Task-Oriented Automatic Steering for AMR Utilizing Depth Vision Deep Learning

Wei-Chih Chen and Chih-Hung G. Li*, *IEEE Member*

***Abstract*— Automatic navigation for Autonomous Mobile Robots (AMRs) is a challenging problem, as the environment often exhibits large varieties of scenarios that require proper reactions of the AMRs. Whereas model-based methods may lack flexibility in dealing with highly diversified environmental features, learning-based methods may also face the problem of overwhelming data size. In this article, we proposed a task-oriented auto-steering framework for autonomous navigation of AMRs utilizing pure vision without any environmental modification. Depending on the navigation tasks such as moving forward, corner left turn, or corner right turn, the proposed framework determines the steering actions based on the depth images received by the onboard camera reflecting the real-time scenarios. The Convolutional Neural Network (CNN)-based action policymaker of each task is trained with the images annotated for proper actions. Our preliminary road test results verified the proposed framework, as the AMR successfully navigates in corridor environments encountering different scenarios such as obstacles, crossroads, and T-junctions, in addition to following the right-passing rules.**

I. Introduction

Autonomous navigation is an active research field in mobile robotics, e.g., Simultaneous Localization And Mapping (SLAM) [1]. With the rapid development in modern computing, new methods based on machine learning and deep learning [2], [3] are being developed for autonomous navigation, where visual features of the environments are collected and annotated to train a deep neural network-based classifier [4]. Previously, the authors' team proposed a preliminary version of an end-to-end automatic steering system utilizing depth images of the robot's front view for end-to-end generation of steering commands [5], [6]. In that study, basic functions such as path adjustment and corner turning were achieved with acceptable accuracy. Here, we attempted to improve the performance of the auto-steering system by including more visual patterns to build a model that responds to a larger variety of scenarios properly. Specifically, the AMR with the proposed system can automatically generate more sophisticated reactions such as adjusting its path to the implicit target line in corridor environments, passing obstacles, moving through crossroads and T-junctions, etc.

To achieve the aforementioned functions, a task-oriented steering framework was designed to generate suitable actions upon any scenarios encountered in the working environment. For example, when the AMR autonomously navigates in corridor environments, the navigation tasks can be categorized

Fig. 1. The proposed auto-steering framework utilizes the front depth views to directly generate steering commands without environmental modification.

as moving forward, corner left turn and corner right turn. When a specific task is issued, the action commands for maneuvering the AMR are automatically generated to fulfill the task. As shown in Fig. 1, the front depth camera of the AMR captures the front view of the environment; a deep neural network is trained to generate steering actions to accomplish the task.

The contributions of this work are two-folded. We proposed the task-oriented steering framework for AMR navigation in indoor corridor environments; a complete set of scenario-action pairs were defined to train an end-to-end auto-steering Convolutional Neural Network (CNN). Experiments for verification of the proposed method were also reported; the results demonstrated largely improved performance that allows AMRs to react to more complicated scenarios such as crossing the crossroads and T-junctions.

II. Methodology

A. *Task-Oriented Automatic Steering Framework*

The proposed task-oriented auto-steering framework is illustrated in Fig. 2, which includes the command layer and the detection layer. When a navigation task such as Moving Forward is activated, all the steering actions will be determined to accomplish the task. As shown in Fig. 2, Moving Forward requires the robot to follow the implicit target lines that were defined to comply with common social rules. For example, in the countries adopting right-passing rules, the target lines are in general defined at the right side of the corridor. As the target lines are implicit, the auto-steering system needs to make judicious decisions on the steering action based on the visual input that reflects the robot's pose relative to the global environment. In other words, the auto-steering system must not be confused by the large variety of

The research is supported by AXIS-Group: 209A060.

Wei-Chih Chen is with the Graduate Institute of Manufacturing Technology, National Taipei University of Technology, Taipei, Taiwan, R.O.C. (phone: +886-2-2771-2171; fax: +886-2-2776-4889; e-mail: open46037@gmail.com).

Chih-Hung G. Li* is with the Graduate Institute of Manufacturing Technology, National Taipei University of Technology, Taipei, Taiwan, R.O.C. (phone: +886-2-2771-2171; fax: +886-2-2776-4889; e-mail: cL4e@mail.ntut.edu.tw).

