A Verification Framework for Physical AI Systems Based on LLM-Based Environment Recognition Using AR Images

Seiya Takazumi[1*] and Masanobu Koga[2]

[1]Department of Creative Informatics, Kyushu Institute of Technology
[2]Department of Intelligent and Control Systems, Kyushu Institute of Technology, Japan
(takazumi.seiya866@mail.kyutech.jp, koga@ics.kyutech.ac.jp) *Corresponding author

Abstract: This study proposes an AR-AI verification framework that enables physical AI systems to be verified in an AR environment. In the proposed framework, an LLM recognizes the positions and poses of target objects and a robot from AR images. The use of AR technology enables operation verification in a real environment without using an actual robot. In this study, a coordinate estimation method that draws grid lines on AR images to assist the spatial reasoning of the LLM is proposed. The effects of the presence or absence of grid lines, grid-line colors, capture conditions, target objects, and capture environments on the coordinate estimation accuracy of the LLM were evaluated. The experimental results showed that the relative error in straight-line distance was reduced from 56.7% to 8.3% by displaying grid lines. In addition, the difference in the relative error of straight-line distance due to the grid-line color was small. On the other hand, under capture conditions in which the grid lines appeared slanted, the relative error increased to a maximum of 92.8%. Furthermore, even for different target objects and capture environments, a tendency was observed for the relative error of straight-line distance to be smaller when grid lines were displayed.

Keywords: LLM, Image Recognition, AR, Physical AI

1. INTRODUCTION

In recent years, research and development on Physical AI[1], which has a physical body and acts in the real world, such as autonomous vehicles and robots operating in factories, has been actively conducted. As shown in Fig. 1, Physical AI understands the real world using information obtained from sensors and plans and executes actions based on the recognition results. However, verifying Physical AI by connecting it to automobiles, robots, and other systems in their actual operating environments involves challenges in terms of cost and safety. When verification is conducted using a virtual environment (VR), the workload required to construct a virtual environment that matches the real environment is not negligible. If a simplified virtual environment is constructed in consideration of this workload, the discrepancy from the real environment becomes large, reducing the items that can be verified in advance.

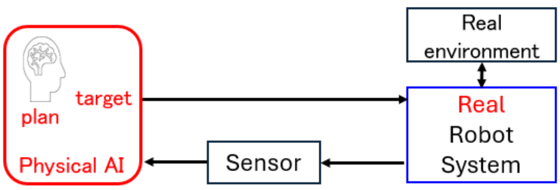

Fig. 1. Overview of Physical AI System

Therefore, this study proposes an AR-AI verification framework that enables verification of Physical AI in an AR environment. An overview of the framework is shown in Fig. 2. In a typical Physical AI system, the positions and orientations of target objects and robots are measured or estimated using sensors such as cameras. In contrast, in the proposed framework, AI recognizes the positions and orientations of the target object and the robot from AR images in which a virtual robot is displayed against the background of a real environment including the target object. By using AR technology, a virtual robot can be superimposed on the real environment, enabling operation verification in the real environment without using actual hardware. This paper focuses on environment recognition using AR images by AI, and employs a multimodal LLM as the AI.

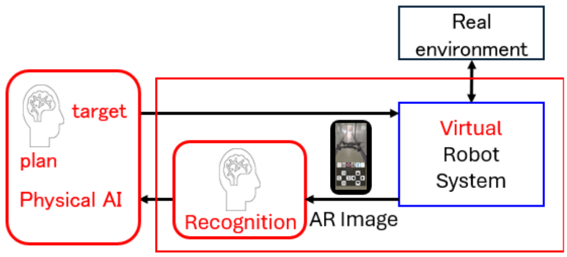

Fig. 2. Overview of the AR-AI Verification Framework

The spatial reasoning capability of LLMs has limitations[2]. Although methods that perform object detection using bounding boxes[4] can recognize objects, they cannot determine the coordinates of a target. In this study, we propose a coordinate estimation method that draws grid lines on AR images to support the spatial reasoning of an LLM.

The structure of this paper is as follows. Chapter 2 describes the evaluation of distance measurement accuracy in AR space. Chapter 3 explains the proposed method, including the method for acquiring AR images and the prompt design provided to the LLM for target coordinate estimation. Chapter 4 reports experiments and evaluations on the effects of grid line color and capture conditions on estimation accuracy. Finally, Chapter 5 presents the conclusions of this study and future prospects.

