International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Hidden Surface Removal in Augmented Reality: Hand Region Extraction using PCA

Takahiro Ishizu, Makoto Sakamoto, Kenji Sakoma

Abstract*: Recently, augmented Reality (AR) is growing rapidly and much attention has been focused on interaction techniques between users and virtual objects, such as the user directly manipulating virtual objects with his/her bare hands. Therefore, the authors believe that more accurate overlay techniques will be required to interact more seamlessly. On the other hand, in AR technology, since the 3-dimensional (3D) model is superimposed on the image of the real space afterwards, it is always displayed on the front side than the hand. Thus, it becomes an unnatural scene in some cases (occlusion problem). In this study, this system considers the object-context relations between the user's hand and the virtual object by acquiring depth information of the user's finger using a depth sensor. In addition, the system defines the color range of the user's hand by performing principal component analysis (PCA) on the color information near the finger position obtained from the depth sensor and setting a threshold. Then, this system extracts an area of the hand by using the definition of the color range of the user's hand. Furthermore, the fingers are distinguished by using the Canny method. In this way, this system realizes hidden surface removal along the area of the user's hand. In the evaluation experiment, it is confirmed that the hidden surface removal in this study make it possible to distinguish between finger boundaries and to clarify and process finger contours.

***Keywords* : Occlusion problem, Principal component analysis (PCA), Canny edge detection, Image processing.**

I. INTRODUCTION

Augmented reality (AR) has been used in many fields including education, medicine and entertainment, and its applications has become widespread [1]-[3] in recent years.

Further, interaction with the virtual objects is required in those papers.

Therefore, it is expected that interaction technology using AR technology will be more and more developed in the future, but we think that it requires a more accurate technology.

On the other hand, since hands are our main means of interaction with objects in real life, it is necessary for AR interfaces to be able to manipulate virtual objects with the user's bare hand.

Furthermore, it is important to be able to adapt to more detailed tasks by hand in order to handle a wide range of fields such as medical care and education.

In the conventional study [4], by putting a marker, a special glove, etc. on a user, the user's position information was acquired and interaction with a virtual object was realized. However, these methods may give the user a sense of discomfort such as weight due to wearing. Therefore, bare hand interaction is required.

However, since the 3-dimensional (3D) model displayed by the AR is superimposed on the image of the real space afterwards, the 3D model is always displayed on the front side and user's hand is hidden by virtual objects. Thus, the scene may become an unnatural scene, and the user cannot see the object-context relations of the virtual object and his/her hand, and feels that it is difficult to manipulate the virtual objects.

In the existing study [5], the system used a transparent 3D model and the 3D model followed each fingers of the user. In this way, they performed hidden surface removal based on the depth information of user and 3D model. However, since the 3D model to be followed is larger than the finger, a wider range than the actual finger was displayed on the front (see Fig. 1).

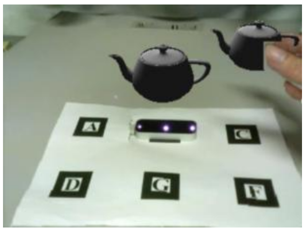

Fig. 1. Hidden surface removal in existing study

In other related study [6], using the Kinect sensor in order to obtain depth information of the user, hidden surface removal was realized when an arm was inserted between multiple virtual objects.

Also, it was the same even when it came to deformable virtual object.

However, the Kinect sensor is a sensor that roughly recognizes human motion and depth information, and the study didn't focus on the interaction between hands and virtual objects.

Manuscript published on January 30, 2020.
* Correspondence Author
Takahiro Ishizu*, Department of Computer Science and System Engineering, University of Miyazaki, 889-2192, Japan. Email: hm14002@student.miyazaki-u.ac.jp
Makoto Sakamoto*, Department of Computer Science and System Engineering, University of Miyazaki, 889-2192, Japan. Email: fruits2000jp.yahoo.co.jp
Kenji Sakoma, Department of Computer Science and System Engineering, University of Miyazaki, 889-2192, Japan. Email: hm15022@student.miyazaki-u.ac.jp

In this study, we aim to handle more detailed virtual object operations by performing hidden surface removal along the outline of the user's hand.

At the end of this paper, we perform an operation of grasping the virtual object of the primitive model by hand and confirm that it works correctly.

Moreover, in this study, we will deal with a case that there are multiple virtual objects that have not been discussed in the existing study, and consider the execution results.

