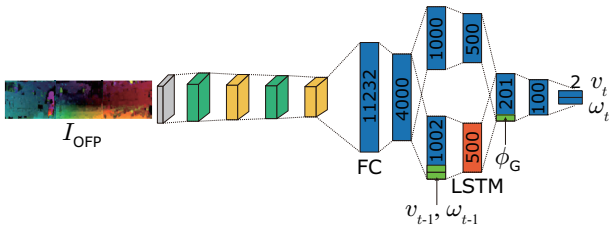
In each motion planner, two convolutional layers consist of 32 filters with a size of 5×5 and a stride of 1. The first pooling layer consists of 32 filters with a size of 2×2, while the second pooling layer consists of 32 filters with a size of 3×3, both with a stride of 2. Through these processes, obstacle features in the images are extracted and flattened to be fed into the subsequent fully



(a)Panoramic optical flow cropped images I_{OFCP}



(b)Optical flow cropped images I_{OFC}

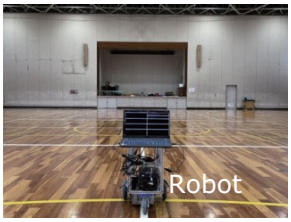


(c)Panoramic optical flow images I_{OFP}

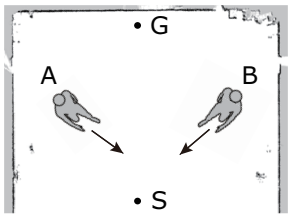
Fig. 6 Structures of motion planners using optical flow image inputs

connected layers. In the third layer, the velocity v_{t-1} and angular velocity ω_{t-1} from the previous time step $t-1$ are used as additional inputs, while the goal direction ϕ_{G} is fed into the fifth layer. Finally in the output layer, the next velocity v_t and angular velocity ω_t are derived using an activation function based on ReLU.

All the motion planners shown in Fig. 6 are developed using imitation learning based on instructions, as described in Section 2. **Fig. 7** shows the environment used for motion instruction.



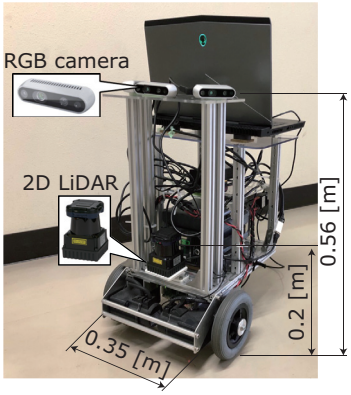
(a)Robot in actual environment



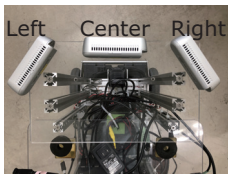
(b)Dynamic obstacles, A and B, in occupancy grid map

Fig. 7 Settings for motion instruction

In the environment shown in Fig. 7(a), the expert provides the robot with the desired motions for autonomous navigation toward the goal, including obstacle avoidance, for the three types of image inputs shown in Fig. 6. Fig. 7(b) presents an occupancy grid map of the environment. During the instruction phase, the robot moves toward the goal destination, G, which is 15 meters ahead of the starting position, S. Meanwhile, a person walks in the indicated direction from either A or B as a dynamic obstacle at a speed of 1.0 [m/s]. Since the walking speed is faster than the robot's, a collision may occur if the robot avoids obstacle A from the right and obstacle B from the left. To prevent the collisions, the robot is instructed to avoid obstacle A from the left and obstacle B from the right before continuing toward the goal destination G. **Fig. 8** provides the detail of the mobile robot.



(a)Mobile robot



(b)Top view

Fig. 8 Mobile robot equipped with three RGB cameras

The robot, shown in Fig. 8(a), moves at a speed of 0.5 [m/s], with an angular velocity ranging from -0.8 to 0.8 [rad/s]. It is equipped with a 2D LiDAR, which is used exclusively for localization based on AMCL [10] to determine the goal direction ϕ_{G}. As shown in Fig. 8(b), three RGB cameras are mounted on top of the robot, providing a horizontal field of view of 210 [deg] and a vertical field of view of 42.5 [deg].

