IEIE Transactions on Smart Processing and Computing, vol. 5, no. 4, August 2016
http://dx.doi.org/10.5573/IEIESPC.2016.5.4.267

MPEG-U–based Advanced User Interaction Interface Using Hand Posture Recognition

Gukhee Han and Haechul Choi*

Department of Multimedia Engineering, Hanbat National University / Daejeon, South Korea
guk_k@hanbat.ac.kr and choihc@hanbat.ac.kr

* Corresponding Author: Haechul Choi

Received August 18, 2016; Accepted August 28, 2016; Published August 30, 2016

* *Regular Paper*

***Abstract*:** Hand posture recognition is an important technique to enable a natural and familiar interface in the human–computer interaction (HCI) field. This paper introduces a hand posture recognition method using a depth camera. Moreover, the hand posture recognition method is incorporated with the Moving Picture Experts Group Rich Media User Interface (MPEG-U) Advanced User Interaction (AUI) Interface (MPEG-U part 2), which can provide a natural interface on a variety of devices. The proposed method initially detects positions and lengths of all fingers opened, and then recognizes the hand posture from the pose of one or two hands, as well as the number of fingers folded when a user presents a gesture representing a pattern in the AUI data format specified in MPEG-U part 2. The AUI interface represents a user's hand posture in the compliant MPEG-U schema structure. Experimental results demonstrate the performance of the hand posture recognition system and verified that the AUI interface is compatible with the MPEG-U standard.

***Keywords*:** MPEG-U, Hand and finger recognition, Advanced user interaction, User interface

1. Introduction

Recently, in the human–computer interaction (HCI) field, a lot of research has been conducted into the interactions that provide communication between humans and machines. The conventional interface methods include the switch-based method using tools (like a keyboard or a mouse), and the methods using pointing devices. But, these are not methods as natural as human communication. A natural interface method should consider the characteristics of the human communication system. Voice and gestures are the most common means of human communication. According to Hong and Lee [1], humans acquire more than 80% of their information visually; thus, a visual interface is one of the most familiar and convenient methods for humans from among the various interface methods. To enable a visual interface between a human and a machine, technology to recognize the user's intent, as represented by gestures or the posture of hands, is needed. Moreover, a practical interaction system utilizing recognition results as the natural interface should be developed, which requires compatibility with a variety of machines.

There are various studies on hand recognition. Many hand recognition methods segment the hands based on color information, such as skin tone [2-4]. However, the methods dependent on color information may be sensitive to the practical environment, such as illumination changes, and to the skin color of different races. To detect the hands more efficiently, a specific glove or hand-wrist cover was utilized [5, 6], a three-dimensional hand model was used [7], and three-dimensional image data from two-dimensional image data plus information on depth were used [8].

In order to recognize the hand's posture correctly, the fingers should also be recognized. There are various works in the literature. Malik [9] detected fingers by using the curvature between the fingers obtained from a gradient of luminance. Davatzikos and Prince [10] used convex hull–based convexity defects as a feature for finger detection. Oikonomidis et al. [11] introduced the edge-based finger phase model for probabilistic matching. Choi et al. [12] detected fingers based on the connectivity of horizontal lines within a hand region.

To provide a familiar and natural user interface, this study considers hand posture as a natural interface. The robustness of hand detection that copes with various environmental changes is realized based on three-dimensional image data, including depth information [8]. The depth information can be obtained with two or more cameras or a depth sensor. Recently, there has been an increasing number of depth cameras available at commodity prices, and there is a trend toward releasing products where a smart machine is equipped with a depth camera. Thus, the usage of the depth camera can be a more practical solution than using specific equipment, like the glove or a hand-wrist cover. As for finger detection, this study also uses convex hull–based convexity defects [10], which is robust against noise and requires relatively low computational complexity.

One of the points that should be considered in the user interface is interoperability among various machines. The conventional interface data format representing a user's hand gesture or posture is specified differently in customized software or on hardware platforms, which may not be compatible with various machines. Thus, in the ISO/IEC JTC1/SC29/WG11 Moving Picture Experts Group (MPEG), the advanced user interaction (AUI) interface data format is being standardized, and that project is called MPEG-U part 2 [13]. The purpose of MPEG-U part 2 is to support effective interoperability and interface methods by providing a data format that has consistency, defined as a standard interface [14]. In MPEG-U part 2, the above-mentioned AUI interface data format is defined based on the Extensible Markup Language (XML) schema. The design goals of XML emphasize simplicity, generality, and usability over the internet. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example, in web services. Accordingly, this study introduces an MPEG-U part 2–based AUI interface that is interoperable between different machines.