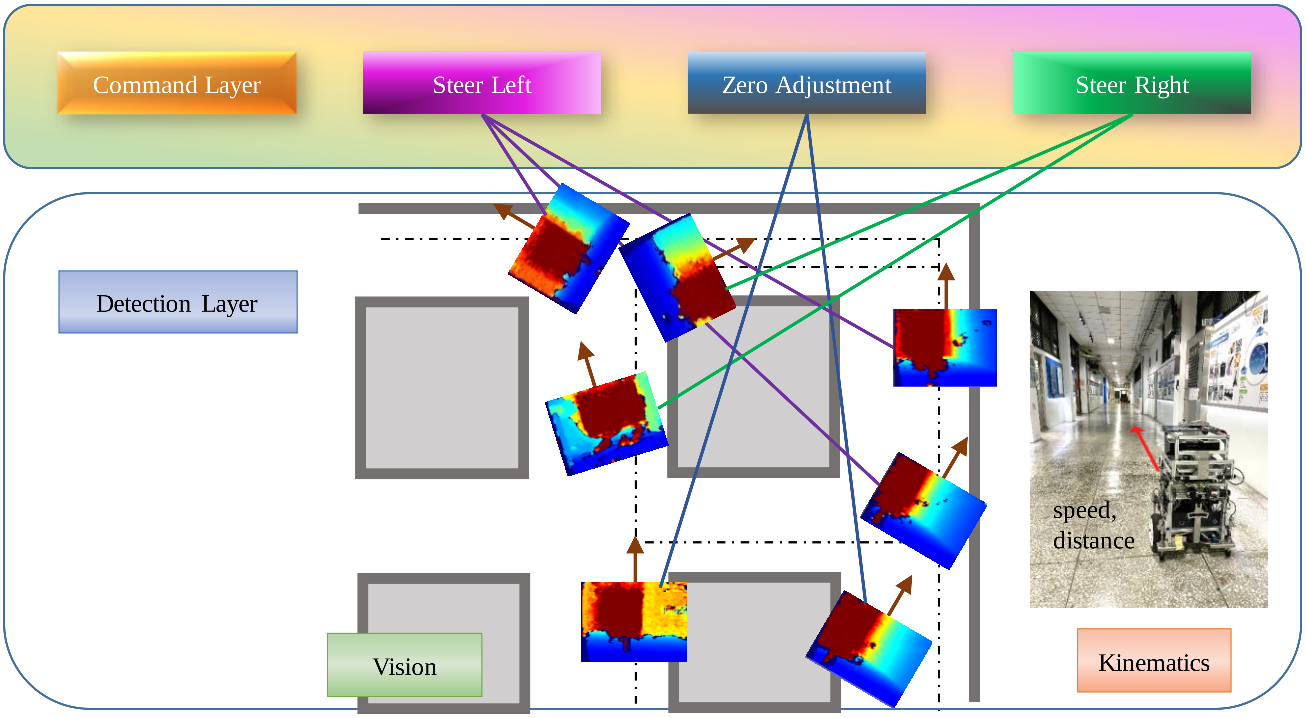

Fig. 2. The structure of the task-oriented automatic steering framework. In the example where the task of Moving Forward is activated, the CNN classifier receives front depth images and outputs steering commands; combined with the kinematic information, steering actions are determined and issued.

scenarios it may encounter but always steer correctly to accomplish the task of moving forward. Three steering actions are defined as left-steering, right-steering, and zero-adjustment. Thus, a state-dependent policymaker is formed to generate the steering command given a detected visual state. As will be clearer later, we adopted CNN as the visual state classifier, which receives the depth image of the front view and outputs the recommendation of the steering action. In addition to the visual state, kinematic information such as the robot's speed is also acquired to determine the steering intensity for better performance.

B. *Visual Training*

For each specific task, possible types of scenarios encountered in corridor environments must be identified; images reflecting similar scenario types are grouped to form the training set for the CNN classifier. Take the task of Moving Forward as an example, the 11 types of scenarios are shown in Fig. 3. As the goal of Moving Forward is to guide the robot to move on the implicit target line (dash lines in Fig. 3) along the hallways, different scenarios reflecting the structure of the environment and the pose of the robot relative to the environment are identified. For each scenario, the corresponding steering action that is needed for fulfilling the task requirement is also identified. For example, when the robot has moved to the right side of the target line and is posing toward the right wall, the action of left-steering should be taken to prevent it from hitting the wall. Note that Fig. 3 demonstrates a design for the social rule that adopts right-side moving; for the left-side moving countries, one would adjust the target line to the left side of the corridors.

C. *Convolutional Neural Network*

We adopted a popular architecture of CNN – ResNet18 [7] for the visual classifier. As discussed earlier, the training dataset consists of 11 types of scenarios, each assigned with one of three steering actions – left-steering, zero-adjustment, and right-steering. The depth images are grouped into three categories according to Fig. 3. The CNN classifier receives input images of 64×48 pixels and outputs one of the three steering actions.

