Human-Computer Interaction & Emerging Technologies, Vol. 195, 2025, 314–323
https://doi.org/10.54941/ahfe1006250

Robot Autonomy Through Learning From Multi-Camera Images and Human Selection Behavior

Manabu Motegi

Takushoku University, 815–1 Tatemachi, Hachioji-shi, Tokyo 193-0985, Japan

ABSTRACT

The COVID-19 pandemic, which began in early 2020, highlighted the need for technologies that can mitigate the risks of human exposure during infectious disease outbreaks. Given the ongoing threat of emerging pandemics, it is crucial to develop robotic systems that can be remotely operated by humans and eventually achieve autonomous behavior through learning from such interactions. As a fundamental study in this direction, this paper presents a method for enabling robots to autonomously behave in environments. The proposed system integrates real-time and past images from multiple cameras and learns human selection behavior based on these images to enable autonomous decision-making. Experimental results demonstrate that the proposed system achieves significantly longer autonomous behaviour without collisions compared to the author's previous approach.

Keywords: Machine learning, Multi-camera images, Autonomous behavior

INTRODUCTION

We developed a system for autonomous robot behavior that learns from current and one-step previous camera images together with human selection behaviour.

The rapid global spread of COVID-19 since the end of January 2020 remains vivid in our memory. The World Health Organization (WHO) officially declared the end of the public health emergency, which had lasted for three years and three months, on May 5, 2023. However, COVID-19 infections have not been completely eradicated, and the virus continues to mutate. In today's globalized society, the possibility of future pandemics, including those caused by other infectious diseases, cannot be ruled out. During the spread of COVID-19, remote work supported by teleconferencing systems became part of the so-called "new normal." However, in domains that require interaction with the physical world, such as caregiving, logistics, and activities involving travel, teleconferencing systems alone are insufficient to address the associated challenges. Therefore, in such domains, it is expected that embodied robots can be utilized as avatars. By remotely operating these robots, it becomes possible to avoid direct human-to-human contact while enabling more productive activities. However, if robots lack autonomy and require constant remote operation, this imposes

a significant burden on human operators. Moreover, enabling robots to exhibit fully autonomous behavior from the outset is technically challenging, especially in unfamiliar environments with high levels of uncertainty, such as offices and homes. In recent years, extensive research has been conducted on autonomous navigation for robots, drones, and vehicles. However, many of these approaches rely not only on camera images but also on various additional sensors such as infrared, radar, and ultrasonic sensors, which increases system cost. In addition, the algorithms for autonomous navigation also tend to become complex (Nieuwenhuisen, 2014). In addition, conventional image-based navigation systems typically required a multi-stage process. For example, a common approach involved first extracting features from camera images (Vale, 2004), then constructing a map based on these extracted features (Jeong, 2006), and finally determining the robot's actions according to predefined rules (Belker, 2002) (Kim, 2018). However, in such multi-stage processes, each stage must be recalibrated when the environment changes. Moreover, errors can occur at each stage, and these errors tend to accumulate throughout the pipeline. Therefore, to address these issues, several studies have explored the use of deep learning as an end-to-end approach, where control outputs are generated directly from camera images without intermediate steps (Kim, 2018) (Liu, 2017).

In this study, we organized the research objectives and conditions as a fundamental investigation into the aforementioned issues, as follows:

- A bipedal robot is utilized with a view toward future realization of avatar systems.
- Since the target environment is assumed to be a daily living space, only passive sensors, similar to those used by humans, are employed. In particular, this study focuses on utilizing multiple image inputs.
- The study targets autonomous walking, which is the most fundamental form of autonomous behavior.
- The objective is to enable the robot to continuously walk autonomously over a specified section without colliding with obstacles in the environment.

Based on the above considerations, this study aims to develop a system in which a bipedal robot can autonomously walk without colliding with obstacles, using sensor data collected during human teleoperation as a first step.

SYSTEM REQUIREMENT

In this chapter, we organize the system requirements.

First, when utilizing robot camera images in conjunction with human operation commands, it is necessary to understand how and in what situations the human operator controls the robot. Furthermore, in such cases, it is also important to take into account the operator's past control history. In other words, it is essential to acquire not only the camera images and the corresponding human operation commands as paired data, but also to include the operator's control history in the data format.

In this study, the small bipedal robot equipped with two cameras is utilized. We consider the use of both cameras and incorporate images from both into the training data.

Furthermore, it is desirable for the robot to be able to move autonomously without colliding with obstacles in its environment.