2. Proposed System

This paper introduces an MPEG-U–based AUI natural interface using hand postures. The hand posture is recognized with the Microsoft Kinect depth camera. Fig. 1 shows the proposed system, which consists of two parts: hand posture recognition and MPEG-U XML creation. Hand posture recognition is further divided into a detection module and a recognition module. In the detection module, a hand region is roughly detected by applying the hand detection algorithm provided in the Candescent natural user interface (NUI) [8]. To minimize the size of the detected hand region, the position of the wrist and the end positions of the fingers are located, and the roughly detected hand region is refined with these positions. Next, the center point of the minimum hand region is calculated for finger detection, and the end points of the fingers are detected using convex hull–based convexity defects. The recognition segment applies an algorithm that differentiates between the left hand and the right hand, and from among the thumb, index finger, middle finger, ring finger, and little finger. It differentiates between them based on the detected result, and judges the hand's posture and the fingers' spread status to recognize the final hand posture. Next, the MPEG-U XML creation segment takes the posture that has the characteristics in accordance with the MPEG-U part 2 standard and parses it into an XML document in accordance with the MPEG-U part 2 standard in order to express the recognized hand posture in an interoperable data format.

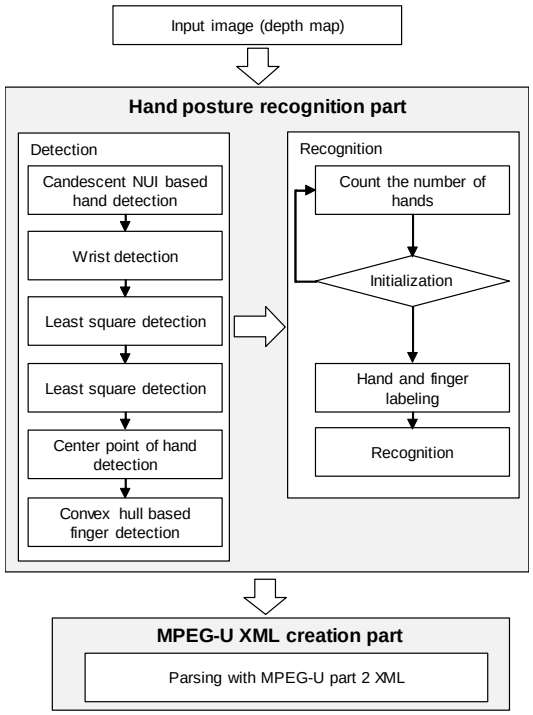


Fig. 1. The proposed hand posture recognition and MPEG-U based XML creation.

2.1 Hand Posture Recognition

2.1.1 Detection

In the Kinect, the time it takes infrared light to be shot and reflected is measured by the detection sensor in order to acquire the depth information [15]. In order to use the Kinect, 3D motion recognition middleware that enables compatibility with the PC is needed. The middleware used mainly for the Kinect is the Kinect Software Development Kit (SDK) from Microsoft and PrimeSense OpenNI. Candescent NUI provides the open source code, which recognizes the hand and the fingers by utilizing the middleware. Candescent NUI has the advantage of being able to detect the hand precisely and swiftly, utilizing three-dimensional image data, and is robust to changes in complex background and illumination; computational complexity is also very low. Hand and finger recognition in the proposed system uses the open source code of Candescent NUI [8] for depth information input from the Kinect in order to detect the hand region, as seen in Fig. 2(a). In order to detect the minimum hand region from among the overall detected hand regions, the following processes are performed.

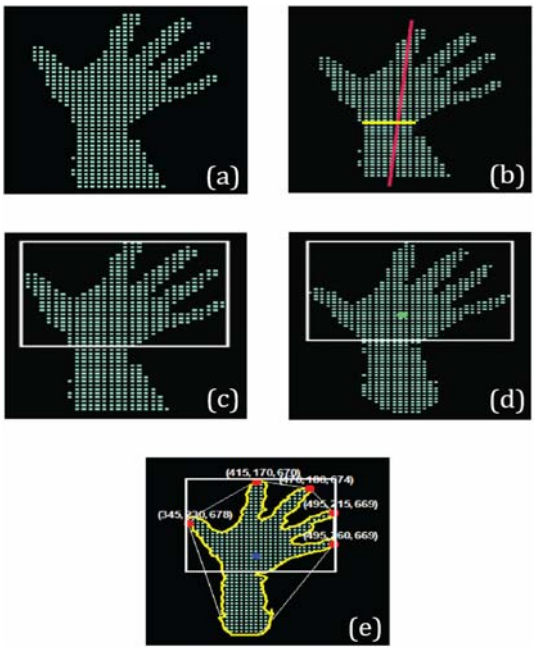


Fig. 2. Results of hand and finger detection (a) hand detection using the Candescent NUI, (b) wrist detection by applying the line fitting algorithm, (c) hand region detection by least squares, (d) the center of mass of the hand by applying the distance transform algorithm, (e) finger detection by using the convex hull–based convexity defects.