A total of 15,000 depth images were collected in a large office building for training, each category containing 5000 images. When the CNN classifier is operating on a Raspberry Pi 4 onboard the robot, a maximum output frequency of 4 fps is achieved. When training the CNN classifier on a PC with a CPU of Intel Core i7-7700HQ, a GPU of NVIDIA GeForce GTX 1050 with 2 GB GDDR5, and a slot of RAM with 8 GB, the elapsed time was around 87 minutes. The training accuracy reached 99.1%; the testing accuracy and the road test results will be discussed in section III.

D. *The AMR*

To test the proposed auto-steering framework, the self-developed two-wheeled dynamically balancing robot was adopted [5], [6]. An RGB-D camera – Intel RealSense D415 was installed onboard the AMR to detect the front view for the input of the CNN classifier. When a steering action is determined, a motor command is issued to rotate a ball screw that moves the handlebar that controls the direction of the robot's motion platform. The handlebar angle determines the angular velocity of the turning platform; thus, the total rotation angle of the motion platform can be determined by the duration

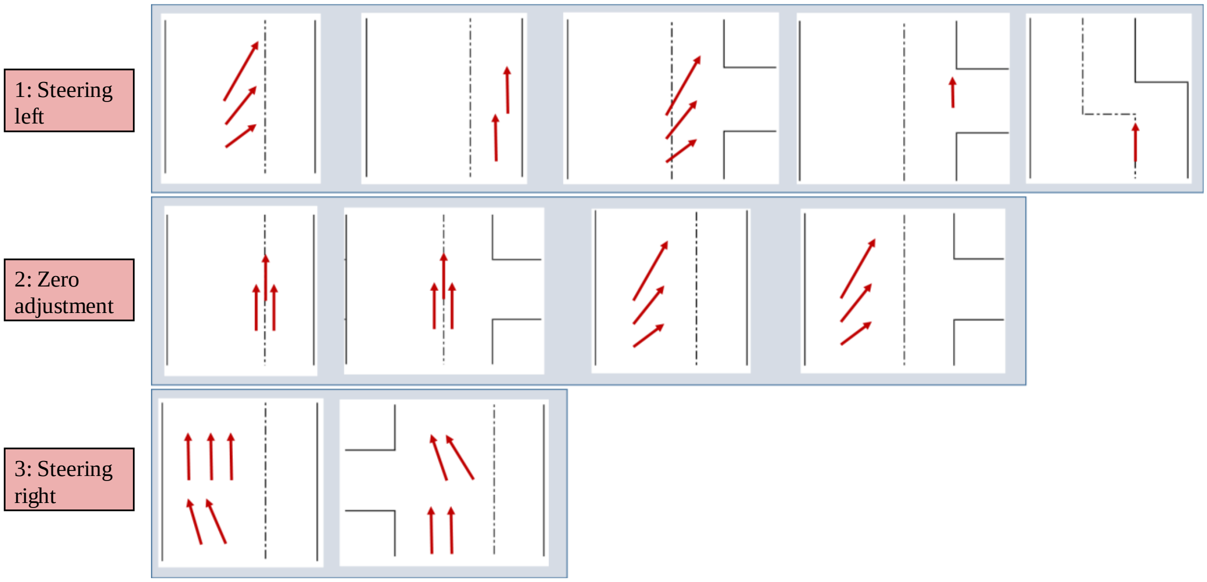

Fig. 3. Illustration of the 11 types of scenarios in the task of Moving Forward with the recommended steering actions. Dash lines represent the implicit target lines for robot movement; arrows represent the robot poses in the corridor environments.

and the magnitude of the handlebar operation. In the course of the AMR movement, the current speed of the robot is also detected with the encoders installed on the wheel axles. The handlebar operation takes into account the speed information, e.g., a swifter steering response for a higher robot speed [5].

III. Results

To verify the performance of the CNN classifier, test images were collected at different locations of the office building from those of the training set. Ground truths of the test images were judged by the human operator based on the 11 scenario types; an overall test accuracy of 94.5% was obtained.

To further verify the proposed auto-steering system, road tests at various locations in the office building were conducted. The images online captured by the depth camera were recorded; the online detection results were compared with the ground truths to calculate the accuracy of the CNN classifier. The results are shown in Fig. 4 in the form of a confusion matrix; the average accuracy is 87.6%.