II. PROPOSAL

Author In order to conduct hidden surface removal along the contour of the user's hand, this system detects hand area of the user by extracting depth information, color information and edge detection in the user's hand.

First, in order to extract hand area, this system uses the RGB image based on color information.

Furthermore, in order to determine “skin color” in real-time, this system performs principal component analysis (PCA), and sets threshold.

Next, based on the depth information of the hand, a blue point group is created on the thumb, and a green point group is created on the other fingers.

In this study, we use Leap Motion Controller to acquire hand depth information. These point groups are displayed at their positions only when the hand is in front of the virtual object. In this way, when the user's hand is near from camera than the virtual object, the user's hand is correctly displayed on the front side of the virtual object, and the user can grasp the anteroposterior relationship between the user's hand and the virtual object.

Furthermore, the fingers are distinguished by using the Canny method.

Finally, in order to cope with multiple virtual objects and distinguish the object-context relations of the blue point group and the green point group, we use the Z-buffer method. Thus, the outline of the hand is extracted, and hidden surface removal along the outline of the hand is realized.

A. System Component

This system consists of a Web camera, the Leap Motion Controller, PC and AR marker.

In this study, since this system needs accurate position information of user's fingers, we use the Leap Motion Controller that can acquire various data relating to user's hand with high accuracy.

The purpose of use of each of the devices shown above is as follows:

- Web camera: Acquisition of real image and recognition of marker.
- Leap Motion Controller: Obtaining 3D coordinates of the user's fingers.
- Monitor: Video output.
- AR marker: Acquisition of position and tracking of virtual objects.

The Leap Motion Controller and the webcam both use camera features, which causes the camera features to interfere with each other. Therefore, this system addresses this problem by using network programming.

This system acquires data of 3D coordinates of the user's fingers with the Leap Motion Controller on the server side.

In addition, this system sends the data to the client side.

On the other hand, this system acquires images with the Web camera, and conducts image processing and video output using the data which received from the server side on the client side. This system controls the position and orientation of the virtual objects is by recognizing the AR marker. Therefore, the AR marker needs to be recognized by the web camera.

However, when the user manipulates virtual objects, the AR marker may be covered by his/her own hand.

Therefore, this system adopts a method of treating Marker A to Marker F as one marker (see Fig. 2). Thus, even when a part of the marker is covered, the virtual objects can be displayed properly. We place the Leap Motion Controller on the position of Marker B.

This system displays the virtual objects on the position of Marker C and Marker G (see Fig. 2).

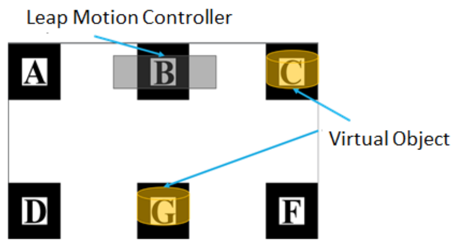

Fig. 2. AR marker

B. Development Environment

We use ARToolKit for AR marker recognition and camera control.

In addition, we develop this system using OpenGL for displaying virtual objects and OpenCV for various image processing.

Here, the resolution of the camera is 320 × 240 pixels.

C. Processing of Camera Image

This system gets RGB image with the web camera (see Fig. 3).

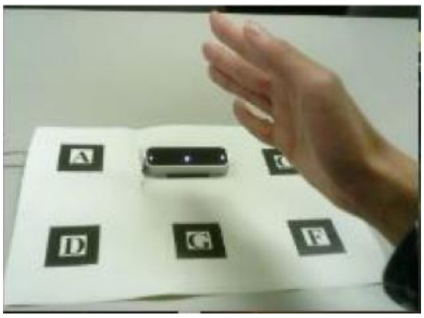

Fig. 3. RGB Image

D. Determination of “skin color”

This system extracts the area of skin color for detection of the user's hand area. Therefore, it is necessary to define the “skin color” color.

In this section, we will describe how to define this "skin color".

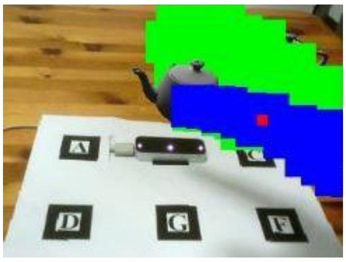

Fig. 4. Blue point groups

First, in order to determine the range of “skin color”, This system compresses RGB image data into 2-dimensional data using PCA.