2. EVALUATION OF MEASUREMENT ACCURACY IN AR SPACE

In this study, because a method is used in which an LLM estimates the positions of target objects and robots from AR images, the measurement values in AR space are compared with the measured values in the real environment to quantitatively confirm the accuracy of measurement values in AR space.

2.1 Experimental Environment

Table 1 shows the devices and software used in the experiment. The AR function of Babylon.js used in the experiment is based, on Android devices, on device position and pose estimation and plane detection by Visual-Inertial SLAM using Google ARCore[5][6]. Therefore, the measurement accuracy in AR space evaluated in this experiment depends on the tracking accuracy of ARCore.

Table 1. Experimental Environment

Item	Specification
Device	Google Pixel 7
OS	Android 16
Browser	Google Chrome 148
ARCore	1.54
Babylon.js	8.31.0

2.2 Experimental Method

Fig. 3. Camera Image

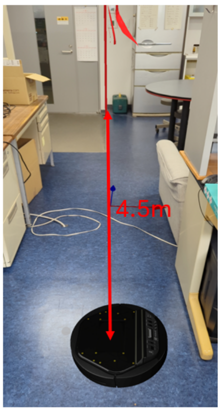

Fig. 4. AR Image

An example of a camera image of the real environment is shown in Fig. 3, and an AR image in which a virtual robot and a virtual target object are composited is shown in Fig. 4. The virtual target object is a flag mesh, and a mobile robot[7] was used as the virtual robot.

In the coordinate system of the AR space used in this experiment, the x-axis and z-axis are two axes on the horizontal plane defined based on the orientation of the device at the start of the AR session, and the y-axis corresponds to the vertically upward direction. In the experiment, a virtual robot and a virtual target object are placed at marks installed at two points separated by 4.50 [m] in the real environment, and their respective coordinates are obtained. Then, the distance between the two points in AR space is calculated from the obtained coordinates, and the error from the measured value is evaluated.

Measurements were conducted five times as independent AR sessions. By starting a new AR session each time, errors including variations in the coordinate system at session initialization were evaluated.

2.3 Experimental Results

The coordinates of the robot and the target object, and the measured distance in AR space for each trial are shown in Table 2.

Table 2. Results of distance measurement

Robot(X,Z)	Target(X,Z)	Distance [m]	Absolute error [m]
(0.13, 0.59)	(0.13, 5.06)	4.47	0.03
(-0.01, 0.46)	(-0.38, 4.93)	4.48	0.02
(0.27, -0.25)	(-3.55, 2.00)	4.43	0.07
(0.01, 0.67)	(-0.31, 5.15)	4.49	0.01
(0.05, 0.61)	(-0.22, 5.08)	4.47	0.03

From Table 2, for the actual distance of 4.50 [m] in the real environment, the mean absolute error was 0.032 [m] and the relative error was 0.71%. Based on this result, the distance measurements in AR space generated by the AR function of Babylon.js were judged to be usable for verification of tasks such as moving a mobile robot close to a target point or moving the end effector of a robot arm close to an object.

3. PROPOSED METHOD

In this study, we propose a coordinate estimation method that assists the spatial reasoning of the LLM by rendering grid lines on AR images. An overview is shown in Fig. 5.

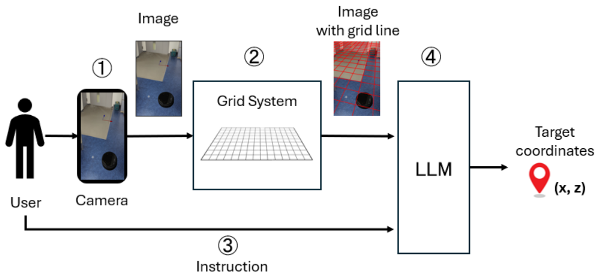

Fig. 5. Overview of the Proposed Method

3.1 Acquisition Method for AR Images

First, camera image frames are acquired as pixel data, and coordinate-axis adjustment processing is performed to generate a base image. Next, 3D models of the robot and the reference coordinate axes are rendered into an internal buffer, and after coordinate-axis adjustment processing, they are synthesized with the base image. As a result, an AR image such as that shown in Fig. 6, in which the virtual robot and reference coordinate axes are synthesized onto the camera image of the real environment, is obtained.

