

Detection and Recognition of Hand Gesture for Wearable Applications in IoMT

Anna YANG*, Sung Moon CHUN**, and Jae-Gon KIM*

*School of Electronics and Information Engineering, Korea Aerospace University, Korea

**Insignal, Korea

nayang@kau.kr, smchun@insignal.co.kr, jgkim@kau.ac.kr

Abstract— To support an efficient media consumption in wearable and IoT (Internet of Things) environments, the standardization of IoMT (Internet of Media-Things) is in the progress in MPEG (Moving Picture Experts Group). In this paper, we present a method to detect and to recognize hand gestures for generating hand gesture-based commands to control the media consumption in smart glasses. First, we present a detection method that utilizes depth image obtained by incoming stereo image sequences and skin color information in a combined way. Secondly, we are going to present the representation of detected hand contours based on Bézier curve as metadata to provide an interoperable interface between a detection module and a recognition module in an IoMT framework. In addition, the comparison with existing standard tools that can be used for hand gesture representation is given. In the recognition module, the detected hand contour is reconstructed by parsing delivered metadata. A set of hand gestures featured with diverse combination of open fingers and rotational angles is used for the hand gesture recognition in the proposed recognition method. Finally, the recognized hand gesture is mapped into one of the pre-defined set of gesture commands. Experiment results show that the proposed method gives quite stable performance of detection and recognition of hand gesture along with interoperable interface between both processing modules.

Keyword— MPEG Internet of Media-Things (IoMT), Smart Glasses, Hand Gesture Recognition, Hand Gesture Detection, Bézier curve

I. INTRODUCTION

RECENTLY, the standardization of Internet of Media-Things (IoMT) is undergoing in MPEG to support efficient media consumption in wearable and IoT (Internet of Things) environments [1]. The normative parts of the IoMT are mainly the specification of API (Application Programming Interface) and the representation of

information to be delivered over the interfaces between Media Things (called as MThings) that can be wearable/IoT devices or processing units in media-centric IoT and wearable applications. On the other hand, as hand gestures are attracting more attention as NUI (Natural User Interface) of wearable devices such as smart glasses, efficient detection and recognition of hand gestures is strongly required [2]. In this paper, we propose a hand gesture detection and recognition method for controlling the media consumption and/or wearable device itself. In addition, the XML (eXtensible Markup Language)-based description of hand gesture is presented to support interoperable interfaces in the context of IoMT standard. In the use cases of gesture-based wearable applications in IoMT [1], it is likely required that general hand gestures having any contour and/or trajectory should be described in an interoperable way in terms of two aspects: 1) hand gestures to be mapped into commands that are used for interactions between a user and a wearable device should be general enough to support diverse potential use cases; 2) detection of hand gestures and their recognition can be done separately in different processing units (PUs) to cope with the limitation of computational power of the wearable device.

We adopt a simple method of gesture detection that combines depth map and color image acquired from the incoming stereo image sequence, and represents the detected hand contour based on a Bézier curve as XML-based metadata. The detected hand contour delivered as a form of metadata is reconstructed in a recognition module. The reconstructed hand contour is recognized as one of the set of hand gestures predefined in a given application. Diverse types of hand gestures with different open fingers and hand angles can be identified mainly based on convexity defects of hand contour. A post processing is applied on the recognition results of consecutive frames in temporal domain to obtain more reliable results. In addition, the results of comparison with the existing MPEG standards that can be used for description of hand gesture are presented.

The rest of the paper is organized as follows. In Section II, we present an introduction of MPEG IoMT and gesture-based wearable application scenarios considered in IoMT. A gesture detection method is presented in Section III. A method of hand contour description using Bézier curve along with the comparison with the existing MPEG standard is presented in Section IV. In Section V, a hand gesture recognition method using the described hand contour is presented, and experimental results are presented in Section VI. Finally, the conclusions are given in Section VII.

Manuscript received Aug. 23, 2017. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by Korean Government (MSIT) (R0127-15-1015).

Anna Yang and is with School of Electronics and Information Engineering, Korea Aerospace University, Goyang, Korea (nayang@kau.kr).

