

# Intuitive Manipulation of Virtual Clay in Augmented Reality

Taketo Kamasaka, Kodai Miyamoto, Takahiro Ishizu, Kenji Aoki, Makoto Sakamoto

**Abstract:** *In recent years, there has been a lot of research on how to achieve interaction between users and virtual objects using augmented reality. Interaction technologies in augmented reality need to enable users to handle virtual objects intuitively. In addition, since hands are the main means of interaction with objects in real life, it is also necessary to enable interaction operations with hands on virtual objects [1]. In order to make it possible to intuitively handle objects in virtual space using hands in real space, it is necessary to consider whether physical phenomena in real space and virtual space are correctly superimposed (physical consistency). In this study, we proposed a system that allows users to intuitively handle the deformation, movement, and merging of virtual objects in augmented reality. The system was then used by four university students to compare it with existing studies [2].*

**Keywords :** *Augmented reality, Hidden surface removal, ARToolKit, Leap Motion Controller*

## I. INTRODUCTION

In recent years, interaction technology in augmented reality (AR) has been required to realize intuitive interaction between users and virtual objects. In augmented reality, in order to overlap real and virtual objects without any sense of discomfort, the following four consistencies are considered: first, geometric consistency, which is the consistency of alignment between real and virtual space; second, temporal consistency, which is the consistency of synchronized motion between real and virtual space; third, engineering consistency, which is the consistency of optical phenomena in real and virtual space, and the fourth is physical consistency, which is the consistency of physical phenomena in real and virtual space [3]. In this paper, we propose a system that enables users to perform operations such as transforming, moving, and merging virtual objects by means of bare hand gestures. In order to enable users to handle virtual objects intuitively, we used their hands, which

are the main means of interaction in real life. As for the operations of grasping and crushing the virtual object, it is difficult for the system to distinguish between them since the user will operate the virtual object with similar appearance. To realize the distinction, we considered the Euclidean distance between the thumb and index finger that the user brings close to the virtual object. The goal of this system is to achieve a higher level of interaction between the user and the virtual object.

## II. RESEARCH BACKGROUND

In recent years, there has been a lot of research on the realization of user interaction with virtual objects using augmented reality. N. Kato, H. Ishigaki, S. Suzuki, and M. Sato [4] introduced a flexible virtual object by using a physics engine, and realized deformation manipulation of the virtual object through interaction with the user by reflecting the crushing motion with bare hands on the virtual object. In addition, R. Suzuki, S. Suzuki, M. Sato [5] realized the movement operation of a rigid virtual object by user interaction. However, operations such as maintaining the deformed shape, moving, and merging have not been realized for flexible virtual objects. In this study, we aimed to realize intuitive operations such as deforming, maintaining the deformed shape, and merging of virtual objects, as if the user were kneading clay. The significance of this research is that this system can be applied to 3DCG modeling work to enable more intuitive modeling.

## III. PREPARATION FOR RESEARCH

### A. ARToolKit

ARToolKit [3] is a programming library for C/C++ that helps implement AR applications. It was developed by Hirokazu Kato, a professor at Nara Institute of Science and Technology. With ARToolKit, you can create applications that use a camera to read patterns printed on paper and overlay 3D objects (3DCG) on top of them. There are four main functions provided: acquiring images from the camera, detecting markers and recognizing patterns, measuring the 3D position and orientation of the markers, and synthesizing and displaying live-action images and 3D CG. Since it does not include functions for manipulating 3DCG, programmers will have to use ARToolKit, OpenGL and OpenGL Utility Toolkit to create applications.

### B. OpenCV

OpenCV [6] is an open-source computer vision library that can be used on Linux, Windows, and Max OS X. It was designed with a priority on computational efficiency and an emphasis on real-time applications.

Manuscript received on September 02, 2021.

Revised Manuscript received on September 08, 2021.

Manuscript published on September 30, 2021.