We define a data matrix X_D in which the pixel values x_{xy} of the pixel xy having the coordinates (x, y) in the image space are arranged vertically as shown in equation (1).

$$X_D = \{x_{xy} | x_{xy} = (R_{xy}, G_{xy}, B_{xy}), R_{xy} \in (0,255), G_{xy} \in (0,255), B_{xy} \in (0,255), if\ xy \in C_{hand}\} \quad (1)$$

C_{hand}: Set of blue and green point cloud pixels.

Next, we determine equation (2) so that the variance of Z is maximized.

$$Z = \omega_1 X_R + \omega_2 X_G + \omega_3 X_B \quad (2)$$

where $\sum \omega_i^2 = 1$.

The eigenvectors obtained by this principal component analysis are shown below.

Table- I: Eigenvector

	R	G	B
PC1	-0.575	-0.583	-0.574
PC2	0.682	0.046	-0.73
PC3	0.452	-0.811	0.371

The formula for calculating the principal component score from equation (2) and the eigenvectors (Table-I) is shown below.

$$Z_{1xy} = -0.575R_{xy} + (-0.583)G_{xy} + (-0.574)B_{xy} \quad (3)$$

$$Z_{2xy} = 0.682_1 R_{xy} + 0.046_2 G_{xy} + (-0.730)B_{xy} \quad (4)$$

$$Z_{3xy} = 0.452R_{xy} + (-0.811)G_{xy} + (0.371)B_{xy} \quad (5)$$

The principal component score is calculated using these equations, and the distribution of the pixels is shown in Fig. 5.

Therefore, based on the distribution shown in Fig. 5, the thresholds are set using the first and third principal component scores, which are considered to be easy to set.

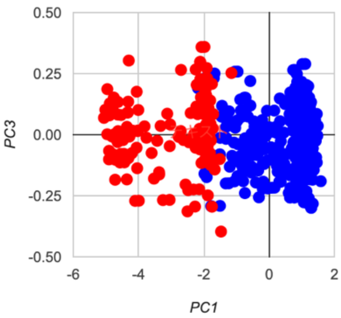

Fig. 5. Distribution of 1st-3rd principal component scores

We use a box-and-whisker plot to determine the threshold. This system uses the first and third quartiles as thresholds, respectively, and regards pixels whose RGB values fall within the interquartile range as skin color area.

E. Generation of Binary Image

This system generates binary image to detect the user's hand area using “skin color”.

The binary image generated by this system contains salt and pepper noise at first (see Fig. 6).

Fig. 6. Noise removing

Since this binary image is used in the subsequent processing, the noise still remains in the final result image (see Fig. 7).

Therefore, this system conducts erosion and dilation twice in 8-neighbor in order to remove noise.

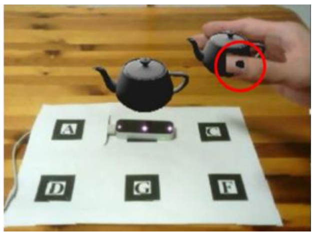

Fig. 7. Result image including noise

However, that alone cannot remove the noise sufficiently. This is because, in this method, this system has to conduct erosion and dilation multiple times in order to remove loud noise. However, it needs complicated calculation.

Therefore, regarding to the loud noise, this system removes the noise in the area of the hand by conducting the following processing (see Fig. 8).

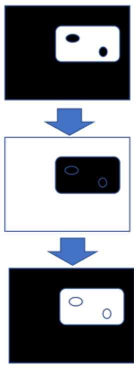

Fig. 8. Noise removing process

1) Set all the values of the outer frame of the obtained binary image to 0.

2) Determine white area including red point (which is put on the finger which is closest to the camera) as hand area.

3) Fill the inside of the hand area with black and the outside of the hand area with white.

4) Invert white and black.

In this way, all noise in the hand area is removed (see Fig. 9).

Fig. 9. Binary image

F. Generate Image Considering Depth Information

This system acquires 3D coordinates of the distal bone, middle phalanx, basal bone and metacarpal bone of the user's hand (see Fig. 10) with Leap Motion Controller.

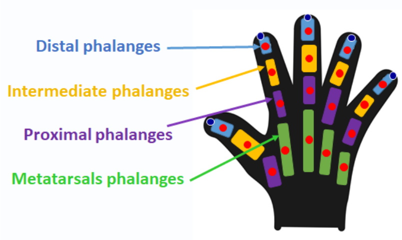

Fig. 10. Hand joint

Based on the acquired position information of the fingers, this system generates images plotting blue point group (following the position of the thumb) and green point group (following the position of a finger other than the thumb).