First, the hand region is fitted to a line conforming to the minimum least square error [16]. Then, pairs of two intersection points between the outermost line of the hand and the normal line of the fitting line are found. When the variance of the distance between two pairs of points is constant, the normal line through the two pairs of points is decided upon as the wrist line. Fig. 2(b) indicates the point of the wrist by indicating one line that represents the overall hand region. After finding the wrist, calculate the least square including the minimum hand region from the wrist point to the ends of the fingers. Fig. 2(c) indicates the least square including the minimum hand region from the wrist to the end points of the fingers. Next, find the center point of the minimum hand region by applying a distance transform algorithm [17]. The center point is necessary for the detection and recognition of the fingers afterwards, and is often utilized as the position value of the hand in the MPEG-U AUI interface data format. Fig. 2(d) indicates the center point of the minimum hand region to which the distance conversion algorithm applies. Next, detect the ends of the fingers using convex hull–based convexity defects [10]. This method has the advantage of assuming the correct fingers and obtaining directional information on the corresponding fingers, but there is a disadvantage in that it needs an additional post-processing procedure as follows. Assess the goodness of fit with the oval approximation model as improved, based on the oval morphological characteristics of the finger joints. Based on the results above, detect the final end points of the fingers by using information on the vector angle difference between the end points of each finger and the center point.

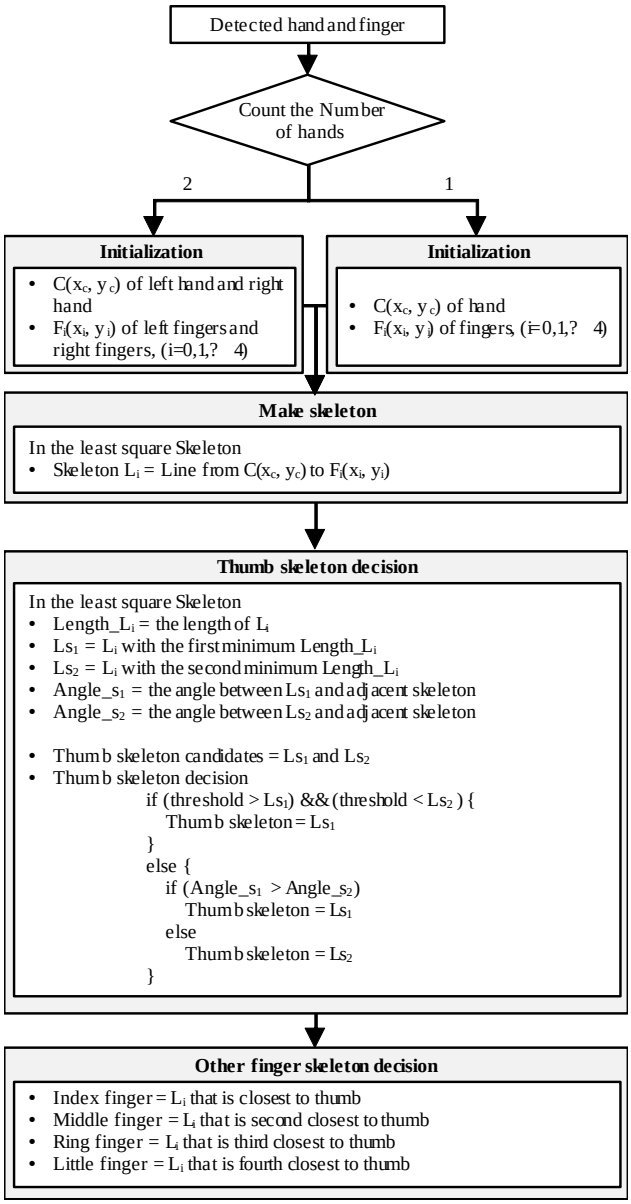


Fig. 3. Proposed finger skeleton classification algorithm.

In Fig. 2(e), the red dots are the final, detected end points of the fingers.