\* Correspondence Author

**Taketo Kamasaka\***, Faculty of Engineering, Miyazaki University, Miyazaki, Japan. Email: hm16011@student.miyazaki-u.ac.jp

**Kodai Miyamoto**, Faculty of Engineering, Miyazaki University, Miyazaki, Japan. Email: hm16043@student.miyazaki-u.ac.jp

**Takahiro Ishizu**, Faculty of Engineering, Miyazaki University, Miyazaki, Japan. Email: fruits2000jp@yahoo.co.jp

**Kenji Aoki**, Faculty of Engineering, Miyazaki University, Miyazaki, Japan. Email: aoki@cc.miyazaki-u.ac.jp

**Makoto Sakamoto\***, Faculty of Engineering, Miyazaki University, Miyazaki, Japan. Email: fruits2000jp@yahoo.co.jp

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

## C. OpenGL

OpenGL [7] is a software interface to graphics hardware. The interface consists of approximately 250 individual commands that can be used to specify the objects and operations required to create interactive 3D applications. It is designed to be an efficient, hardware-independent interface that can be implemented on a variety of hardware platforms.

## D. Bullet Physics

Bullet Physics [8], [9] is an open source 3D physics engine developed by Erwin Coumans, which can handle not only rigid bodies but also flexible bodies. Bullet Physics is widely used in the development of game software for movies and game consoles. Bullet Physics is widely used in the development of movies and game software for various game consoles, etc. Since the main purpose of Bullet Physics is to solve the laws of physics, CG processing must be done using OpenGL, etc.

## E. Leap Motion Controller

The Leap Motion Controller is a sensor that specializes in hand and finger detection. As shown in Fig. 1, the system consists of three infrared LEDs and two cameras. The top plate is a filter that blocks visible light and allows infrared light to pass through. The top plate is a filter that allows infrared light to pass through while visible light does not. This allows us to measure the target object using a stereo camera with infrared images.

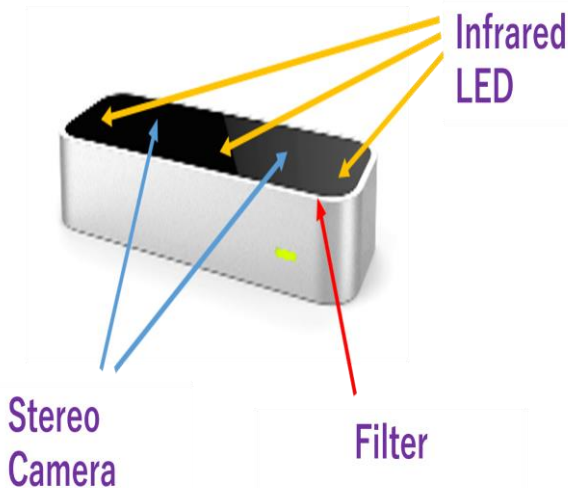


Fig. 1: Leap Motion Controller [10]

## IV. CONFIGURATION OF THE SYSTEM

As shown in Fig. 2, this system consists of a web camera for acquiring images in real space and recognizing markers, a Leap Motion Controller for acquiring 3D coordinates of the fingers, and a PC for performing other operations and outputting images. In this system, network programming is used to prevent interference between the Leap Motion Controller and the Web camera [11]. In the network programming, UDP is used as a transport protocol suitable for real-time communication. The server side handles the acquisition of the 3D coordinates of the hand by the Leap Motion Controller, and the client side handles the processing from the acquisition of the image by the web camera and the physical calculation to the image output.