Accordingly, the system requirements for this study were organized as follows:

1. During human operation, it must be possible to acquire the two camera images and the robot control commands (log acquisition).
2. To consider historical information, past and current camera images should be combined into a single image, and the corresponding robot control commands must be obtained (utilization of camera image history).

This paper focuses particularly on investigating the above requirements. Furthermore, the following requirement was evaluated:

3. During autonomous behavior, the robot must be capable of behaving without colliding with the environment (autonomous behavior decision).

SYSTEM IMPLEMENTATION AND EXPERIMENTATION

Figure 1 shows the system configuration, the robot used in the experiments, and the GUI used by the user to operate the robot during the data acquisition phase.

Figure 2 illustrates the experimental environment. Figure 3 presents the method used to integrate the images from the two cameras. Figure 4 depicts the learning architecture where a Support Vector Machine (SVM) is added to the CNN. Finally, Table 1 presents an example of the confusion matrix obtained when the SVM was trained as shown in Figure 4.

Implementation Related to Requirement (1) (Log Acquisition)

First, we describe the log acquisition component corresponding to Requirement (1) within the developed system.

As shown in Figure 1(a), the robot used in the experiments was NAO6 (manufactured by SoftBank Robotics). Primitive actions were predefined for the robot, including forward, right turn, left turn, and backward behavior. The parameters for these were set such that forward and backward behaviors each covered 10 cm, and right and left turns each rotated the robot by 10 degrees. NAO6 is equipped with two cameras: one mounted on its forehead and another at the mouth area. For sending control commands to the robot and executing the training process, a notebook PC (FUJITSU LIFEBOOK WA3/D3, CPU: Intel Core i7-9750H, Memory: 32 GB) was used, connected to the NAO6 via a Wi-Fi access point. The system was developed using the Python programming language.

Additionally, as shown in Figure 1(b), a GUI was developed. The camera mounted on the robot's forehead is referred to as Camera 1, while the camera

at the mouth area is referred to as Camera 2. In the experiments, the human operator initially controlled the robot using this GUI.

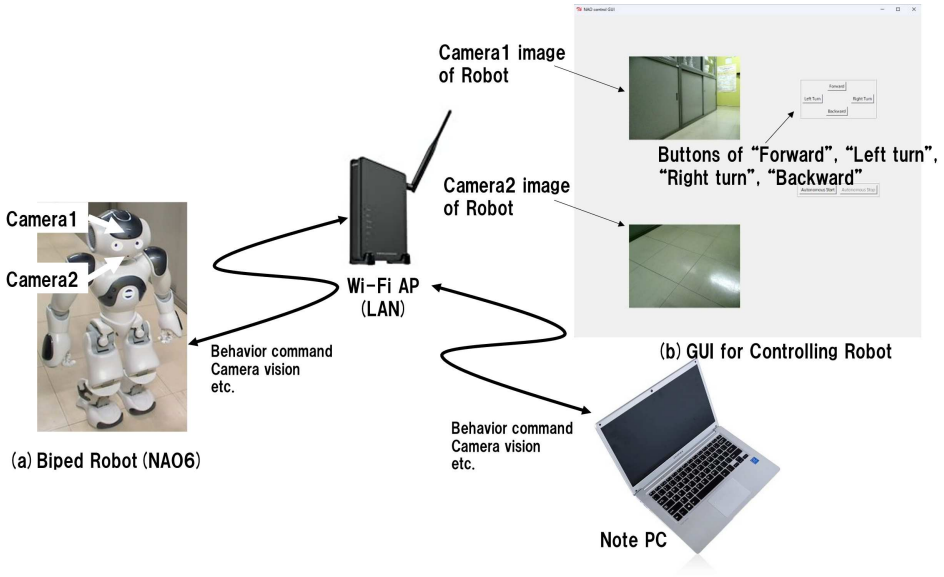


Figure 1: System architecture: Robot (NAO6) and GUI used in the experiment.

Implementation Related to Requirement (2) (Use of Camera Image History)

Next, we describe the implementation of the camera image history utilization corresponding to Requirement (2).

The experimental environment, as shown in Figure 2, was the authors' university laboratory. In this environment, the robot was operated using the GUI shown in Figure 1(b).

While the human operator controlled the robot, the system simultaneously generated training data images as illustrated in Figure 3. The method for creating these images is detailed below:

(1) First, when the operator clicks a control button on the GUI in Figure 1(b), the system captures the two images from Camera 1 and Camera 2 prior to the robot's action. These two images are combined side by side into a single wide-format image file and temporarily saved.