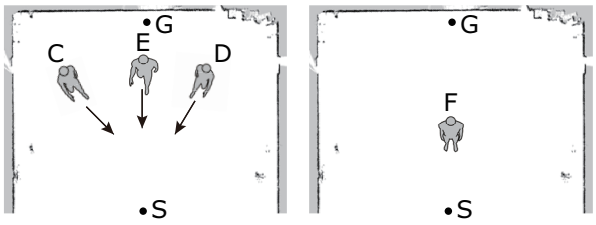
The dataset D, collected during the motion instruction phase, consists of input data, including three types of optical flow images—I_{OFCP}, I_{OFC}, and I_{OFP}—as well as goal directions ϕ_{G}. The corresponding outputs include velocity v_t^* and angular velocity ω_t^*. Additionally, the velocity and angular velocity from the previous time step $t-1$, denoted as v_{t-1}^* and ω_{t-1}^*, are also used as inputs. The dataset size, as expressed in Eq. (1), is $N = 2611$, with 1367 samples for obstacle A and 1244 samples for obstacle B. During the imitation learning phase, the network parameters θ in Eq. (2)—which include the convolutional filters and connection weights in the CNN—are optimized using Adam [11]. As a result of imitation learning over 400 epochs, the loss values in Eq. (3) for the motion planners illustrated in Fig. 6(a), Fig .6(b), and Fig .6(c) were minimized at the 160th, 77th, and 107th epochs, respectively. Thus, in the next section, the optimized policies—$\pi_{\mathrm{OFCP}}^{168}$, π_{OFC}^{77}, and π_{OFP}^{107}—are used as motion planners for autonomous navigation.

5. AUTONOMOUS NAVIGATION OF MOBILE ROBOT

5.1. Experimental Conditions

In this experiment, the effectiveness of the three motion planners presented in Section 4 is compared in terms of autonomous navigation, including obstacle avoidance.

For this purpose, three motion planners based on the policies $\pi^{168}_{\mathrm{OFCP}}$, π^{77}_{OFC}, and π^{107}_{OFP} are applied to the robot. The robot first attempts to autonomously move from S to G in the instructed environments, which are known settings, as shown in Fig. 7, while avoiding obstacle, A or B. Then, the motion planner with the highest navigation performance is further tested in different environments, as shown in **Fig. 9**, to evaluate its generalizability.



(a)Dynamic obstacle (b)Static obstacle

Fig. 9 Unknown environments with obstacles at different positions

While both the start position (S) and goal destination (G) were predefined, along with the environment shown in Fig. 7(b), obstacles C to F were placed at different positions. In Fig. 9(a), a person walks in the direction indicated by the arrows from positions C, D, and E, as a dynamic obstacle at a speed of 1.0 [m/s]. In contrast, in Fig. 9(b), the person remains stationary at position F, located 5 meters ahead of the start position, serving as a static obstacle. The robot was not instructed to avoid obstacles C to F. Thus, these environments with the obstacles are unknown to the robot. Additionally, the optical flow of dynamic obstacle E and static obstacle F is relatively smaller compared to the other dynamic obstacles, C and D. For each environment containing one of these four obstacles, the robot attempts five navigation trials.

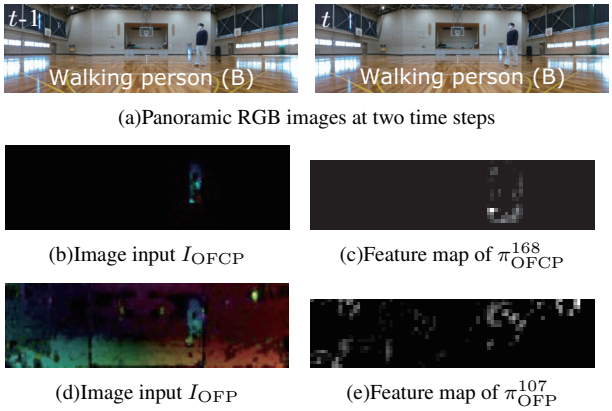
5.2. Navigation Results

To evaluate the navigation performance of each motion planner, we focused on whether the robot successfully reached the goal destination, G, after avoiding the obstacle. **Table 1** shows the number of trials in which the robot successfully reached G.