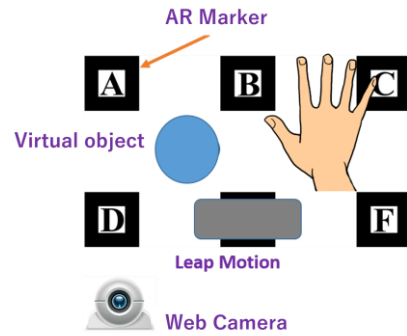


Fig. 2: Configuration of the system

### A. AR Marker

In our system, the virtual object on the AR marker is manipulated with the fingers. In our system, the virtual object on the AR marker is manipulated by the user's hand, so the AR marker may be hidden by the user's hand, and the CG overlay may not be displayed. To deal with this problem, we use a method that treats multiple AR markers whose relative positions are known as one AR marker. In this method, if at least one of the AR markers is successfully detected, the position of the hidden AR marker can be calculated using the relative relationship. This method is implemented by ARToolKit, which creates a configuration file describing the location of AR markers in advance, and loads it when the program is executed.

### B. Virtual object

Bullet Physics can handle flexible bodies using a model called a spring/ploid system. Unlike rigid bodies, the motion of a flexible body cannot be represented by a single position and posture. Therefore, the shape is represented by points and connections between them, and the motion of each point represents the motion of the whole body.

## V. PROPOSED METHOD

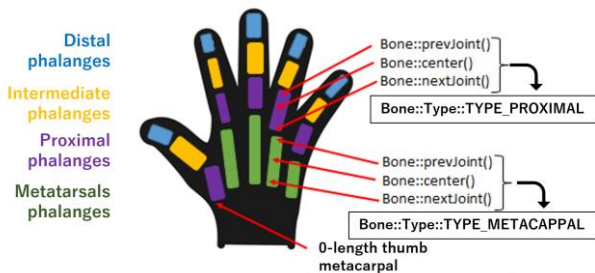
In this research, we aim to enable users to manipulate virtual objects more intuitively by creating them and performing basic operations such as deforming, moving, and merging them. V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn [1] used various hand gestures to manipulate virtual objects, such as grasping and moving a rigid virtual object, using AR markers attached to the fingertips. They showed that gestures are a very intuitive way of interacting with virtual objects. Therefore, in this study, we propose intuitive manipulation of virtual objects using gestures in order to realize more natural interaction between users and virtual objects. In this study, we use the Leap Motion Controller [12], which specializes in acquiring 3D coordinates of the fingers, to manipulate the virtual object by detailed movements of the fingers. In addition, we introduce a flexible virtual object using Bullet Physics to realize interaction between the user and the virtual object, taking into account the physical consistency in augmented reality. We also perform hidden surface processing so that the user can recognize the back and forth relationship between his own hand and the virtual object.

**A. Processing flow of the system**

As shown in Fig. 2, a piece of paper with AR markers printed on it and a Leap Motion Controller are placed and observed by a web camera. The virtual object is displayed when the AR markers are recognized. The system first obtains the 3D coordinates of the fingers by the Leap Motion Controller. Next, the system uses a web camera to recognize the AR marker and acquire images of the real space. Then, the 3D coordinates of the hand are sent and received between the server side and the client side. Finally, the virtual object is generated, deformed, moved, merged, and hidden surface processed according to the user's gestures.

**B. Coordinates to be acquired by Leap Motion Controller**

The Leap Motion Controller acquires the x, y, and z coordinates of the fingertips in order to track the virtual object of the fingertips in Bullet Physics. Also, for OpenCV hidden surface processing, get the x, y, z coordinates of the bones::center() of the terminal, metatarsal, basal, and metacarpal bones in Fig 3. In addition, when generating a virtual object by gestures, the recognition of the left hand and the right hand by the Leap Motion Controller is related to the processing by gestures. Therefore, in our system, we distinguish whether both hands are recognized by the size of the acquired finger data. The above process is performed on the server side, and the acquired data is sent to the client side.