(2) Similarly, when the operator clicks a control button for the next step, the system captures and saves another side-by-side image from Camera 1 and Camera 2.

At this point, the image obtained in step (1) is referred to as the "previous step image," while the image obtained in step (2) is referred to as the "current image."

Then, the "previous step image" is concatenated below the "current image," creating a single image that integrates the four images: the current and previous images from both Camera 1 and Camera 2.

This composite image has a resolution of 640×480 pixels and is hereafter referred to as the "4-segment image."

This 4-segment image, along with the robot behavior selected by the human operator at the time the "current image" was captured, is saved as a set of training data.

As described above, the behavior selected during the operation and the corresponding 4-segment image created as shown in Figure 3 were saved as training data

Figure 2: Experimental environment.

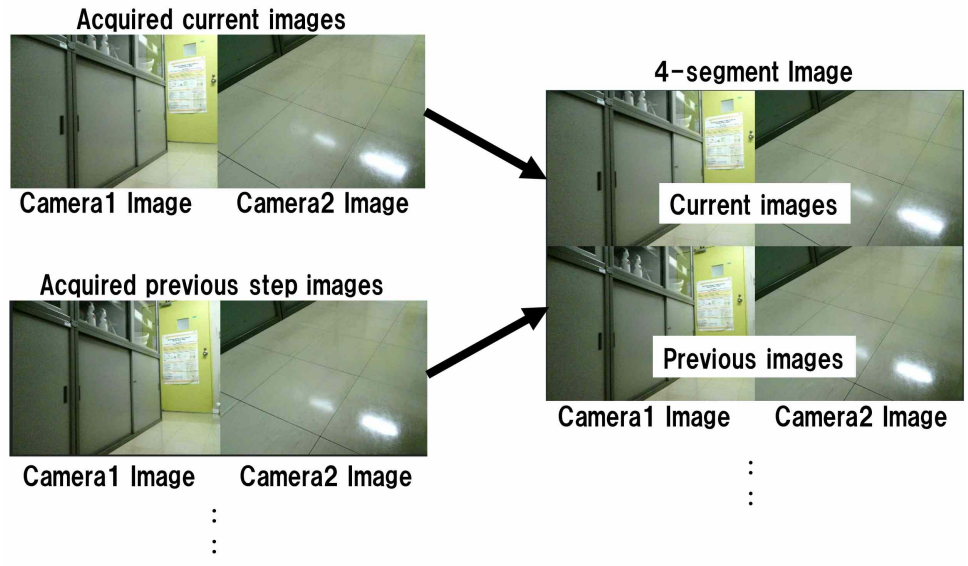


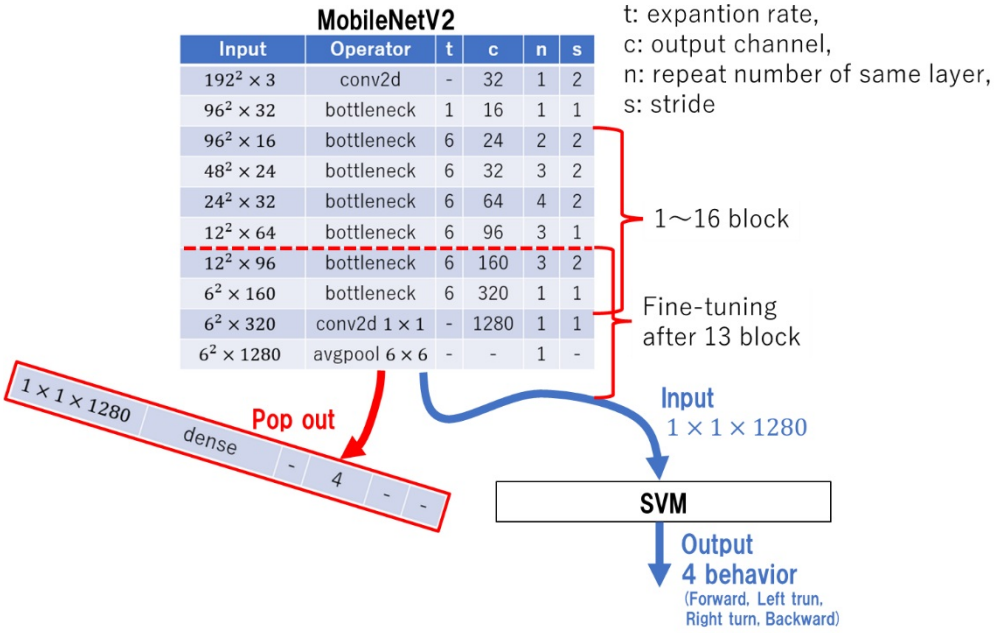
Figure 3: 4-segment image creation methods.