Sung Moon Chun is with Insignal, Seoul, Korea (e-mail: smchun@insignal.co.kr).

Jae-Gon is with School of Electronics and Information Engineering, Korea Aerospace University, Goyang, Korea (corresponding author to provide phone: +82-2-300-0414; e-mail: jgkim@kau.ac.kr).

II. GESTURE-BASED WEARABLE APPLICATION IN IoMT

Gesture-based smart glasses applications are considered as one of the typical wearable use cases in IoMT [1], [3]. When a user wears smart glasses, hand gesture is becoming a promising user interface that allows a user to use both hands without operating an input device such as keyboard or mouse, etc. In such applications, in order for a user to control efficiently media consumption, the hand gesture recognition should be supported in an IoMT framework.

As shown in Fig. 1, user (user), MThing, and processing unit (PU) are key functional components of an IoMT framework. Media Thing is specified as a Thing capable of sensing, acquiring, actuating, or processing of media or metadata [1]. Smart glasses and processing unit are kinds of MThing. In order to support various types of hand gesture-based use cases including the control of media consumption in a wearable device and/or device itself, hand gestures which include static ones or dynamic ones such as a trajectory of hand motion should be recognized into a user invoked command.

As shown in Fig. 1, the incoming stereo image sequence acquisition by a stereo camera mounted on the smart glasses is transmitted to a PU, in which user's gesture is detected (① in Fig. 1). It is assumed that gesture detection and gesture recognition are more likely to be performed in separate PUs with the considerations as follows: 1) The detected hand contour or trajectory can be recognized as one of various types of gesture to be mapped into a command depending on the given application rather than limiting the gesture types, and/or 2) the computational power of a PU may not be sufficient to perform both processings of detection and recognition in a wearable environment [4], [5].

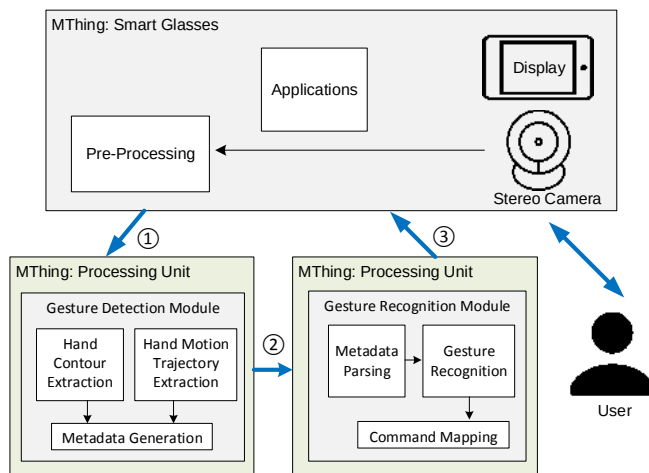


Fig. 1. A framework of IoMT for supporting use cases of hand gesture based wearable applications

Under this assumption, we need to specify a normative interface between both PUs performing detection and recognition, respectively. In aspect of standard, we focus on the representation of the detected hand contour to be delivered to the recognition module (② in Fig. 1). In other words, a hand region is detected and its contour is described as metadata in the hand detection module. Then, in the hand gesture recognition module, the detected hand contour is reconstructed by parsing the metadata delivered from the detection module. Finally, the reconstructed contour is

mapped into a user command. The gesture-based commands are predefined in many different ways according to the given application.

The generated command controls the wearable device and/or the application software running on the wearable device (③ in Fig. 1). In this way, the use case of efficient media consumption in a wearable device is supported based on hand gestures.

III. HAND GESTURE DETECTION

Fig. 2 illustrates the overall procedure of hand gesture detection that combines depth map and color image extracted from the incoming stereo image sequence. A depth map is generated by stereo matching of the incoming stereo images. Due to the nature of the smart glasses, it is assumed that a user's hand pose is given at a certain short distance range (30cm ~ 50cm) from the camera. Based on this assumption, a rough hand region is simply obtained by thresholding the value of the depth map. Then, the detected rough hand region is refined by using color information and morphological filtering, which results in more accurate hand contour.