**Fig. 3: Coordinates of the joint to be acquired [13]**

**C. Collision processing**

In order to deal with collisions, we need to find out whether objects have collided with each other or not. This is called collision detection. To perform collision detection, we need to use the geometry of the objects. Here, the visible shape and the collision shape must be different. For example, the visible geometry is very complex, and collision detection would require a huge amount of computation. Therefore, a simplified shape for collision processing is provided to reduce the computational complexity of collision processing. In this study, OpenGL is used as the 3D graphics to represent visible shapes. A typical model consists of several hundred to several tens of thousands of polygons. If the collision judgment of these models is performed as it is, it is necessary to try all combinations of polygons that make up each model. This would be impossible to calculate in real time. Therefore, in Bullet Physics, collision detection is divided into two phases: a broad phase for rough judgment and a narrow phase for fine judgment. Broad phase is a detailed collision detection method to obtain the accuracy of physical phenomena. Algorithms such as those based on the CJK method are introduced to take into account the shape of the two colliding objects, their contact points and intersection

shapes. Narrow-phase detects potentially colliding pairs using a simple shape that wraps around the shape. This shape is called the boundary volume. In Bullet Physics, the boundary volume is an axis-parallel bounding box (AABB), which is a rectangular box with each side parallel to the x, y, and z axes.

**D. Hidden surface removal**

In augmented reality, virtual objects are superimposed on real space images later. Therefore, even when the virtual object is farther away than the real object, it is not hidden by the real object, but is always displayed as if it were in front of the real object, as shown in Fig. 4. This is called the occlusion problem in augmented reality, and it prevents the user from correctly recognizing the front-back relationship between the virtual object and the hand when interacting with the virtual object [14]. This is thought to make the user feel visually uncomfortable and prevent him/her from performing intuitive operations. Our system solves this problem by processing the hidden surface of the hand and the virtual object in the virtual space, using the 3D coordinates of the fingers obtained from the Leap Motion Controller and the RGB image obtained from the web camera shown in Fig. 5. The procedure is as follows: first, we obtain the RGB image from the web camera, detect the hand region by extracting the skin color region, and generate the binary image in Fig. 6; then, we obtain the 3D coordinates of the hand by the Leap Motion Controller, and if a certain part of the hand is in front of the virtual object, we generate the image with that part in Fig. 7. If a part of the hand is in front of the virtual object, the masked image is generated. Synthesize the real space and the virtual object, and generate an image in which the virtual object is always displayed in front. Combine the first extracted hand region with the original RGB image to generate an image in which the hand is always displayed in front, Fig. 8. Combine the masked region in the 3D image of the hand with the image in which the hand is always displayed in front, and generate the image shown in Fig. 9. With the above procedure, when the user's hand is in front of the virtual object, the user's hand is correctly displayed in front of the virtual object, and the user can grasp the front-back relationship between his or her hand and the virtual object.



**Fig. 4: Image without hidden surface processing**





Fig. 5: RGB image



Fig. 6: Binary image

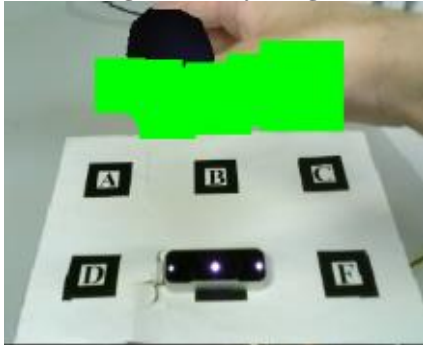


Fig. 7: Image with masked finger in front of virtual object



Fig. 8: Image of a hand being displayed in front of a virtual object



Fig. 9: Synthesis result

## VI. COMPOSITION OF VIRTUAL OBJECTS

In our system, we use the physics engine Bullet Physics to create virtual objects that are capable of physical operations [8][9][14]. Fig 10 shows the structure of a virtual object in our system. The virtual objects in our system are composed of a "virtual object to be manipulated," which is the object to be actually manipulated with bare hands, a "virtual object at the fingertips," which follows the tips of the user's thumb and index finger, and a "virtual object for movement," which is attached to the "virtual object to be manipulated" and provides its position information. Here, the shape of the "virtual object to be manipulated" is assumed to be a sphere, and the number of spheres is limited to two. The "fingertip virtual object" and the "moving virtual object" are not shown. The vertex near the north pole and the vertex near the south pole of the "virtual object to be manipulated" are fixed at the time of generation. In addition, a "moving virtual object" is attached near the north pole of the "virtual object to be manipulated" in order to move it. In this study, we define a spring/mass point only on the surface, and apply a force that maintains the volume of the object. This is to prevent the object from collapsing by applying a force calculated from the pressure corresponding to the current volume of the object to the mass points. In this study, we use a method using only the surface triangle mesh.