2.1.2 Recognition

The recognition process classifies the right and left hands, along with the thumb, index finger, middle finger, ring finger, and the little finger. It is based on the detected hand and fingers. The classification of the hand is performed considering situations where the minimum hand region is either one hand or both hands. If it is both hands, by the topology judgment, what is to the left is classified as the left hand, and what is to the right is classified as the right hand. Finger classification applies the proposed finger skeleton classification algorithm, as seen in Fig. 3. First, the number of regions detected in the hand is calculated. When the number of hand regions is 2, the number of center points and the number of end points of fingers are initialized at 2 and 10, respectively. When the

number of hand regions is 1, the numbers of the respective points are initialized at 1 and 5. Next, a skeleton of the hand is made as follows. When the center point is $C(x_c,y_c)$ and the end points of each finger are $F_i(x_i,y_i)$, (i=0~4), the skeleton of each finger L_i will be defined as the line that connects C and F_i. The length is defined as Length_L_i. The two shortest skeletons from among the Length_L_i of each skeleton are found, and one thumb candidate skeleton is found, called Ls_1, and another possible thumb candidate skeleton is called Ls_2. The skeleton for which the value is lower than the threshold value from among the thumb candidate skeletons is selected. Then, it is called the optimal thumb's skeleton. And when the value is lower than or greater than the critical value, it is called the optimal thumb's skeleton by using the angle among the adjacent skeletons. When we set the angle from among the adjacent skeletons of Ls_1 as Angle_s_1 and set the angle among the adjacent skeleton of Ls_2 as Angle_s_2, the skeleton is classified, which has the maximum angle. After classification of the thumb's skeleton, the rest of the fingers in the order most adjacent to the thumb are classified as the skeletons of the index finger, the middle finger, the ring finger, and the little finger. After that, by using the classified fingers' skeletons, if the end points of the fingers lie on the corresponding skeleton or lie adjacent to the corresponding skeleton, that end point of the finger is classified as the end point of the finger of that corresponding skeleton. For real-time application, a new skeleton is not created on each input, and only the detection procedure is executed and compared with the acquired least square from the previous point of least squares. By calculating the coefficient of movement and rotation conversion and applying this to the skeleton created at the previous point, we decrease the complexity for creation of the skeletons. Next, the final hand posture is recognized by judging the hand posture and whether each finger is spread.

2.2 MPEG-U XML

The process for MPEG-U XML creates the XML document in accordance with the MPEG-U part 2 standard's schema structure. MPEG-U part 2 supports the schema structure for all data types as an XSD document. If someone has scissors in a recognized hand and fingers, the MPEG-U XML process creates an XML document using the scissors-type schema structure. The information about the scissors type is included in the document, and the parameters processed for hand and fingers recognition are divided into the X, Y, and Z coordinates in accordance with the MPEG-U part 2 standard and is narrated. Fig. 4 was parsed in accordance with the MPEG-U part 2 standard for a posture that has the characteristics that correspond to the MPEG-U part 2 standard.

3. Experiment

This section defines the end points of the fingers using convex hull–based convexity defects through an experiment, and shows the results from recognition using

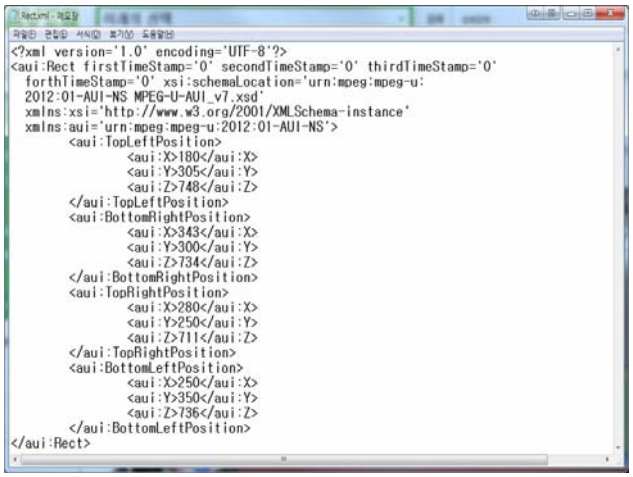


Fig. 4. Rect XML of MPEG-U part 2 geometric pattern parsed when a user takes the Rect posture.

