

Augmented Reality Display of Robot with Graphs of Property Response Using Its USD Model

Kazuki Tsukamoto^{1*} and Masanobu Koga²

¹ Department of Creative Informatics, Kyushu Institute of Technology,
(tsukamoto.kazuki676@mail.kyutech.jp) * Corresponding author

² Department of Intelligent and Control Systems, Faculty of Computer Science and Systems Engineering,
Kyushu Institute of Technology, (koga@ics.kyutech.ac.jp)

Abstract: This study proposes a method that can easily grasp the relationship between the actual machine and the graphs. In recent years, there has been a lot of research on augmented reality displays. The fields of research range from education to welfare. In the development of control systems, when evaluating the performance of a system by simulation or experiment, the results are often checked as graphs. Since the graphs are checked on a PC using CAD or other means, it is difficult to know which part of the actual machine each graph corresponds to. Therefore, we developed a tool that displays graphs in augmented reality around the actual machine through a camera on a mobile terminal. To display graphs in augmented reality, it is important to obtain the coordinates of the actual machine and display them in a location associated with the device. Therefore, a USD model with the same shape and size as the actual machine is used. This is achieved by displaying the USD model in augmented reality so that it is superimposed on the actual machine. The accuracy of the tool was also examined and its usefulness was evaluated.

Keywords: Augmented Reality, USD model

1. INTRODUCTION

In recent years, there has been a lot of research on augmented reality displays. Augmented reality, such as Pokémon Go and other AR applications, is now becoming a familiar part of our lives. Not only that, augmented reality is being used in a variety of fields. In the field of education, various systems have been developed for learning purposes and their usefulness has been discussed [1]. In the field of social welfare, automatic driving of welfare vehicles and obstacle avoidance systems based on spatial mapping have been proposed [2].

In the development of control systems, when evaluating system performance through simulation and experimentation, the results are often viewed as graphs. Some graphs have similar shapes, which may cause users to misidentify them. In addition, some graphs have labels, but without knowledge of the actual machine, the user may not understand which part of the device the graph refers to from the labels. Thus, when checking experimental results on graphs, it is difficult to understand which part of the actual machine each graph corresponds to.

In this study, we develop an augmented reality display tool that displays graphs of experimental results in AR around the actual machine through a mobile device camera, making it easier to understand the relationship between the graphs.

The structure of this paper is as follows. First, we propose a method for displaying graphs in augmented reality in Chapter 2. Next, in Chapter 3, we explain the development of the tool based on our proposal. In Chapter 4, we evaluate the usefulness of the developed tool.

2. AR ROBOT WITH GRAPHS

Research on augmented reality display of graphs has been conducted in the past, and there are applications that convert spreadsheets into 3D graphs in real space and superimpose them[3]. The tool developed in this study for augmented reality display of graphs is to superimpose a virtual graph around an actual machine in real space, as shown in Figure 1.

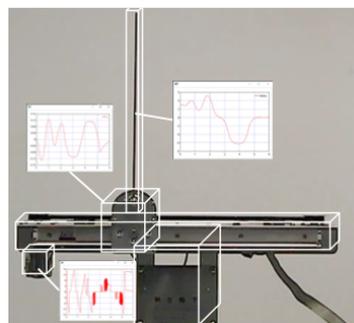


Fig. 1. Image of the tool.

Until now, graphs in the development of control systems have been checked on a PC screen using CAD or other tools. However, using the proposed method, it is possible to add information as a graph to the actual machine to be controlled in real space through the camera of a mobile terminal, enabling intuitive understanding.

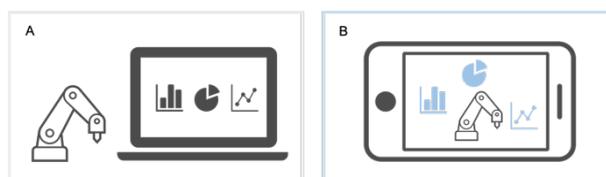


Fig. 2. Conventional Method(A) and Proposed Method(B).