Blue point group displays at the positions of each joint of the thumb when the user's thumb is front of the back object (see Fig. 11).

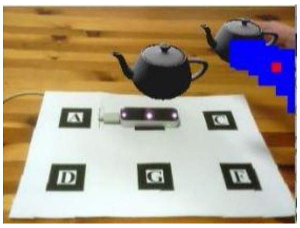

Fig. 11. Blue point group following the thumb

In addition, green point group displays at the positions of each joint of the fingers except thumb when the user's fingers except thumb are front of the back object (see Fig. 12).

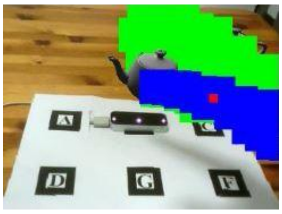

Fig. 12. Blue point group and green point group following the whole hand

Furthermore, this system makes it possible to handle multiple virtual objects of different depths. The object-context relations between the back virtual object and the hand of the user is determined depending on whether or not blue point group and green point group are displayed.

On the other hand, the object-context relations between the virtual object on the near side and the user's hand is determined by using the Z-buffer method.

Using the Z-buffer method, when a finger is positioned behind the virtual object, blue point group and green point group following the position of the finger are hidden by the virtual object.

In this way, this system can determine the object-context relations between the user's hand and the virtual objects even if there are multiple virtual objects.

G. Generate Images which Always Display The Hand in Front of Virtual Objects

This system synthesizes the original RGB image (see Fig. 13-a) only in the white area in Fig. 9 to the image in Fig. 13-b. Thus, this system generates images that the user's hand is always displayed on the front of the virtual object (see Fig. 13-c).

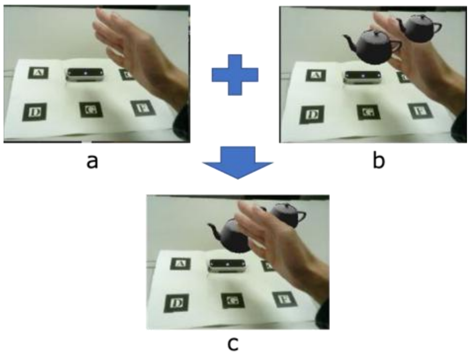

Fig. 13. Generation of images which hand is displayed front

H. Generate Result Images

The area in Fig. 13-b corresponding the area of the blue point group, the green point group and the red point group (in the case where the position of the proximal phalanges of thumb is in front of the virtual object) of Fig. 14-a are replaced with the image shown in Fig. 14-b (Fig. 13-c).

Thus, based on the depth information, this system generates images that the hand is displayed in front of the virtual object (see Fig.14-c).

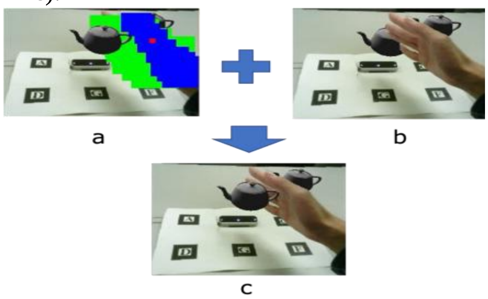

Fig. 14. Generation of result images

Distinction between Fingers Using Canny Edge Detection
The white area in Fig. 9 indicates the hand area. However, no distinction between fingers is made from this image. Therefore, if hidden surface removal is conducted based on this image, even if only the thumb should be subjected to hidden surface removal like Fig. 15, other fingers are involved and processing is conducted.

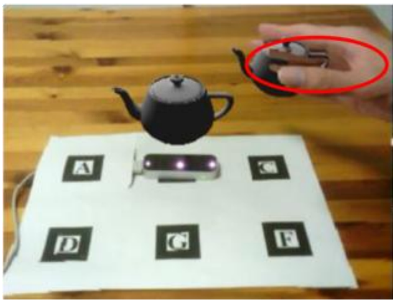

Fig. 15. Hand joint

Therefore, we need edge detection of the user's thumb in order to solve this problem.

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or more formally, has discontinuities. The point at which image brightness changes sharply are typically organized into a set of curved line segments termed as edges.