3.2 Rendering of Grid Lines

In this study, grid lines are rendered on the ground plane of the AR space. The grid lines function as a visual scale reference for coordinate estimation, enabling the LLM to estimate the coordinates of the target object in AR space with reference to the robot. An example of an AR image with synthesized grid lines is shown in Fig. 7. By referring to this grid, the LLM estimates the coordinates of the target object in AR space.

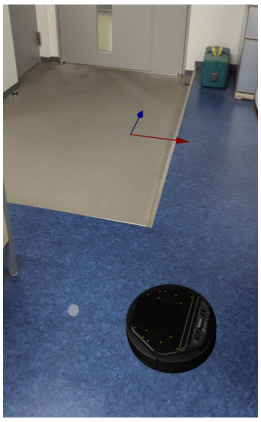

Fig. 6. AR image without grid lines

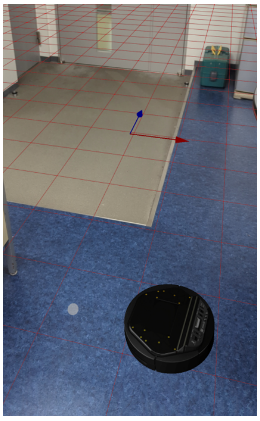

Fig. 7. AR Image with Grid Lines

The design policy for the grid lines is as follows.

1. The grid interval is set to the width of the robot. This allows the LLM to use the robot size as the unit of the grid, and to associate the scale of the image with that of the AR space.
2. Because the LLM may fail to recognize grid positions if auxiliary lines are lighter than main lines, the appearance of the main lines and auxiliary lines is unified.
3. To prevent the grid lines from interfering with recognition of the target object, the opacity of the grid is limited.

The parameters of the grid lines are shown in Table 3.

Table 3. Parameters of grid material

Parameter	Description
gridRatio	line width ratio (default: 1)
majorUnitFrequency	Interval of major grid lines
minorUnitVisibility	Visibility of minor lines
opacity	Opacity

3.3 Prompt Design for the LLM

The input to the LLM consists of an AR image with rendered grid lines and a prompt. The system prompt describes information on the robot in the AR image and how to interpret it, the procedure for estimating the target coordinates, and the criterion for determining whether the target object is located in front of the robot. The LLM is instructed to perform coordinate estimation and frontal alignment determination according to the following procedure.

1. Identify the center of the contact area of the target object in the AR image. For objects such as poles and flags, the reference point is the center of the point where the support touches the ground; for objects such as boxes and obstacles, the reference point is the center of the bottom surface.
2. Determine whether the robot and the target object are located on the same grid line in the depth direction in the AR image.
3. If they are located on the same grid line in the depth direction, output the estimated target coordinates (x_t, z_t). Otherwise, output the movement required for the robot to align with the target object as a natural-language instruction.
4. Measure the number of grid cells, expressed as real values, from the robot center to the target object in the X direction (Δg_x) and in the Z direction (Δg_z), respectively.
5. Convert the number of grid cells into meters.
6. Calculate the target coordinates (x_t, z_t). Here, (x_r, z_r) denotes the current coordinates of the robot, and size denotes the interval per grid cell [m].

$$x_t = x_r + \Delta g_x \times \text{size} \tag{1}$$

$$z_t = z_r + \Delta g_z \times \text{size} \tag{2}$$

3.4 Structured Output

Because the output of a typical LLM is natural-language text, string parsing is required to extract coordinate values from the program, and variations in the output format create a risk of failure in obtaining the values.

Therefore, in this study, we use Structured Output, which makes it possible to obtain LLM output in a format that strictly follows a predefined schema. By using this function, the types and values of the required fields are guaranteed at the API level. The LLM generates output conforming to the above schema according to the AR image and the system prompt, and outputs the estimated target coordinates (x, z) [m].