2.1 Relating graphs to real space

It is difficult to directly connect a virtual graph to an actual machine in the real world. We need the spatial coordinate information of the actual machine as a basis for superimposing the graph. If the object to be connected to the graph is not in the real space but in the virtual space, it is not so difficult to obtain the coordinate information. To solve this problem, a 3D model of the same shape and size as the actual machine is created in virtual space, and a graph is attached to it. If the 3D model with the graph attached can be superimposed on the actual machine in real space, the graph associated with the actual machine can be displayed in augmented reality.

Superimposing the 3D model on the actual machine means that the six pieces of information on the actual machine and the 3D model match: the 3D orthogonal coordinate system (x, y, z) and the rotation around the coordinate axes (α, β, γ). In order to match these information, we detect planes by spatial recognition and move the 3D model by mobile device operation.

2.2 Plane detection by spatial recognition

Plane detection is a function that acquires plane information if there is a plane in the real space when the camera recognizes the real space. Using this function, it is possible to acquire information on the xz plane where the actual object is located. By superimposing the 3D model on the real space based on this planar information, the y -coordinates, α -axis rotation, and γ -axis rotation of the actual machine and the 3D model can be matched.

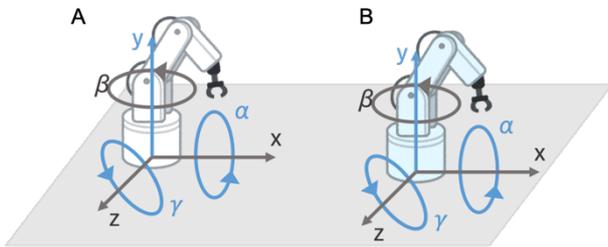


Fig. 3. Planar match between actual machine(A) and vertical 3D model(B).

2.3 Moving 3D models by mobile device operation

The xz -plane is aligned by plane detection, so the 3D model is moved on the plane and superimposed on the actual machine. We propose the method shown in Figure 4 as a way to move the 3D model.

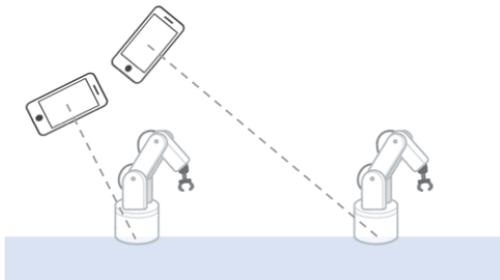


Fig. 4. Plane translation of 3D model.

In order to move the 3D model using intuitive operations such as tilting the mobile terminal, the center coordinates of the camera are acquired. The information of the camera's center coordinates (x, z coordinates) is reflected in the x, z coordinates of the 3D model. In this way, the 3D model is always superimposed on the center of the camera, and when the smartphone is tilted, the 3D model moves as if it were sliding on a flat surface. By superimposing the 3D model on the actual machine using this smartphone operation, the x, z coordinates of the actual machine and the 3D model can be made to match.

Since the 3D model retains the β -axis rotation information at the time of generation, the orientation of the 3D model does not change even if the mobile terminal is moved as shown in Figure 5.

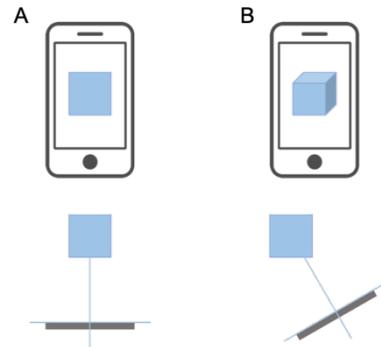


Fig. 5. Obtaining the center coordinates of the camera. (A) Viewed from the front. (B) Viewed from an angle.

Since the 3D model retains the β -axis rotation information at the time of generation, the orientation of the 3D model does not change even if the mobile terminal is moved. Since we want to rotate the β -axis of the 3D model according to the tilt of the mobile terminal, the β -axis rotation information of the mobile terminal is reflected in the 3D model to unify the orientation of the 3D model and the camera. This unification ensures that the front of the 3D model always faces the direction of the camera in accordance with the movement of the mobile terminal. These plane detection and mobile terminal operations enable superimposition of the 3D model on the actual machine.