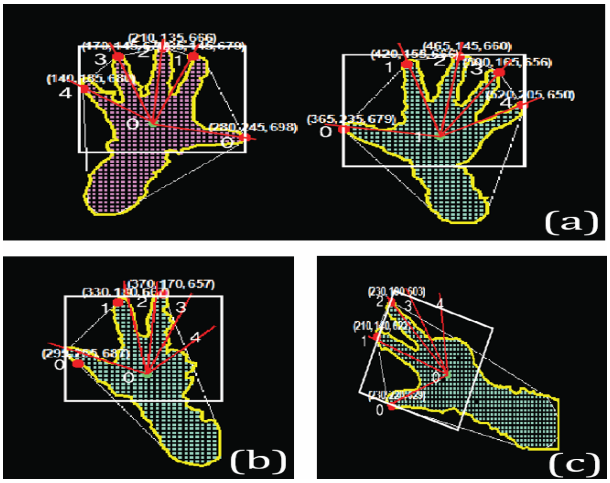


Fig. 5. Hand posture recognition result of the proposed method (a) result when all 10 fingers are spread out, (b) result when a few fingers are unfolded, (c) result under translation and rotation of the hand.

the proposed finger classification algorithm. Also, it validates through experiments the points where hand posture recognition is possible in various environments. Finally, it validates the suitability of the MPEG-U–based advanced user interface for interconnection by experimenting with the proposed system and MPEG-U part 2 reference software.

3.1 Proposed Finger Classification Algorithm Verification

The proposed system recognizes hands as either the left or right hand and fingers as the thumb, index finger, middle finger, ring finger, and little finger. The posture is robustly recognized in the experiment when all 10 fingers are spread, or if only a few of them are spread, or when they move or rotate. Fig. 5(a) is the test result that shows the hand and fingers when all 10 fingers are spread, and Fig. 5(b) is the test result that shows the hand and the

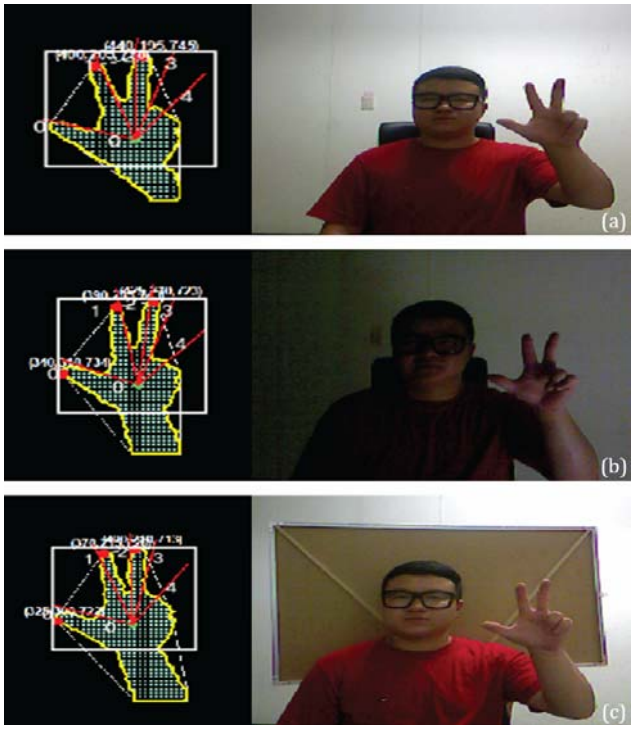


Fig. 6. (a) Recognition result of the hand posture under a bright background, (b) recognition result of the hand posture under a dark background, (c) recognition result of the hand posture under a background having a color similar to the skin.

fingers when only a few fingers are spread. Fig. 5(c) is the test result that classifies the hand and the fingers when the user assumes a posture with change of movement and rotation. In the figure, the 0 and 1 of the hand region indicate the left and right hands, and 0 to 4 in the adjacency of the fingers indicates each thumb, index finger, middle finger, ring finger, and little finger.

3.2 Hand Posture Recognition Verification in a Variety of Environments

The proposed system uses three-dimensional image data, and thus, is robust to various environments. The Candescent NUI, which detects the hand utilizing three-dimensional image data, is robust to various environmental changes, and computational complexity is low. In the experiment, the system shows the performance of the proposed method with a background where the color is similar to the skin color and there is change in illumination. Fig. 6(a) shows the hand recognition result with a bright background and high brightness value, and Fig. 6(b) shows the hand recognition result recognized with a dark background and low brightness value. Fig. 6(c) shows the hand recognition result where the background has a color similar to the skin color. As shown in the figure, this proposed method is robust to the changes in the environment and illumination.