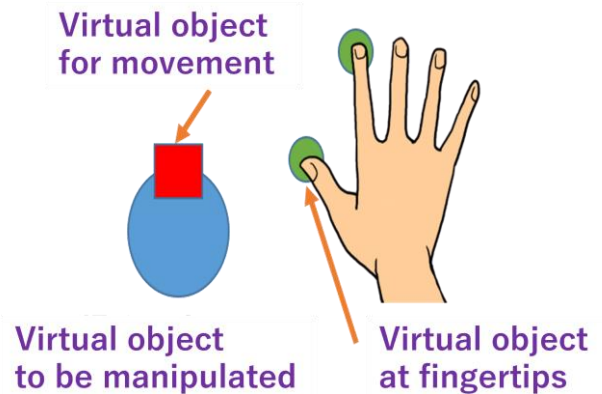


Fig. 10: Composition of virtual objects

## VII. MANIPULATION OF VIRTUAL OBJECT

In this study, we propose two gestures, grabbing and moving. The distance  $d$  between the thumb coordinates  $(x_1, y_1, z_1)$  and index finger coordinates  $(x_2, y_2, z_2)$  in the gesture was calculated using the formula for finding the 3D Euclidean distance.

$$d = \sqrt{(x_1^2 - x_2^2)^2 + (y_1^2 - y_2^2)^2 + (z_1^2 - z_2^2)^2} \quad (1)$$

### A. Deformation of virtual object

The user can deform the "virtual object to be manipulated" by gestures using the thumb and index finger of the right hand. The procedure is described below. First, the user opens the right hand on the Leap Motion Controller and holds it out palm down. Then, the "virtual object of the fingertips" is placed at the tips of the thumb and index finger.

When the user squashes the object with the thumb and forefinger, the virtual object and the fingertip object collide. The "virtual object to be manipulated" is deformed by the above operations.

**B. Movement of virtual object**

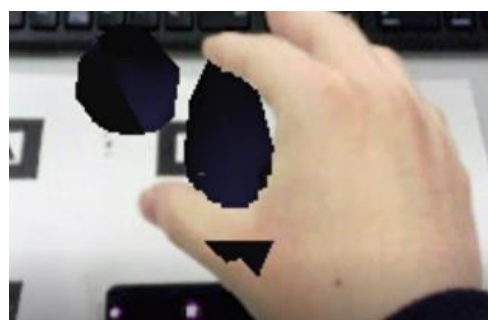
In this study, we changed the gesture from scooping up a virtual object in existing research [2] to grabbing it and moving it. The user can move the "virtual object to be manipulated" by gestures using the thumb and index finger of the right hand. In this system, only the second virtual object and the combined virtual object can be moved. The procedure is described below. First, the user opens his or her right hand on the Leap Motion Controller and holds it out palm down. The user grasps the virtual object to be manipulated with the thumb and index finger, and the virtual object to be moved is unfastened. When the user moves his/her hand while grasping with the thumb and index finger, the coordinates of the "moving virtual object" also move according to the position coordinates of the user's fingers. The "virtual object to be manipulated" has a "virtual object for movement" attached to it beforehand, so by moving the "virtual object for movement," the "virtual object to be manipulated" can also be moved. When the user releases the thumb and forefinger at the point where they want to place the object, the "virtual object to be manipulated" is fixed and placed on the spot.