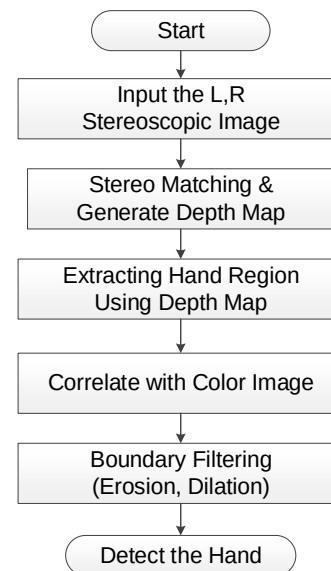


Fig. 2. Overall procedure of hand gesture detection

IV. HAND GESTURE DESCRIPTION

As mentioned before, the detected hand contour to be delivered to a recognition module should be represented in an interoperable way to support diverse types of hand gestures in an IoMT framework. In this paper, we present an XML schema for describing hand contour based on Bézier curve. In addition, we briefly introduce the existing MPEG tools of MPEG-U and MPEG-7 Curvature Scale-Space (CSS) that can be used for hand gesture description, and present the comparison results between the MPEG-7 CSS and the Bézier curve based description in terms of several aspects.

A. Bézier Curve Based Description

As shown in Fig. 3, a hand contour consists of a set of consecutive curves each of which is represented by using n th-order Bézier curve given by Eq. (1). The n th-order Bézier curve is expressed by $(n + 1)$ points, which consist of a starting point, an end point, and a few control points that

determine the curvature of the curve. In this paper, we use a cubic Bézier curve with the order three, so there are two control points P_1, P_2 in a curve representation as shown in Fig. 4.

Fig. 3 shows an example of the result of this representation process. The given hand contour consists of a set of curves each of which is expressed by four Bezier points in the case of cubic Bézier curve. For example, four Bézier points are shown in Fig. 3 (b) (starting point, two control points, and end point are given as $(P_{0x}, P_{0y}), (P_{1x}, P_{1y}), (P_{2x}, P_{2y}),$ and (P_{3x}, P_{3y}) , respectively). As a result, a list of Bezier points representing a set of consecutive curves corresponding to the detected hand contour is obtained as shown in Fig. 3 (a).

$$Bezier(n, t) = f(t) = \sum_{i=0}^n \binom{n}{i} \cdot (1-t)^{n-i} \cdot t^i \quad (1)$$

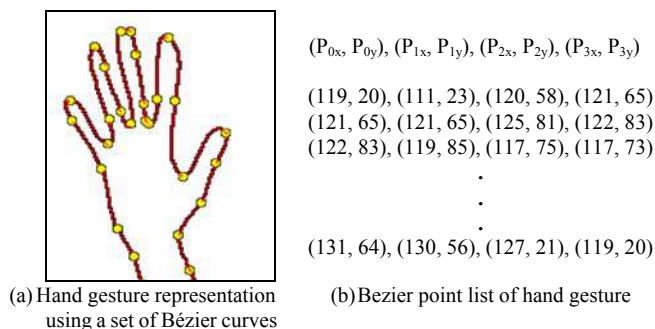


Fig. 3. An example of Bézier curve representation

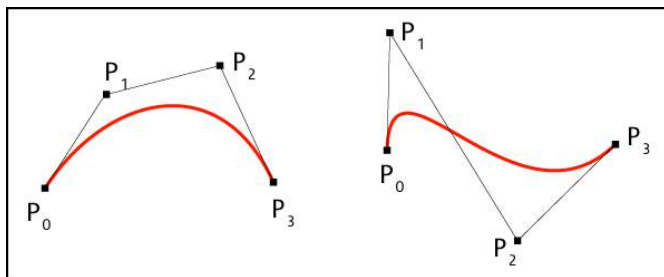


Fig. 4. The 3rd-order Bézier curve representation using four points (P_0 : start point P_1, P_2 : control points P_3 : end point)