Table 1 Results of five navigation trials from S toward G in known environments

Environment w/ obstacle	Motion planner		
	$\pi^{168}_{\mathrm{OFCP}}$	π^{77}_{OFC}	π^{107}_{OFP}
A	5	2	0
B	5	0	0

The robot using motion planner $\pi^{168}_{\mathrm{OFCP}}$ successfully completed autonomous navigation in all 10 trials. It is noticeable that the robot avoided obstacle A from the left and obstacle B from the right, as instructed by the expert. In contrast, the robot using motion planner π^{77}_{OFC} reached G only twice in the environment with obstacle A, failing in the remaining trials. The robot using motion planner π^{107}_{OFP} failed in all trials. In all failed cases, the robot collided with the obstacles. These results highlight the advantage of using a motion planner that incorporates panoramic optical flow cropped images to focus solely on the target obstacle. To further analyze this advantage, we examined the feature maps extracted in the second convolutional layer of motion planners $\pi^{168}_{\mathrm{OFCP}}$ and π^{107}_{OFP}, as shown in **Fig. 10**.



(a)Panoramic RGB images at two time steps

(b)Image input I_{OFCP} (c)Feature map of $\pi^{168}_{\mathrm{OFCP}}$

(d)Image input I_{OFP} (e)Feature map of π^{107}_{OFP}

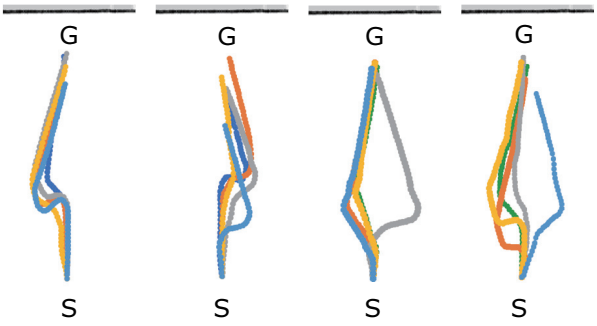
Fig. 10 Comparison of feature maps of $\pi^{168}_{\mathrm{OFCP}}$ and π^{107}_{OFP}

In Fig. 10(a), three RGB images were horizontally concatenated to create a panoramic view. In these RGB images, captured at two time steps, a walking person, B, can be seen near the middle right. Fig. 10(b) shows the panoramic optical flow cropped image generated from Fig. 10(a). For this image input, Fig. 10(c) shows that features were extracted only from the obstacle. In contrast, for a similar panoramic image without cropping the obstacle, as shown in Fig. 10(d), features were extracted not only from the obstacle but also from the entire background, as shown in Fig. 10(e). By comparing the extracted image features, we found that using panoramic optical flow cropped images allowed the robot to plan avoidance motions considering only the moving direction of the obstacle.

Regarding the difference in navigation performance between the motion planners $\pi^{168}_{\mathrm{OFCP}}$ and π^{77}_{OFC}, we compare the ratio of completely black images among all 2,611 images in the dataset D used for imitation learning. The ratio of completely black images in the panoramic optical flow cropped images used as inputs for motion planner $\pi^{168}_{\mathrm{OFCP}}$ was 28.4 [%], which is relatively low. This is because an image becomes completely black only when no obstacles appear in any of the three viewpoints. On the other hand, the optical flow cropped images used as inputs for motion planner π^{77}_{OFC} had significantly higher ratios: 79.4 [%] for the left, 76.9 [%] for the center, and 91.9 [%] for the right images. As a result, compared to π^{77}_{OFC}, the convolutional filters of $\pi^{168}_{\mathrm{OFCP}}$ were more correctly optimized to extract features of obstacles in the inputted images, which are crucial for determining motion outputs. This enabled the robot using $\pi^{168}_{\mathrm{OFCP}}$ to successfully avoid obstacles, whereas the robot using π^{77}_{OFC} failed.

Based on the results of navigation trials in known environments, the motion planner with the highest performance, $\pi^{168}_{\mathrm{OFCP}}$, was applied to the robot for navigation trials in unknown environments, as illustrated in Fig. 9. As a result, the robot successfully achieved autonomous navigation, including obstacle avoidance, in all trials.