Regarding Training Data and System Configuration

For the training data, the operator manually controlled the robot to move back and forth seven times between the area in front of the location shown in Figure 2 (a) and the refrigerator shown in Figure 2(b), collecting the 4-segment images described above.

During the data collection process, the operator observed both the robot's camera images and its physical position to guide the operation. The robot always started from nearly the same position and orientation near the front of Figure 2(a), and turned right to reverse direction upon reaching the refrigerator in Figure 2(b).

During autonomous behavior, the system also generated 4-segment images using the method shown in Figure 3, and input them into the pre-trained system to determine the robot's behaviors.



Input	Operator	t	c	n	s
$192^2 \times 3$	conv2d	-	32	1	2
$96^2 \times 32$	bottleneck	1	16	1	1
$96^2 \times 16$	bottleneck	6	24	2	2
$48^2 \times 24$	bottleneck	6	32	3	2
$24^2 \times 32$	bottleneck	6	64	4	2
$12^2 \times 64$	bottleneck	6	96	3	1
$12^2 \times 96$	bottleneck	6	160	3	2
$6^2 \times 160$	bottleneck	6	320	1	1
$6^2 \times 320$	conv2d 1 × 1	-	1280	1	1
$6^2 \times 1280$	avgpool 6 × 6	-	-	1	-

Figure 4: Structure of the training part with SVM added to convolutional neural network.

However, using only the initial training data, the robot occasionally collided with obstacles during autonomous behavior. Therefore, when a collision occurred, the corresponding 4-segment images were analyzed and training data were manually refined.

As a result, the final dataset comprised 1,107 images: 34 for backward behavior, 490 for forward behavior, 366 for left turns, and 217 for right turns.

Approximately 80% of the total data were used for training, with the remaining 20% reserved for evaluation.

Using the training data described above, fine-tuning was performed on the pre-trained convolutional neural network, with the 4-segment images as inputs and the corresponding human-selected actions as outputs.

In this study, the convolutional neural network used for fine-tuning was MobileNetV2 (Sandler, 2018), which had been pre-trained on ImageNet (Deng, 2009), a dataset for image recognition containing 1,000 classes.

In this experiment, we fine-tuned MobileNetV2 from block 13 onward, as shown in Figure 4, using the aforementioned training data resized to 192 × 192 pixels (Motegi, 2023). As shown in Figure 4, after fine-tuning, the fully connected layers used for behavior selection were removed from the fine-tuned MobileNetV2, and the output from the average pooling layer was used as the input to an SVM. The SVM was configured to output one of the four behaviors: forward, right turn, left turn, or backward. For the SVM implementation, the scikit-learn library in Python was utilized. Furthermore, the 4-segment images created as training data were input into the system, and the SVM parameters were optimized using the GridSearchCV function.

Table 1: Confusion matrix of SVM output for evaluation data.

		Predicted Behaviors			
		Backward	**Forward**	**Left Turn**	**Right Turn**
Correct behaviors	Backward	65.4%	0.0%	0.0%	34.6%
	Forward	0.0%	95.2%	4.1%	0.8%
	Left turn	0.3%	16.5%	83.2%	0.0%
	Right turn	1.6%	33.7%	7.1%	57.6%

The confusion matrix of the SVM obtained using the evaluation data is shown in Table 1. In Table 1, the rows indicate the ground truth data, and the columns represent the predicted results. The prediction accuracy for the backward behavior was approximately 65.4%, with 34.6% of the cases being misclassified as right turn. This misclassification is considered to have occurred because the human operator often chose to perform backward behavior even in situations where a right turn could have sufficed for obstacle avoidance.

In contrast, the forward behavior showed a high prediction accuracy of 95.2%. This is likely due to the fact that, the images taken when the operator selected the forward behavior generally showed an open space ahead compared to the images associated with other behaviors, making them relatively easier to distinguish.

For the left turn behavior, the prediction accuracy was 83.2%, with 16.5% of the cases being misclassified as forward. This may have been caused by the presence of similar images in the training data where, in some cases, the operator chose to make an early left turn, while in others, the robot continued to move forward under similar visual conditions.

Regarding the right turn behavior, the prediction accuracy was 57.6%, with 33.7% of the cases being misclassified as forward and 1.6% as backward. This is likely because, in the experimental environment, obstacles

were frequently located on the robot's left side, leading the operator to often preemptively execute a right turn even when forward behavior was possible. Additionally, the operator sometimes used a combination of backward and right turn maneuvers to avoid obstacles, which contributed to the misclassification.