We need to detect finger edge in order to distinguish between thumb and other fingers.

Furthermore, at the time of the edge detection, the edge must be one clear line, since the area of the thumb extracted by edge detection is filled in the area surrounded by the edges in the later processing.

In this study, we use Canny edge detection algorithm [7] to find an edge of the user's thumb. Because it provides better results and efficiency than other available algorithm when we want to detect clear edges [8].

Canny edge detector uses two thresholds and if the intensity of that pixel is below that lower threshold than it is assigned " 0" value that can be none edge and if the intensity of any pixel is above that the higher threshold than it is assigned " 1" value and can be edge.

In this study, we distinguish the thumb from the other fingers.

In order to detect the thumb, the thumb is surrounded by an ellipse, and the inner area surrounded by it and the boundary displayed by edge detection is extracted. The area is regarded as a thumb area (see Fig. 16).

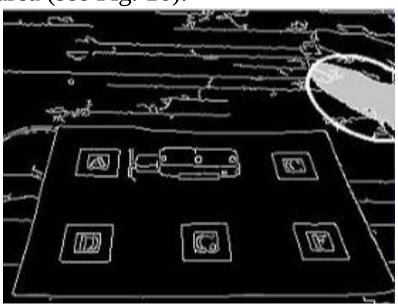

Fig. 16. Extraction of the thumb area

The following process is conducted to detect the thumb area.

1) Get two coordinates of upper left end and lower right end of blue point group if the value of y coordinate of the position of the tip of the thumb is smaller than the value of the y coordinate of the position of the base of the thumb. Otherwise, get two coordinates of upper right end and lower left end of blue point group.

Here, the y-axis is positive in the downward direction of an image.

2) Make an ellipse based on a line connecting the two points (which are the coordinates acquired in step 1).

3) Overlap the ellipse surrounding the thumb on the image generated based on the Canny edge detection (see Fig. 17-b).

4) Fill the area surrounded by the ellipse and the edge in Fig. 17-b which corresponding to the position including the red point group in Fig. 17-a with gray.

5) Regards this gray area as the thumb area.

In this way, this system generates result image (see Fig. 17-c).

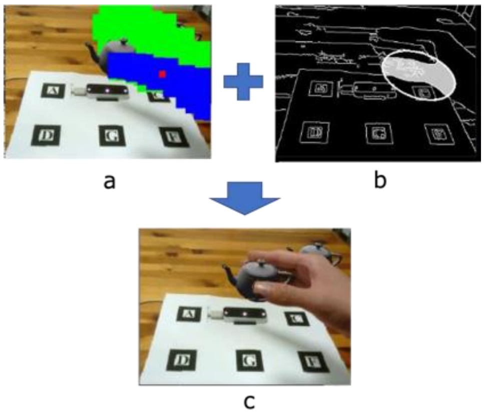

Fig. 17. Distinction between the thumb and the other fingers

III. EXPERIMENTAL RESULTS

We executed this system to confirmed whether hidden surface removal along the fingers is possible or not.

First, we checked whether this system can cope with multiple virtual objects.

In this system, the object-context relations between the back virtual object and the hand of the user is determined depending on whether or not blue point group and green point group are displayed.

In order to confirm whether this is properly processed, we placed the virtual object at position C in Fig. 2 and performed manipulation that a user grabs the virtual object.

On the other hand, the object-context relations between the virtual object on the near side and the user's hand is determined by using the Z-buffer method.

In order to confirm whether this is properly processed, we placed the virtual object at position G in Fig. 2 and performed manipulation that a user grabs the virtual object.

Furthermore, we performed grasping manipulations on virtual objects of 3D primitive figures from various directions in order to confirm whether this system is applicable to virtual objects of various shapes.

In the case of an object behind

Since only the thumb comes to front side than Marker C, the blue point group follows only the thumb (see Fig. 11) and draw the thumb properly on the front of the virtual object (see Fig. 18).

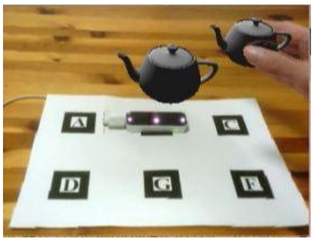

Fig. 18. Result at the position of C in the AR marker

In the case of an object in front

This system conducted the hidden surface removal considering multiple virtual objects by the Z-buffer method. The blue point group follows only the thumb and the green point group follows the other fingers.