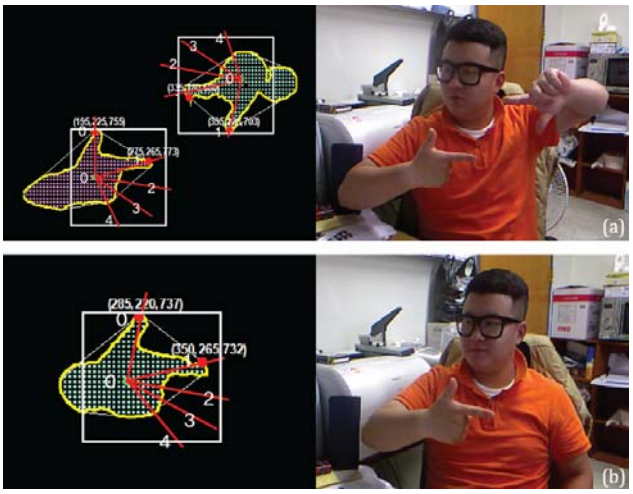
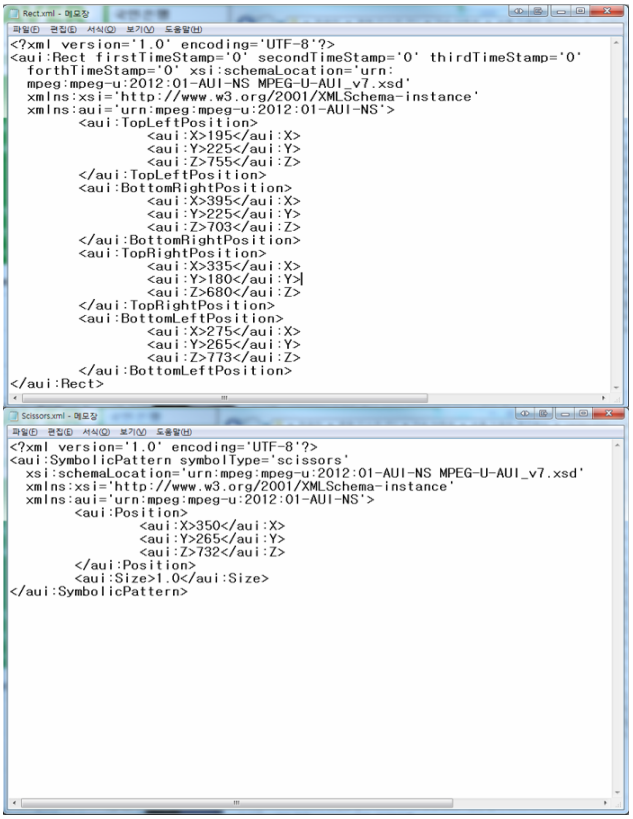


Fig. 7. (a) Hand posture of Rect, (b) Hand posture of Scissors.



```
<?xml version='1.0' encoding='UTF-8'?>
<aui:Rect firstTimeStamp='0' secondTimeStamp='0' thirdTimeStamp='0'
  forthTimeStamp='0' xsi:schemaLocation='urn:
  mpeg:mpeg-u:2012:01-AUI-NS MPEG-U-AUI_v7.xsd'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:aui='urn:mpeg:mpeg-u:2012:01-AUI-NS'>
        <aui:TopLeftPosition>
                <aui:X>195</aui:X>
                <aui:Y>225</aui:Y>
                <aui:Z>755</aui:Z>
        </aui:TopLeftPosition>
        <aui:BottomRightPosition>
                <aui:X>395</aui:X>
                <aui:Y>225</aui:Y>
                <aui:Z>703</aui:Z>
        </aui:BottomRightPosition>
        <aui:TopRightPosition>
                <aui:X>335</aui:X>
                <aui:Y>180</aui:Y>
                <aui:Z>680</aui:Z>
        </aui:TopRightPosition>
        <aui:BottomLeftPosition>
                <aui:X>275</aui:X>
                <aui:Y>265</aui:Y>
                <aui:Z>773</aui:Z>
        </aui:BottomLeftPosition>
</aui:Rect>
```

```
<?xml version='1.0' encoding='UTF-8'?>
<aui:SymbolicPattern symbolType='scissors'
  xsi:schemaLocation='urn:mpeg:mpeg-u:2012:01-AUI-NS MPEG-U-AUI_v7.xsd'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:aui='urn:mpeg:mpeg-u:2012:01-AUI-NS'>
        <aui:Position>
                <aui:X>350</aui:X>
                <aui:Y>265</aui:Y>
                <aui:Z>732</aui:Z>
        </aui:Position>
        <aui:Size>1.0</aui:Size>
</aui:SymbolicPattern>
```

Fig. 8. XML document in accordance with the MPEG-U part 2 standard (a) Rect, (b) Scissors.