4. EVALUATION EXPERIMENTS

An experiment is conducted to evaluate the coordinate estimation accuracy of the LLM. The LLM model used in the experiment is GPT-5.4. The output of the LLM is obtained using Structured Output provided by the OpenAI API. At this time, the output schema of the LLM is defined using Pydantic's BaseModel. The fields included in the output schema are shown in Table 4.

Table 4. Schema of LLM Output

Field	Description
description	Explanation of the estimation method
advice	Movement instruction when not facing the target
target_position	Target position (x, z) [m] in world coordinates

All experiments in this chapter are conducted in the same environment as that used for the evaluation of distance measurement accuracy in the Babylon.js AR space described in the previous chapter (Table 1).

4.1 Experimental Method

4.1.1 Experiment With and Without Grid Lines

We verify the effectiveness of grid lines as a visual scale reference in coordinate estimation by the LLM. Ten trials are conducted for each of the cases without grid lines and with grid lines, and the effect of the presence of grid lines on estimation accuracy is evaluated. The camera angle and the position of the target object are fixed. The evaluation is performed using the relative errors of the X and Z coordinates and the straight-line distance.

4.1.2 Experiment on Color of Grid Lines

The image-recognition characteristics of an LLM may differ depending on the grid-line color in coordinate estimation. Therefore, in this experiment, ten trials were conducted under each of four conditions, namely red, blue, green, and black, to investigate the effect of grid-line color on estimation accuracy. The camera angle and target-object position were fixed, and only the grid-line color was changed to evaluate the difference in accuracy caused by the color. The evaluation was performed using the relative errors of the X coordinate, Z coordinate, and straight-line distance.

4.1.3 Experiment on Angle between Grid Line and Camera

Because the coordinate origin of the AR space is placed at the device position at the start of the AR session [8][9], the orientation of the grid lines depends on the posture of the device at the start of the session. Therefore, even in the same real environment, the orientation of the grid lines changes depending on the orientation at the start of the session.

In coordinate estimation using the grid, the appearance of the grid lines changes greatly depending on the camera capture angle, and shape distortion caused by perspective projection is considered to affect the LLM's measurement of the number of grid cells. In addition, when the angle θ between the robot's forward direction and the target direction is not $0°$, the target object deviates from the image center, which is expected to affect estimation accuracy.

Therefore, an experiment is conducted under four conditions combining two factors, the camera angle and the position of the target object, and the effect of capture conditions on estimation accuracy is evaluated. The details of the four conditions are shown in Table 5, and an overview of the positional relationship among the robot, target object, and smartphone is shown in Fig. 8. In the figure, the green arrow indicates the forward direction of the smartphone, and the blue arrow indicates the direction of the robot relative to the target object; the angle between the green line and the blue line is denoted by θ.

Table 5. Condition of grid

Condition	Grid direction	Relative positions
A	Perpendicular	On the same line
B	Perpendicular	Not on the same line
C	Slanted	On the same line
D	Slanted	Not on the same line

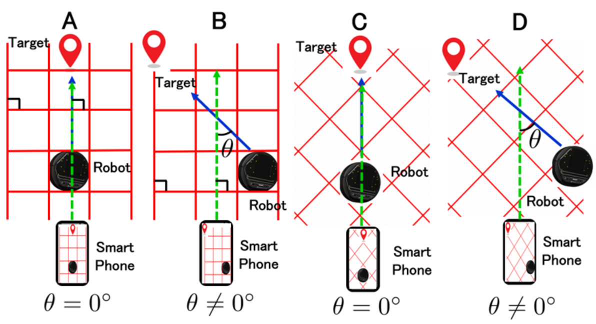

Fig. 8. The positions of smartphone, robot, and target

Under these four conditions, ten trials are conducted for each of the four grid-line colors, red, blue, green, and black. The mean absolute error and relative error calculated for each color are further averaged over the four colors, and the resulting values are used as representative values for each condition. By comparing these values from three viewpoints, namely the X coordinate, Z coordinate, and straight-line distance, the effect of capture conditions on coordinate estimation accuracy is evaluated.

4.1.4 Experiment on Sizes and Shapes of Target Object

This experiment evaluated the effect of differences in target-object size and shape on the coordinate estimation results of the LLM. Five target objects were used, namely a door, refrigerator, flag, chair, and pet bottle, considering differences in size and visibility. For each target object, experiments were conducted under conditions with and without grid lines, and the estimation results were compared. The evaluation was performed using the relative errors of the X coordinate, Z coordinate, and straight-line distance.