SYSTEM EVALUATION

Evaluation on Requirement (3) (Autonomous Operation Decision)

Figure 5 shows photographs taken during the robot's autonomous behavior experiments. Figure 6 indicates the shooting positions where the 4-segment images were utilized, as depicted in Figure 5.

To evaluate the previously described requirement (3), the system was trained using the 4-segment images shown in Figure 3, and subsequently tested for autonomous behavior. The autonomous behavior was compared under the following conditions, continuing until the robot either collided with the environment, stopped due to battery depletion, or encountered other issues:

(1) The system was trained using only the single camera image from the camera mounted on the forehead of the NAO6 robot, as in our previous study (Motegi, 2023).

(2) The system was trained using the 4-segment images shown in Figure 3.

It should be noted that the image set used for condition (1) corresponds to the upper-left quadrant of the 4-segment image shown in Figure 3, which is the forehead camera image.

Figure 5 presents the experimental results of the robot's autonomous behavior under conditions (1) and (2). In each of the two columns, the left column shows the camera images used by the robot when making action decisions during autonomous behavior, while the right column shows the images captured by an external camera for recording purposes. In both conditions (1) and (2), the robot began autonomous behavior from approximately the same position in front of the area shown in Figure 2(a).

In condition (1), as shown in Figure 5, the robot autonomously behaved to the refrigerator from (1–1) to (1–4). Nevertheless, as shown in (1–5), it continued to behave forward in front of the refrigerator and collided head-on. Consequently, the experiment was terminated approximately 249 seconds after the start of autonomous behavior.

On the other hand, in condition (2), as shown in (2–2) of Figure 5 and Figure 6, the robot turned right in front of the refrigerator and behaved to the side of the table. It was then able to return to the vicinity of the starting point of autonomous behavior, as shown from (2–2) to (2–3). Subsequently, it turned left at (2–3) and moved toward the front of the refrigerator as shown in (2–4). Furthermore, in the area near the refrigerator shown in (2–4), it turned right and behaved to (2–5). The robot eventually stopped near (2–5) due to battery depletion. As a result, the experiment was terminated approximately 1380 seconds after the start of autonomous behavior.

Figure 5: Picture of autonomous behaving.

Figure 6: Location of autonomous behavior experiment images.

From the above, it was confirmed that, compared to condition (1), the learning based on the 4-segment images in condition (2) enabled the robot

to perform long-duration autonomous behavior without colliding with the environment.

CONCLUSION

As a fundamental study for realizing avatars in the real world, we examined a system in which the robot autonomously behaves by learning the selected behaviors and the 4-segment images shown in Figure 3, which are obtained during human operation. This system was compared with a system trained using only single-camera images in terms of the duration of autonomous behavior before the robot collided with the environment.

In the case of the system trained with single-camera images, the robot collided with the refrigerator approximately 249 seconds after starting autonomous behavior. However, as proposed in this study, by utilizing the 4-segment images that incorporate the history of images from two cameras, the robot was able to complete approximately two round trips in the experimental environment over about 1380 seconds. Thus, it was confirmed that the proposed learning method enabled longer-duration autonomous behavior without collisions with the environment, compared to the single-camera image-based system.

REFERENCES

A. Vale, J. M. (2004). Feature extraction and selection for mobile robot navigation in unstructured environments. *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, 37*(8), 102–107.

C. Liu, B. Z. (2017). CNN-Based Vision Model for Obstacle Avoidance of Mobile Robot. *MATEC Web of Conferences*.

J. Deng, W. D.-J.-F. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2–9.

M. Sandler, A. H.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520.

M. Motegi. (2023). Autonomous Behavior of Biped Robot by Learning Camera Images.

M. Nieuwenhuisen, D. M. (2014). Obstacle Detectionand Navigation Planning for Autonomous Micro Aerial Vehicles. *International Conference on Unmanned Aircraft Systems*, 1040–1047.

T. Belker, D. S. (2002). Local Action Planning for Mobile Robot Collision Avoidance. *Proceedings of the IEEE/RSJ Inti. Conference on Intelligent Robots and Systems*, 601–606.

W. Y. Jeong, K. L. (2006). Visual SLAM with Line and Corner Features. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2570–2575.

Y-H. Kim, J.-I. J. (2018). End-to-End Deep Learning for Autonomous Navigation of Mobile Robot. *IEEE International Conference on Consumer Electronics*.