Fig. 11 shows the navigation trajectories localized during the trials.



(a)Dynamic obstacle C (b)Dynamic obstacle D (c)Dynamic obstacle E (d)Static obstacle F

Fig. 11 Navigation trajectories including obstacle avoidance in unknown environments

As shown in Fig. 11(a) and Fig. 11(b), the robot successfully avoided obstacles C and D, which approached from different positions than during the instruction phase, by moving behind them, just as it was instructed for obstacles A and B. This result indicates that the robot planned its avoidance motions by considering the moving directions of the obstacles. Moreover, from Fig. 11(c) and Fig. 11(d), the robot successfully avoided obstacles E and F, which had smaller optical flows compared to obstacles C and D, from the left four times and from the right once, respectively. **Fig. 12** shows the panoramic optical flow cropped images for obstacles E and F, captured by the robot just before obstacle avoidance.

(a)I_{OFCP} for dynamic obstacle E (b)I_{OFCP} for static obstacle F

Fig. 12 Comparison of I_{OFCP} for dynamic and static obstacles E and F

In Fig. 12(a) and Fig. 12(b), the optical flow of both dynamic and static obstacles is visualized. This is due to the relative change in the appearance of the obstacles in the image as the robot moved. Moreover, the optical flows are highlighted in red. Since these images were used as inputs to the motion planner $\pi^{168}_{\mathrm{OFCP}}$, the robot interpreted the obstacle as moving from left to right and consequently avoided it by moving to the left.

The results demonstrate that the robot, using the proposed motion planner with panoramic optical flow cropped image inputs (as illustrated in Fig. 6(a)), can move autonomously not only in known environments but also in unknown ones. With enhanced generalizability, the robot successfully avoided obstacles, even those for which avoidance motions were not explicitly instructed. Additionally, it successfully avoided both dynamic and static obstacles with smaller optical flows and reached the goal destination.

6. CONCLUSIONS

To enable the robot to plan avoidance motions based on the velocity of dynamic obstacles, in addition to their positions, we proposed a motion planner using a CNN with optical flow image inputs. Through autonomous navigation experiments, we demonstrated that the robot could successfully plan avoidance motions by considering the moving directions of obstacles. As a result, the robot not only avoided obstacles for which avoidance motions were not explicitly instructed but also successfully moved around static and dynamic obstacles with smaller optical flows, ultimately reaching the goal destination. These results confirm the effectiveness of the proposed motion planner, which utilizes panoramic optical flow cropped images as inputs.

REFERENCES

[1] M. Bojarski *et al.*, "End to End Learning for Self-Driving Cars," *arXiv*, arXiv:1604.07316: 2016.

[2] Y. LeCun *et al.*, "Off-Road Obstacle Avoidance through End-to-End Learning," *Advances in Neural Information Processing Systems*, pp. 739–746, 2005.

[3] Y. Lecun *et al.*, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998.

[4] S. Hoshino and Y. Yoshida, "Motion Planner based on CNN with LSTM through Mediated Perception," *Annual Conference of the Society of Instrument and Control Engineers*, pp. 622–627, 2022.

[5] S. Hochreiter and J. Schmidhube, "Long Short-Term Memory," *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[6] S. Hoshino and Y. Kubota, "Mobile Robot Motion Planning through Obstacle State Classifier," *Annual Conference of the Society of Instrument and Control Engineers*, pp. 120–126, 2023.

[7] S. Hoshino *et al.*, "Motion Planner based on CNN with LSTM through Mediated Perception for Obstacle Avoidance," *SICE Journal of Control, Measurement, and System Integration*, Vol. 17, No. 1, pp. 19–30, 2024.

[8] G. Farnebäck, "Two-frame Motion Estimation based on Polynomial Expansion," *Scandinavian Conference on Image Analysis*, Vol. 2749, pp. 363–370, 2003.

[9] J. Redmon *et al.*, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[10] D. Fox *et al.*, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," *Innovative Applications of Artificial Intelligence Conference*, pp. 343–349, 1999.

[11] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Proceedings of the 3rd International Conference on Learning Representations*, pp. 12–17, 2014.