4.1.5 Experiment on Working Spaces

This experiment evaluated the effect of differences in space on the coordinate estimation results of the LLM. Three spaces were used, namely a studio apartment for a single occupant, a university laboratory, and a corridor, considering differences in the distribution of surrounding objects and spatial structures. For each space, experiments were conducted under conditions with and without grid lines, and the estimation results were compared. The evaluation was performed using the relative errors of the X coordinate, Z coordinate, and straight-line distance.

4.2 Experimental Results

4.2.1 Experiment With and Without Grid Lines

The experimental results with and without grid lines and the corresponding box plots are shown in Table 6 and Fig. 9, respectively. From Table 6, the relative error of

Table 6. Relative error: With and without grid lines

Grid line	X-axis [%]	Z-axis [%]	Relative error [%]
none	387.5	56.3	56.7
with grid lines	160.0	7.3	8.3

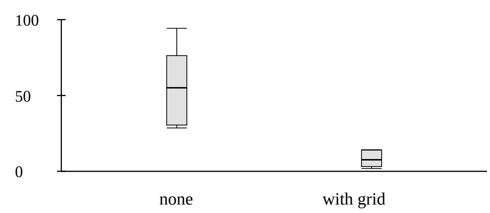

Fig. 9. Relative errors with and without grid

the straight-line distance was reduced by rendering grid lines. The error in the X direction decreased when grid lines were used. The error in the Z direction decreased substantially when grid lines were used. The box plots in Fig. 9 also show that, under the condition with grid lines, the quartiles are contained within a narrow range.

These results confirm that rendering grid lines stabilizes the estimation results and, in particular, improves coordinate estimation accuracy in the Z direction.

4.2.2 Experiment on Color of Grid Line

The results of the experiment conducted while changing the color of the grid lines and the corresponding box plots are shown in Table 7 and Fig. 10, respectively.

Table 7. Relative error for different grid-colors

Color of line	X-axis [%]	Z-axis [%]	Relative error [%]
red	117.5	8.8	9.2
blue	302.5	9.3	11.0
green	72.5	7.5	7.8
black	160.0	7.3	8.3

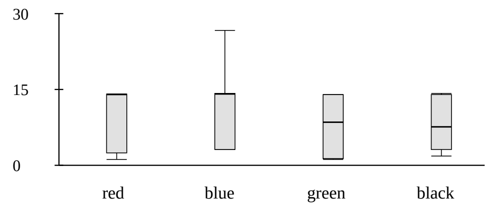

Fig. 10. Relative errors for different grid colors

From Table 7 and Fig. 10, although some differences in the estimation results are observed depending on the color of the grid lines, the effect is limited as long as the relative error of the straight-line distance is used as the metric.

4.2.3 Experiment on Angle between Grid Line and Camera

The results of the experiment conducted under four capture conditions and the corresponding box plots are shown in Table 8 and Fig. 11, respectively. It was confirmed that, under Conditions C and D, in which the grid lines are rendered at a slant, the relative error increases markedly compared with Conditions A and B, in which the grid lines appear orthogonal. In addition, under the conditions in which the grid lines appear orthogonal, the relative errors in the Z direction and the straight-line distance increased from Condition A to Condition B, whereas no similar tendency was observed between Conditions C and D.

Table 8. Error and boxplots by capture and target conditions (averaged over grid colors)

	Absolute error [m]			Relative error [%]		
Condition	X	Z	Distance	X	Z	Distance
A	0.057	0.227	0.246	142.5	8.3	9.0
B	0.225	0.413	0.490	562.5	15.1	17.9
C	1.541	0.996	1.852	138.8	60.0	92.8
D	1.633	0.507	1.734	147.1	30.6	86.8

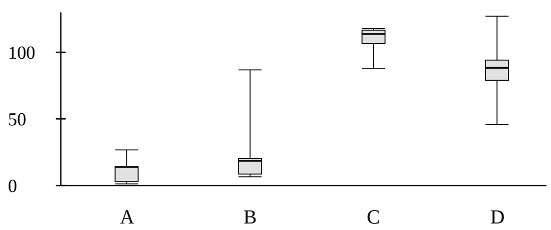

Fig. 11. Relative errors for capture conditions

The box plots in Fig. 11 also showed that differences in capture conditions, particularly whether the grid lines appear orthogonal or slanted, have a large effect on the coordinate estimation accuracy of the LLM.