3.3 MPEG-U–based Advanced User Interaction Interface Verification

We verified the proposed system by experimenting with geometric pattern and symbolic pattern types from among the six different patterns defined in MPEG-U part 2. The types used in the experiment were Rect of Geometric Pattern and Scissors of Symbolic Pattern. Rect of Geometric Pattern needs the data for the top left position, bottom left position, top right position, and the bottom right position; Scissors of Symbolic Pattern needs the size

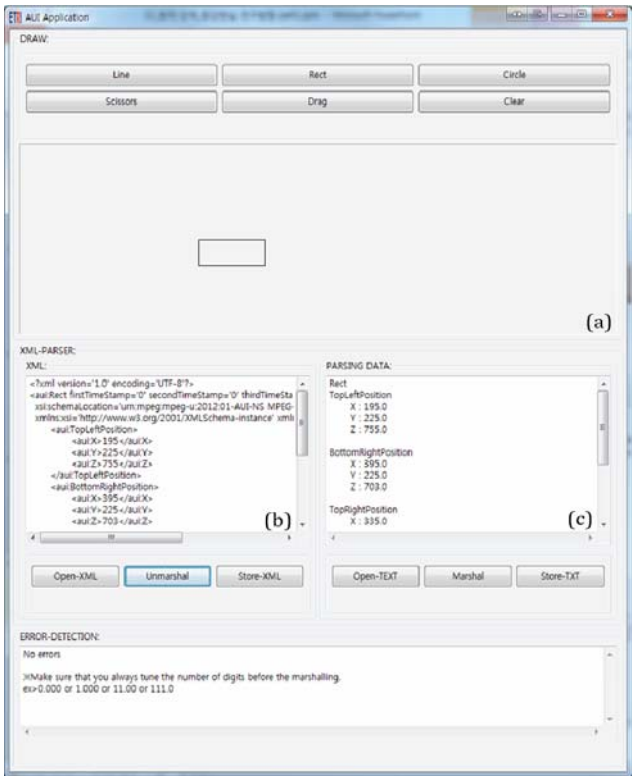


Fig. 9. X-Y coordinate output and GUI output from the XML document for Rect.

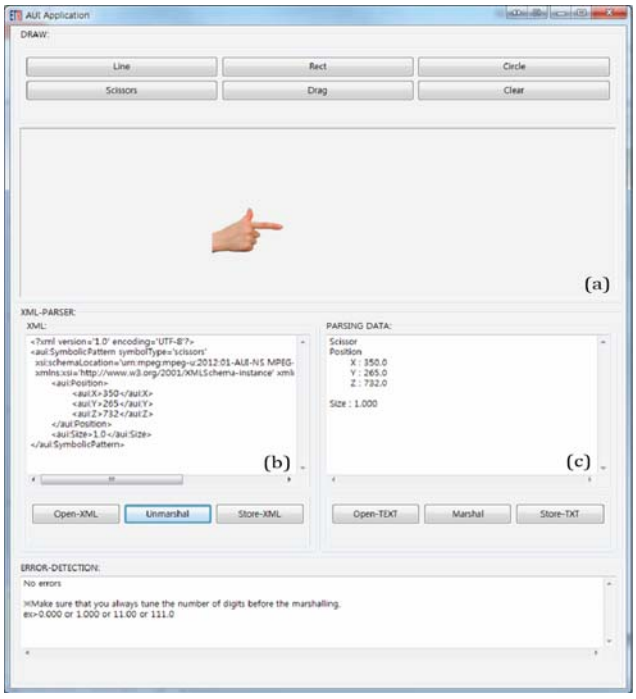


Fig. 10. X-Y coordinate output and GUI output from the XML document of Scissors.

data. In the proposed system, in order to express Rect and Scissors, in Rect, the classified thumbs and index fingers of both hands are used, and in Scissors, the thumb and index finger of one hand and the size of the hand are used. Fig. 7 expresses Rect and Scissors with the recognized hand postures. Fig. 8 shows Rect's posture information and Scissor's posture information, parsed into an XML document in accordance with the MPEG-U part 2 standard. In order to verify the validity of the parsed XML document, verification was done using the MPEG-U part 2 reference software. First, we conveyed the parsed XML document to the MPEG-U part 2 reference software, as shown in Figs. 9(a) and 10(a). After that, the corresponding coordinate values were detected, as seen in Figs. 9(b) and 10(b) through the unmarshal task, printing the detected coordinate values to the GUI of the user interaction interface, as seen in Figs. 9(c) and 10(c). The XML document, in accordance with the MPEG-U standard, can be conveyed like this to a different device, and that different machine can obtain the same result if it is equipped with an XML parser that suits the standard.