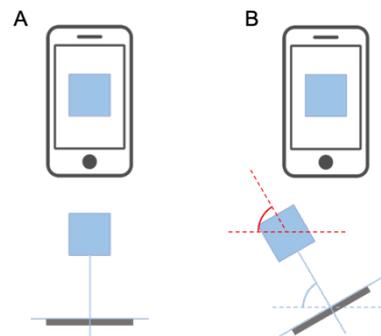


Fig. 6. Frontal match between camera and 3D model. (A) Viewed from the front. (B) Viewed from an angle.

2.4 File format of 3D model

In order to bind the graph on the virtual space side, coordinate information for each of the parts to be linked is required. If the 3D model is made of only one part, a problem arises when associating this graph. That is, obtaining the coordinates of the parts to be associated.

If a graph is to be associated with the part shown in blue in Figure 7 (left), it is necessary to calculate a vector from the reference coordinates of the 3D model in order to obtain its coordinates. If the 3D model consists of multiple parts, as shown in Figure 7 (right), this problem does not arise. Since reference coordinates are needed to create the parts, each part has its own coordinate information. The availability of the coordinate information for individual parts makes it easy to link any of the parts to a graph.

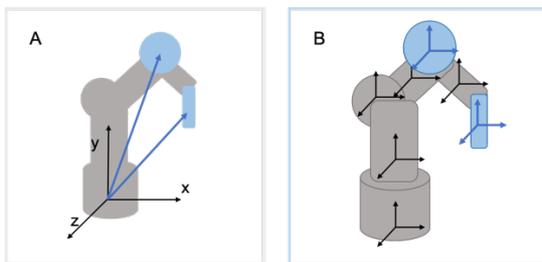


Fig. 7. Composition of parts of the 3D model. (A) Composed of only one part. (B) Composed of multiple parts.

Therefore, the format of the 3D model used in this study is a USD file, which is a 3DCG format developed by Pixar. It has features such as the ability to create individual parts in a hierarchical structure and to add animation information.

2.5 Creating of graphs

First, a graph is created and converted to image format. Next, a planar object is created. The output position of the planar object follows the coordinates of the graph-related parts of the USD model. Then, an image of the graph is pasted as the texture of the planar object. In this way, the graph is associated with the USD model.

The default size of the graph is the smallest rectangle that can cover the USD model when the model is viewed in the xy-plane, and the default size is the size of one rectangle divided into 16 equal parts of 4*4.

3. TOOL DEVELOPMENT

The architecture of the tool is shown in Figure 8. We used SceneKit and ARKit to create the tools in this study, both of which are frameworks provided by Apple Inc. SceneKit is a framework for displaying 3D content, including physics simulations and animations, while ARKit is a framework for AR experiences, including plane detection for superimposing 3D models on the actual device. ARKit is a framework for AR experiences, and supports plane detection functions for superimposing 3D models on the actual device.

First, a virtual space is created using SceneKit. At this point, the USD file is loaded and a graph is created. To display the created virtual space in augmented reality, environmental information about the real space is required, which is requested from ARKit. ARKit then performs environmental recognition, such as plane detection, and sends the acquired real-space information to SceneKit. This real-space information is then reflected in the virtual space and checked by SceneKit. ARKit and SceneKit repeatedly exchange information and update it in real time.

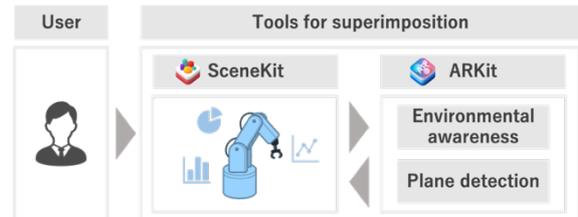


Fig. 8. The caption should be placed after the figure.