4.2.4 Experiment on Sizes and Shapes of Target Object

The results of the experiment using different target-object sizes and shapes are shown in Table 9.

Table 9. Relative errors due to size of target objects

	with grid lines[%]			none[%]		
Target	X	Z	Distance	X	Z	Distance
door	228.0	12.6	13.3	108.0	58.9	59.0
refrigerator	180.0	8.2	9.2	167.5	55.0	55.1
flag	134.0	20.0	20.1	106.0	97.6	97.6
chair	103.9	13.4	15.0	391.7	72.7	76.0
pet bottle	84.5	19.0	19.2	30.0	64.2	64.2

The experimental results showed that, under the condition with grid lines, the relative error in straight-line distance was 9.2% for the refrigerator and 13.3% for the door, indicating relatively small values for the comparatively large target objects. In contrast, the corresponding errors for the flag, chair, and pet bottle were 20.1%, 15.0%, and 19.2%, respectively. Therefore, in this experiment, relatively small errors were observed for the comparatively large door and refrigerator; however, no relationship was confirmed in which the error decreased in proportion to the target-object size. In addition, comparison of the conditions with and without grid lines confirmed that the relative error in straight-line distance was smaller with grid lines for every target object.

4.2.5 Experiment on Working Spaces

Table 10. Relative errors due to environmental spaces

	with grid lines[%]			none[%]		
Environment	X	Z	Distance	X	Z	Distance
a studio apartment	88.9	10.6	13.5	27.4	68.3	68.2
university lab	103.9	13.4	15.0	391.7	72.7	76.0
corridor	710.0	21.0	21.3	1900.0	64.7	65.2

The results of the experiment using different environments are shown in Table 10. The experimental results showed that, under the condition with grid lines, the relative error in straight-line distance was smaller in the studio apartment for a single occupant and the university laboratory, where surrounding objects were present, than in the corridor. On the other hand, under the condition without grid lines, the corridor, where surrounding objects were absent, exhibited a smaller error. In addition, comparison of the conditions with and without grid lines confirmed that the relative error in straight-line distance was smaller with grid lines in all environments, namely, the studio apartment for a single occupant, the university laboratory, and the corridor.

4.3 Evaluation and Discussion

The results of this experiment indicated that the coordinate estimation accuracy of the LLM depended on the capture conditions, particularly the appearance of the grid lines. The smallest error under condition A is considered to have resulted from the grid lines appearing perpendicular and the target object being located on the same line as the robot. In contrast, under condition B, the positional offset between the target object and the robot, and under conditions C and D, the inclination of the grid lines, are considered to have made coordinate estimation difficult.

In addition, when the difference between the X coordinates of the target object and the robot is small, even a small absolute error results in a large relative error in the X direction. Therefore, it is appropriate to use the relative error in straight-line distance when evaluating the distance to the target object.

In the experiments using different target objects and environments, the relative error in straight-line distance was smaller with grid lines for all target objects and environments. These results indicate that the camera pose, which determines the appearance of the grid lines, is one factor affecting the coordinate estimation accuracy of the LLM. Furthermore, within the range of target objects and environments evaluated, the effectiveness of the proposed method was confirmed.

5. PATH-FOLLOWING CONTROL USING AR IMAGES BY AI

This chapter introduces an application in which the target coordinate estimation method based on AR images proposed in this study is connected to a control system for a virtual robot. An overview of the system is shown in Fig. 12. In this system, the target coordinates estimated by the LLM from AR images are input as target values to the virtual robot controller system. This system consists of an AR image generation unit, a target coordinate estimation unit using the LLM, and VirtualRobotSystem, which receives the generated target values. In VirtualRobotSystem, a path is generated using the current position of the virtual robot and the target coordinates estimated by the LLM. The actual path-following behavior is shown in Fig. 13, and the generated path and the tracking results of the virtual robot are shown in Fig. 14.