4. Conclusions

The system proposed in this thesis is the MPEG-U–based advanced user interface, which works via hand posture recognition using a depth camera. The hands and the fingers were more robustly recognized using the depth camera; the proposed algorithm to classify the fingers was robust against changes in various movements and rotations, and it could also robustly classify folding and unfolding actions. Also, the different scene techniques were interoperable under the standards based on MPEG-U. We also showed a standards-based interface between a user interface device and a scene technique device. The proposed system can afterwards be utilized as a modality in a multi-modal interface. Also, if the development of technology utilizing MPEG-U is promoted for smart TV and smart phones, more varied interfaces will be realized.

Acknowledgement

This work was financially supported by Hanbat National University.

References

[1] Seok-Ju Hong and Chil-Woo Lee, "Human-Computer Interaction Survey for Intelligent Robot," The Korea Contents Society, Vol. 4, No. 2, pp. 507-511, Feb. 2006.

[2] Anastasios Roussos, Stavros Theodorakis, Vassilis Pitsikalis and Petros Maragos, "Hand tracking and affine shape-appearance handshape subunits in continuous sign language recognition," ECCV Workshop on Sign, Gesture and Activity, Hersonissos, Crete, Greece, Sep. 2010.

[3] Yuh-Rau Wang, Wei-Hung Lin, and Ling Yang, "A novel real time hand detection based on skin-color," Consumer Electronics (ISCE), IEEE 17th International Symposium on, pp. 141~142, Jun. 2013.

[4] Xintao Li, Can Tang, Chun Gong, Sheng Cheng and Jianwei Zhang, "Hand Segmentation Based on Skin

Tone and Motion Detection with Complex Backgrounds," Chinese Intelligent Automation Conference, Springer Berlin Heidelberg, Vol. 256, pp. 105~111, Jan. 2013. Article (CrossRef Link)
[5] Robert Y. Wang and Jovan Popovi' , "Real-Time Hand-Tracking with a Color Glove," ACM Transactions on Graphics, Vol. 28, Issue. 3, No. 63, Aug. 2009.
[6] R. Lockton and A. Fitzgibbon, "Real-time gesture recognition using deterministic boosting," BMVC, pp. 1~10, Sep. 2002.
[7] V. Argyros and S. Sclaroff, "Database indexing methods for 3D hand pose estimation," Gesture Workshop, pp. 288~299. Apr. 2003.
[8] Candescent NUI Samples & Source code http://candescentnui.codeplex.com/SourceControl/latest#CCT.NUI.Visual/ClusterLayer.cs.
[9] S. Malik, "Real-time hand tracking and finger tracking for interaction," CSC2503F Project Report, Department of Computer Science, University of Toronto, Dec. 2003.
[10] C. Davatzikos and J. L. Prince, "Convexity analysis of active contour problems," Image Vision Computing, Vol. 17, pp. 27~36, Jan. 1999. Article (CrossRef Link)
[11] I. Oikonomidis, N. Kyriazis and AA. Argyros, "Efficient Model-based 3D Tracking of Hand Articulations using Kinect," BMVC, pp. 101.1~101.11, Sep. 2011.
[12] Junyeong Choi, Hanhoon Park and Jong-ll Park, "Hand shape recognition using distance transform and shape decomposition," Image Processing(ICIP), pp. 3605~3608, Sept. 2011.
[13] Information technology - Rich media user interfaces - Part 2: Advanced user interaction (AUI) interfaces. - ISO/IEC 23007, Feb. 2012.
[14] Gukhee Hand, A-Ram Baek, Haechul Choi, "MPEG-U part2 based advanced user interaction interface system," The Korea Contents Association Journal, Vol. 12, No. 12, pp. 54~62, Dec. 2012. Article (CrossRef Link)
[15] KINECT. http://en.wikipedia.org/wiki/Kinect.
[16] Sara Taskinen and David I, "Robust estimation and inference for bivariate line ‐ fitting in allometry," Biometrical Journal, pp. 652-672, Jun. 2011. Article (CrossRef Link)
[17] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Distance Transforms of Sampled Functions," Theory of Computing, Vol. 8, pp. 415-428, Sep. 2012. Article (CrossRef Link)

Gukhee Han received a BSc and MSc from the department of multimedia engineering at Hanbat National University, Daegu, Korea, in 2012 and 2014, respectively. His research interests include image processing, computer vision, and human computer interaction.

Haechul Choi received his BSc in electronics engineering from Kyung-pook National University, Daegu, Korea, in 1997, and his MSc and PhD in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1999 and 2004, respectively. He is an associate professor in the Division of Information Communication & Computer Engineering at Hanbat National University, Daejeon, Korea. From 2004 to 2010, he was a Senior Member of the Research Staff in the Broadcasting Media Research Group of the Electronics and Telecommunications Research Institute (ETRI). His current research interests include image processing, video coding, and video transmission.