3.1 Overlay of USD model and actual equipment

First, when the application is launched, the device's camera is first turned on. Then, spatial recognition is performed, including plane detection. Even if there are multiple planes in the camera space, any of them can be detected as long as they are planes. When a plane is detected on the edge line from the center of the device screen, the coordinates of the intersection of the plane and the vertical line are obtained, and the USD model is superimposed based on the coordinates. The USD model glides along the plane in accordance with the movements of the smartphone. Once the USD model is superimposed on the actual device, the reference coordinates of the USD model are fixed in real space. In this way, after the coordinates are fixed, the USD model is no longer affected by the movement of the device. Now that the superimposition has been completed, superimpose a graph. Since the USD model is only used for superimposition, and the purpose of this research is to superimpose a graph on an object in real space, the graph is already associated with each part of the device. Once the superposition is complete, the USD model is made transparent. In this way, the graph appears to be directly connected to the real object.

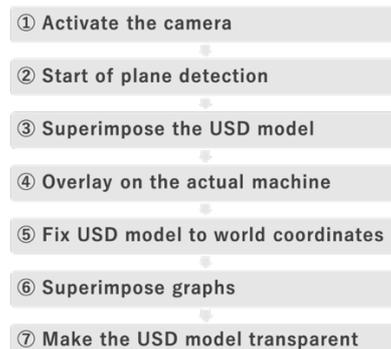


Fig. 9. The tool usage procedure.

3.2 USD model of the actual machine

An example of an inverted pendulum, which is often used to evaluate the performance of control systems, is used to illustrate this point. An actual inverted pendulum is shown in Figure 10. Figure 11 shows a USD model created with the same shape and size based on the measured values. As shown in Figure 12, the USD model has a hierarchical structure, with coordinate information for each part.



Fig. 10. Actual inverted pendulum.

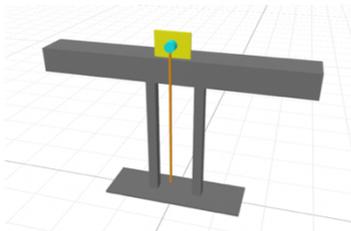


Fig. 11. USD model of inverted pendulum.



Fig. 12. Structure of USD model.

3.3 Creating of graphs

An external library for graph creation called Charts[4] is used. The created graph is pasted onto a planar object associated with the USD model and superimposed in real space, as shown in Figure 13.

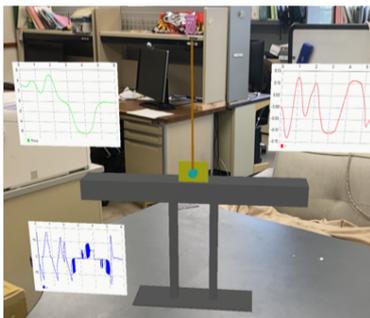


Fig. 13. Superimposing of graphs on real space.

4. EVALUATION

When fixing the USD model in real space, some errors occur, as shown in Figure 14.

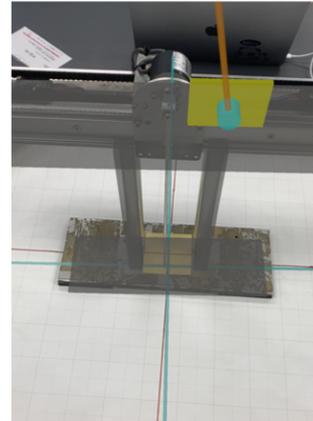


Fig. 14. Overlap error between actual machine and USD model.

A red cross is displayed at the center of the xz -plane of the real machine and a blue cross at the center of the xz -plane of the USD model. The crosses are superimposed 20 times, and the distance between the centers of the crosses and the angle of misalignment are measured. The distance between the centers is shown in Figure 15 (left), and the angle of misalignment is shown in Figure 15 (right).