**VIII. EXECUTION RESULT**

In this study, we implemented the system using OpenCV 3.2.0, which is a library for image processing, OpenGL, which is suitable for rendering 3DCG, and ARToolKit, which is suitable for developing AR applications. As the integrated development environment, Visual Studio Express 2013 for Window Desktop was used on the server side, and Visual Studio Community 2017 was used on the client side to compile the source code. This was because the framework used on the server side was not compatible with the latest Visual Studio compiler. We used Elecom WEB camera UCAMC310FBBK for marker recognition and Leap Motion Controller LM-010 for acquiring hand coordinates.



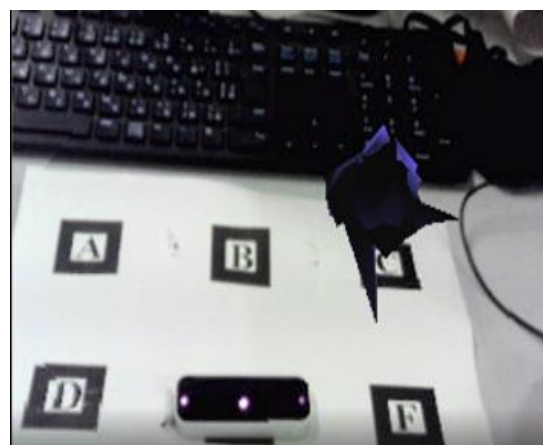
**Fig. 11: Deformation of virtual object**



**Fig. 12: Movement of virtual object**



**Fig. 13: Merging of two virtual objects**



**Fig. 14: Placement of virtual objects**

**IX. EVALUATION EXPERIMENT**

We conducted a questionnaire-based survey to evaluate whether the interaction operations with virtual objects using our system are more intuitive than those using existing systems, and to identify areas for improvement.

**A. Evaluation Method**

We asked four university students, the subjects, to compare the basic operations on virtual objects shown in Figs. 11, 12, 13, and 14 with the existing system. The subjects were asked to try out our system, and after the trial, they were asked to evaluate our system by answering a questionnaire in free-response format for each of the following items: "Good points", "Difficult points", and "Other improvements".

**B. Evaluation Results**

The results of the questionnaire were as follows. As for "good points," the following opinions were raised: "The operation method was intuitive" and "The gestures were easy to understand. As for the "Difficult points", the following opinions were raised: "Sometimes objects did not respond", "It was difficult to understand the difference between grabbing and squeezing gestures". As for "other improvements," the following comments were raised: "I would like feedback on which action is being detected now," "I would like to be able to choose the color," and "I would like to see objects other than spheres implemented.

## X. CONCLUSION

From the "good points," it can be said that the interaction with the virtual object using the new gestures was intuitive and allowed for more natural instruction. As for the reason for the comment, "Sometimes the object did not respond," in the "Difficult points," it was found that when the user tried to manipulate the "virtual object to be manipulated," the finger seemed to be on the "virtual object to be manipulated" as seen from the Web camera, but the position information obtained from the Web camera and Leap Motion Controller was not correct due to a delay. However, the position information obtained from the webcam and the Leap Motion Controller may have errors due to delays, etc., so that the position of the virtual object seen from the webcam differs from the actual position of the fingertip. The reason for the comment, "It was difficult to understand the difference between the grabbing and squeezing gestures," is that the two gestures are similar and it was difficult to adjust the distance between the two points of the thumb and index finger. In the future, we would like to solve the problem of "Difficulty in operation" and add the functions mentioned in "Other improvements" so that users can interact with virtual objects more intuitively.