Besides, only the point group of the thumb was drawn on the front of the virtual object by the Z-buffer method. As a result, it was possible to draw only the thumb on the front (see Fig. 19).

The blue point group and green point group in Fig. 12 are based on the depth information of the user's fingers.

Also, the anteroposterior relationship between the blue point group and the green point group is correctly displayed by the Z-buffer method.

In this way, when the thumb is positioned behind the other fingers, the green point group is displayed on the front side of the blue point group.

Fig. 19. Result at the position of G in the AR marker

IV. CONSIDERATIONN

We confirmed that this system can perform correctly hidden surface removal even when a user manipulates multiple virtual objects by Z-buffer method from the experiment result.

In addition, we also confirmed that this system is effective for virtual objects of various shapes.

The distinction between the virtual object and the user's hand was not clear when using the Z-buffer method when considering the depth between the virtual object and the user's hand.

Therefore, if there is only one virtual object or if the depth coordinates of the virtual objects are all at the same position, it is better to have blue point group follow the thumb of the user only than to have blue point group and green point group follow the palm of the user's hand.

V. CONCLUSION

AR has been applied to a wide range of fields such as education, medicine, entertainment and a guide for various tasks.

In the future, we expect further development of interaction manipulation in AR, and more detailed virtual object manipulation will be required.

Thus, we proposed and implemented hidden surface removal along the fingers, aiming at being able to apply to more detailed works about interaction manipulation in AR.

In this study, we paid attention to detection the contour of the hand to realize hidden surface removal along the user's fingers. This system conducted based on hand depth information, analysis of color information using PCA and edge detection to detect hand contours.

On the other hand, in the existing studies, there had been no precise surface treatment, handling of various hand movements, and handling of multiple virtual objects.

Therefore, we proposed and implemented hidden surface removal along the finger in order to be applicable to more detailed research on interaction operation in AR.

Furthermore, we also aimed to apply this system to multiple virtual objects.

In this study, we focus on the color of the hand to realize hidden surface removal along the user's finger, and use the depth information of the hand, color information, and edge detection using the Canny method to detect the contour of the hand. We performed processing based on.

By these processes, appropriate hidden surface removal was realized without involving other fingers.

In the evaluation experiment, it was shown that it can cope with multiple virtual objects.

REFERENCES

1. Kangdon Lee, “Augmented Reality in Education and Training”, TechTrends, vol.56, no.2, pp.13-21, 2012.
2. Carmigniani J, Furht B, Anisetti M, Ceravolo P, Damiani E, Ivkovic M, “Augmented reality technologies, systems and applications”, Multimedia Tools and Applications 51:341-77, 2011.
3. Abdelkader Bellarbi, Christophe Domingues, Samir Otmane, Samir Benbelkacem and Alain Dinis, “Underwater augmented reality game using the DOLPHYN”, Proceedings of the 18th ACM symposium on Virtual reality software and technology, ACM, 2012.
4. Buchmann V, Violich S, Billinghurst M, Cockburn A. “FingARtips – Gesture Based Direct Manipulation in Augmented Reality”, 2nd international conference on computer graphics and interactive techniques, ACM Press, pp.212–221
5. R. Katahira, M. Soga. “Development and Evaluation of a System for AR enabling Realistic Display of Gripping Motions using Leap Motion Controller”, Procedia Computer Science 60:1595–1603, 2015.
6. J. Adrián Leal-Meléndrez, Leopoldo Altamirano-Robles, and Jesus A. Gonzalez. “Occlusion Handling in Video-based Augmented Reality Using the Kinect Sensor for Indoor Registration”, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 18th Iberoamerican Congress, CIARP 2013, Proceedings, Part II, 2013.
7. J. Canny. “A Computational Approach to Edge Detection”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.8, no 6, pp.679-698, 1986.

8. Abhinandan Julka and Sandeep Bhargava. “A Static Hand Gesture Recognition Based on Local Contour Sequence”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, 2013.

AUTHORS PROFILE

Takahiro Ishizu, is a master student at Department of Computer Science and System Engineering, University of Miyazaki. His current research interests are computer graphics and image processing.

Makoto Sakamoto, received the Ph.D. degree in computer science and systems engineering from Yamaguchi University in 1999. He is presently an associate 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.

Kenji Sakoma, is a master student at Department of Computer Science and System Engineering, University of Miyazaki. His current research interests are educational research using augmented reality and virtual reality.