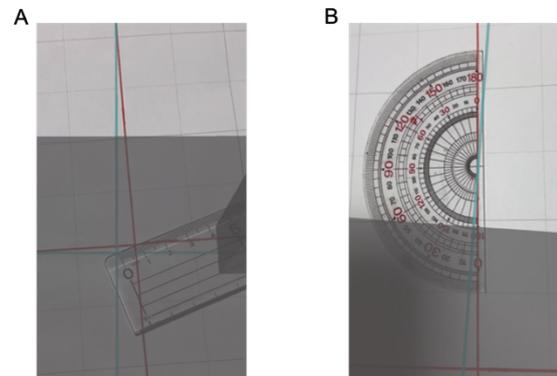


Fig. 15. Measuring Overlap Errors. (A) Measuring distance between the centers. (B) Measuring angle of misalignment.

The average of each error is shown in Table 1. These errors are independent of the size of the USD model and depend on the performance of cameras and their calibration. In other words, the larger the size of the USD model, the more negligible these errors become. Conversely, the smaller the size of the USD model, the higher the error rate will be, making it difficult to achieve the goal of linking the graph to the actual machine. For the USD model used in this study, the error is considered to be acceptable because the user is expected to be able to fully understand the connection of the graph to the actual machine.

Table 1. Average of superimposed errors.

	Average of errors
Distance between the centers	1.3 cm
Angle of misalignment	6°

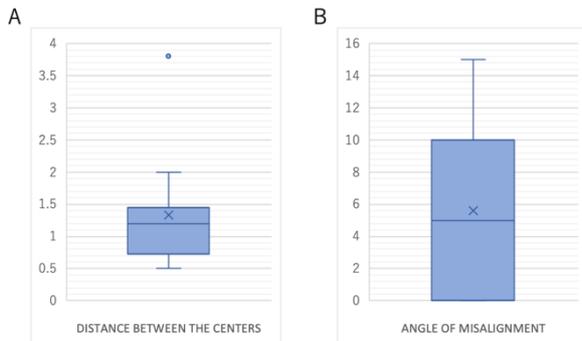


Fig. 16. Box-and-whisker diagram. (A) Measuring distance between the centers. (B) Measuring angle of misalignment.

5. CONCLUSION

In this study, we proposed a method of displaying graphs of simulations and experiments in AR around an actual device through a camera on a mobile terminal, and developed a tool for displaying graphs in augmented reality based on the proposed method. We confirmed the usefulness of the tool by creating a USD model, creating graphs, superimposing them on the actual machine, and evaluating them.

As a future prospect, we are considering adding a function that allows users to search for and load USD models other than the inverted pendulum from the Web and use them, in order to increase the tool's versatility. To improve the accuracy of superposition, we are considering adding functions to move and rotate the USD model by dragging.

The tool would be more convenient if it could be completed using only the USD model without the need for a Csv file. We are also considering the possibility of creating graphs from the animation data in the USD files. In addition, the size of the graph and the position of the output should be considered in the future.

We are also considering conducting a questionnaire to gather subjective opinions on the evaluation method. In the questionnaire, we are considering evaluating whether the tool reduces the burden on the user and the level of achievement, which cannot be evaluated quantitatively. By conducting this questionnaire, we would like to further investigate the usefulness of the proposed method.

REFERENCES

- [1] Sho Yamamoto, "Researches and Issues of Augmented Reality / Virtual Reality for Learning Environment and Educational Support System", *Japan Society for Information and Systems in Education*, vol. 36, no. 2, pp. 49-56, 2019
- [2] Nobutomo Matsunaga, Yudai Takeuchi and Hiroshi Okajima, "A construction method of obstacle avoidance system for welfare vehicle by spatial mapping of driving environment using mixed reality", *The Japan Society of Mechanical Engineers*, vol. 87, no. 894, pp. 1-16, 2021
- [3] UploadVR, graphmented, <https://uploadvr.com/graphmented-brings-google-sheets-real-world-arkit/>, (accessed 14 July, 2022)
- [4] Daniel Cohen Gindi, Charts, <https://github.com/danielgindi/Charts/>, (accessed 14 July, 2022)