## REFERENCES

1. V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn, "FingARTips-Gesture Based Direct Manipulation in Augmented Reality", Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia 2004, 2004.
2. Takahiro Ishizu, "A Proposal for Intuitive Manipulation of Virtual Objects Considering Physical Consistency in Augmented Reality", Graduation Thesis, Faculty of Engineering, University of Miyazaki, 2018.
3. S. Hashimoto, "ARToolKit Introduction to Augmented Reality Programming", ASCII, 2008
4. N. Kato, H. Ishigaki, S. Suzuki, M. Sato, "A Study on the Softness Presentation of Virtual Objects in the Grasping Motion with Bare Hands", Journal of the Institute of Image Information and Television Engineers, 2017, Vol. 71, No. 3, pp. J121-J124.
5. R. Suzuki, S. Suzuki, M. Sato, "Visual Improvement of AR System Focusing on Barehanded Grabbing Motion", Journal of the Institute of Image Information and Television Engineers, 2014, Vol. 68, No. 7, pp. J299-J301.
6. Gary Bradski, Adrian Kaebler (Author), Koichi Matsuda (Translation), "Image Processing and Recognition with the OpenCV Computer Vision Library", O'Reilly Japan, 2009.
7. Open GL Development Committee (Author), Koichi Matsuda (Translation), "OpenGL Programming Guide, Original 5th Edition", Pearson Education, 2009.
8. Maciej Matyka, Mark Ollila, "Pressure Model of Soft Body Simulation", 2004.
9. H. Hashimoto, T. Sasaki, "Getting Started with Bullet Physics: 3D Motion Simulation", Ohmsha, 2011.
10. Developed by Leap Motion. [Online]. Available: [https://developer.leapmotion.com/documentation/v2/cpp/devguide/Leap\\_Overview.html](https://developer.leapmotion.com/documentation/v2/cpp/devguide/Leap_Overview.html).
11. Y. Imamura et al., "Future University Hakodate 2011 System Information Science Practice", Future University Hakodate, 2011.
12. K. Nakamura, "Leap Motion Programming Guide", Kogaku-Sha, 2015.
13. Tracking Hands, Fingers, and Tools. [Online]. Available: <https://www.buildinsider.net/small/leapmotioncpp/002>.
14. R. Katahira, S. Masato, "Development of A System enabling Display of Gripping the AR Object Using Leap Motion Controller", The 29th Annual Conference of the Japanese Society for Artificial Intelligence, 2015.
15. Information and Media Experiments Application Development Using Physics Engine. [Online]. Available: <http://slis.tsukuba.ac.jp/~fujisawa.makoto.fu/lecture/iml/>

## AUTHORS PROFILE



**Taketo Kamasaka** is a master student at Department of Computer Science and System Engineering, University of Miyazaki. His current research interests are computer graphics, entertainment computing, and so on.



**Kodai Miyamoto** is a master student at Department of Computer Science and System Engineering, University of Miyazaki. His current research topic is a fundamental study on educational support using VR technology.



**Takahiro Ishizu** received the Master's degree in computer science and systems engineering from University of Miyazaki in 2020. He is presently a systems engineer in Systems Device Division, NEC Corporation. He has published papers mainly in the fields of image processing, computer graphics and data science.



**Kenji Aoki** received Ph.D. of Engineering from Kagoshima University in 2010. He is currently working in Information Technology Center at University of Miyazaki as Associate Professor, since 2010. His research interests include bio-informatics, evolutionary computation, information system and Intelligent systems. He is a member of IPSJ and JSET.



**Makoto Sakamoto** received the Ph.D. degree in computer science and systems engineering from Yamaguchi University in 1999. He is presently a professor in the Faculty of Engineering, University of Miyazaki. His first interests lay in hydrodynamics and time series analysis, especially the directional wave spectrum. He is a theoretical computer scientist, and 2. his current main research interests are automata theory, languages and computation. He is also interested in digital geometry, digital image processing, computer vision, computer graphics, virtual reality, augmented reality, entertainment computing, complex systems and so forth. He has published many research papers in that area. His articles have appeared in journals such as Information Sciences, Pattern Recognition and Artificial Intelligence, WSEAS, AROB, ICAROB, SJI, IEEE and IEICE (Japan), and proceedings of conferences such as Parallel Image Processing and Analysis, SCI, WSEAS, AROB and ICAROB.