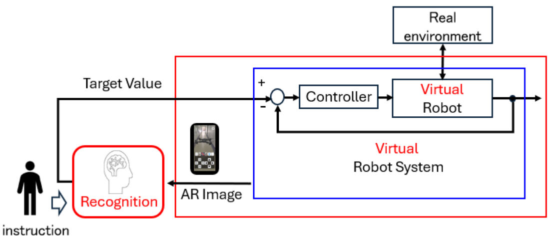

Fig. 12. AI-Based Path-Following control system using AR Images


Fig. 13. Path following

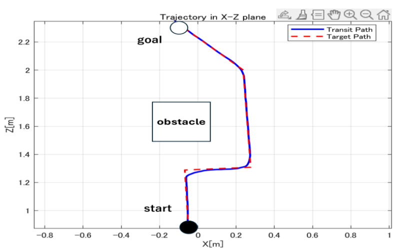

Fig. 14. Results of path following

Figure 13 shows the virtual robot placed in the AR image moving toward the target object based on the generated target values. Figure 14 shows the target path and the trajectory of the virtual robot on the X-Z plane, confirming that the virtual robot follows the path toward the target position while avoiding obstacles. This result confirms that the target coordinates estimated from AR images can be input to VirtualRobotSystem and used for path-following control of the virtual robot.

6. CONCLUSION

This study proposed an AR-AI verification framework that combines AR technology and an LLM to verify the operation of physical AI without using an actual robot. This paper focused on environment recognition using AR images by an LLM and investigated a method for estimating the coordinates of target positions in AR images using an LLM. A method for improving the coordinate estimation accuracy of an LLM through grid-line rendering and prompt design was developed.

The experimental results showed that displaying grid lines reduced the relative error in straight-line distance from 56.7% to 8.3%. On the other hand, the difference in error due to the grid-line color was small. In addition, in experiments using different target objects and environments, a tendency was observed for the relative error in straight-line distance to be smaller when grid lines were displayed under all evaluated conditions.

In the comparison experiment using different capture conditions, the relative error increased to a maximum of 92.8% under conditions in which the grid lines were rendered slanted. In practical use, ensuring a capture pose in which the grid lines appear perpendicular to the forward direction of the smartphone is therefore key to maintaining accuracy. However, in real-world use, users may not always be able to maintain an appropriate camera pose. Accordingly, achieving stable coordinate estimation accuracy regardless of the capture pose remains a subject for future work.

REFERENCES

[1] NVIDIA, “Cosmos World Foundation Model Platform for Physical AI,” arXiv:2501.03575, 2025, accessed May 22, 2026.

[2] Z. Cai et al., “Has GPT-5 Achieved Spatial Intelligence? An Empirical Study,” arXiv preprint arXiv:2508.13142, 2025, accessed May 28, 2026.

[3] OpenAI, “Structured Outputs,” https://developers.openai.com/api/docs/guides/structured-outputs, 2024, accessed May 20, 2026.

[4] H. You, H. Zhang, Z. Gan, X. Du, B. Zhang, Z. Wang, L. Cao, S.-F. Chang, and Y. Yang, “Ferret: Refer and Ground Anything Anywhere at Any Granularity,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2024, accessed June 2, 2026.

[5] Babylon.js Team, “Babylon.js and WebXR,” https://babylonjs.medium.com/babylon-js-and-webxr-or-how-i-learned-to-stop-worrying-and-love-xr-557707748bdb, 2020, accessed May 20, 2026.

[6] Immersive Web Community Group, “Immersive Web Developer Home,” https://immersiveweb.dev/, 2024, accessed May 19, 2026.

[7] adeleuta, “Kobuki.rdf,” Sketchfab, https://sketchfab.com/3d-models/kobukirdftf-09b1892c068f4ad395cbee3dd967d60c, accessed Feb. 6, 2026, CC BY 4.0.

[8] Google for Developers, “Create an immersive AR session using WebXR,” https://developers.google.com/ar/develop/webxr/hello-webxr, 2024, accessed May 20, 2026.

[9] MDN Web Docs, “XRReferenceSpace,” https://developer.mozilla.org/en-US/docs/Web/API/XRReferenceSpace, 2024, accessed May 20, 2026.