In addition to the slight improvement in the steering accuracy compared to the previous model [6], great improvements in the versatility of the Moving Forward task were witnessed. For example, the AMR can steadily cross the crossroads and T-junctions as demonstrated in Fig. 5. When the robot starts from a position on the left side of the target line, the auto-steering system efficiently steers the robot back to the target line within 6 to 8 meters and effectively maintains it on the target line. While encountering an obstacle, the success rate of bypassing the obstacle and resuming to the target line was elevated. In general, the AMR can accomplish Move Forward in common corridor environments consisting of much larger varieties of scenarios with smooth movements.

IV. Conclusion

In this article, we proposed a task-oriented auto-steering framework for autonomous navigation of AMRs. In the training phase of a specific task, various visual scenarios are analyzed and summarized into a finite number of scenario types. The recommended steering action in response to each scenario type is determined; images reflecting different scenarios are collected and annotated with the steering actions to form the training set of a CNN classifier that provides steering commands for the AMRs.

Significant improvements in the performance of the auto-steering framework compared to our previous version were witnessed based on the road test results. As the training set includes more diversified scenario types, the robot can steadily cross the crossroads and T-junctions and effectively bypass an obstacle and resume to the target line. When the robot is placed off the target line and orientation, the auto-steering system efficiently maneuvers the robot back to the implicit target line within 6 to 8 meters and maintains it on the line with smooth movements.

To further enhance the CNN detector, it is recommended that the training set is extended to include visual features that have not been included such as low fences, equipment, or furniture used to define the hallways. In the future, it is also planned to extend the current achievements to corner left-turn

		Predicted 1	Predicted 2	Predicted 3
Actual	1	0.81	0.17	0.02
Actual	2	0.10	0.90	0.00
Actual	3	0.01	0.07	0.92

Fig. 4. Confusion matrix of the road tests.

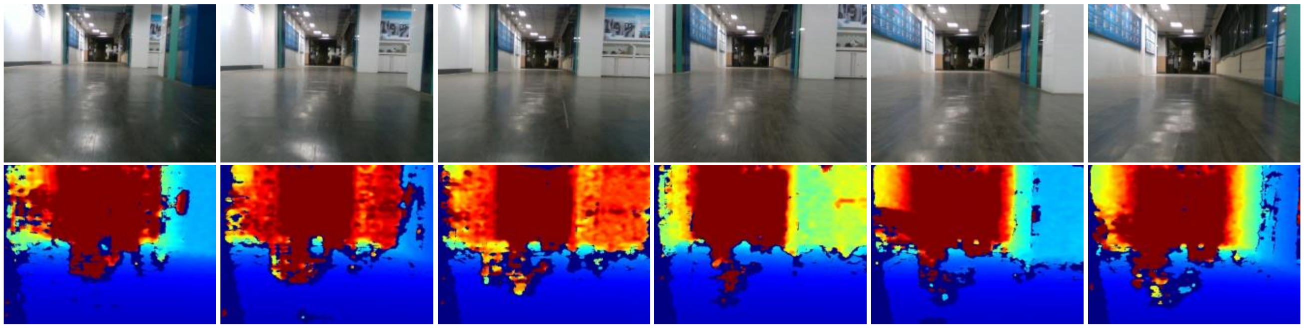

Fig. 5. Sequence images of the road test showing the enhanced performance of the auto-steering framework such as crossing a T-junction.

and corner right-turn to establish an auto-steering system that can complete all the navigation tasks in indoor corridor environments.

ACKNOWLEDGMENT

The support by AXIS-Group under grant no. 209A060 is gratefully noted.

REFERENCES

[1] S. Wang et al., "A localization and navigation method with ORB-SLAM for indoor service mobile robots," in Proc. *IEEE Int. Conf. Real-time Comput. Robot.*, 2016, pp. 443-447.

[2] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 739–746.

[3] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: learning affordance for direct perception in autonomous driving," in *Proc. Int. Conf. Comput. Vision*, 2015, pp. 2722–2730.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Infor. Process. Sys. (NIPS)*, 2012.

[5] C. G. Li, L.-P. Zhou, "Training End-to-End Steering of a Self-Balancing Mobile Robot Based on RGB-D Image and Deep ConvNet," in *Proc. 2020 IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM 2020)*, 2020, pp. 898-903.

[6] C. G. Li, L.-P. Zhou, and Y.-H. Chao, "Self-balancing two-wheeled robot featuring intelligent end-to-end deep visual-steering," *IEEE/ASME Trans. Mechatronics*, early access, 2020. https://ieeexplore.ieee.org/document/9250580.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, 2016, pp. 770-778