Fig. 5 shows an algorithm for representing a hand contour using a set of Bézier curves. To draw a given hand contour using a set of Bézier curves, the whole contour is iteratively divided until the fitting error between the Bézier curve based represented curve and the given contour is less the allowed error predefined. In other words, at the beginning, the entire hand contour is represented by a single Bézier curve. Then, if the fitting error using a Bézier curve is larger than a preset threshold (max_error), then the given contour is divided into two and each curve is represented by a Bézier curve. This contour dividing and representation process is repeated until the required accuracy is met. The fitting error can be defined as the distance between two curves. If there are less than four points left on the contour, then joining the remaining points are not bitted by a Bézier curve. As a result, the accuracy of contour representation is adjusted by setting the value of maximum fitting error, and the number of Bézier curve is determined accordingly.

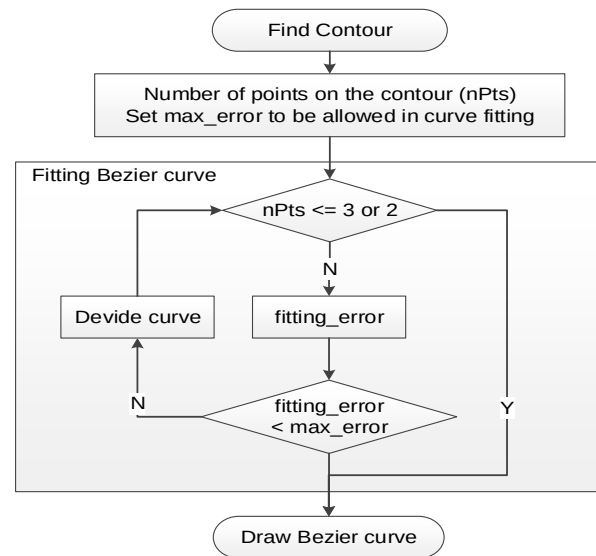


Fig. 5. A procedure of representation for Bézier Curve based hand gesture

To describe a list of Bézier points corresponding to the detected hand contour as XML-formatted metadata, we design a metadata schema as shown in Fig. 6. The detected hand contour can be reconstructed by parsing the delivered metadata in the recognition module. Fig. 7 shows an example of the reconstructed hand contour from the metadata.

The types of hand gesture can be static or dynamic. In the case of dynamic type, a trajectory of hand motion can be mapped into a user-invoked command. As shown in Fig. 6, the HandGestureType allows us to describe static gesture or dynamic gesture by using the element of HandContour or HandTrajectory, respectively. A hand contour consists of a set of consecutive curves each of which is represented by a Bézier curve. The set of consecutive curves are described by the element of GroupBezierCurve. A Bezier curve is represented by a start point, end point, and control points. The end point of a curve is identical to the start point of the following curve since a set of curves are successively connected.

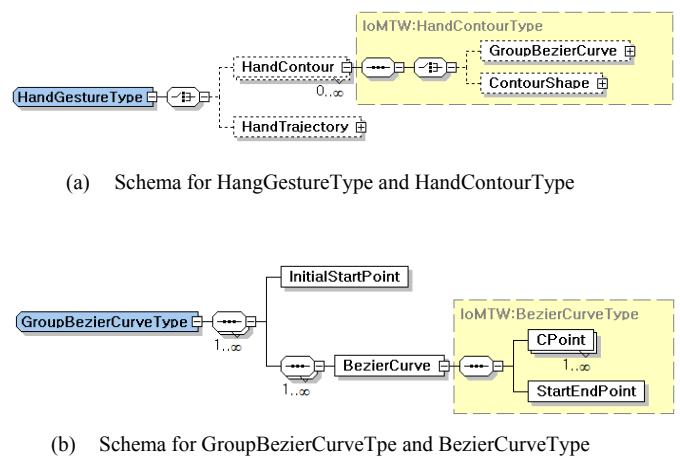


Fig. 6. Schema for hand gesture representation using Bézier curve

Therefore, as shown in Fig. 6 (b) of GroupBezierType, the InitialStartPoint describes the start point of the first curve,

and each of the consecutive curve is described by using the CPoints (control points) and the element of StartEndPoint. In this way, redundant description of the same point which is an end point and a start point of successive curves is avoided.

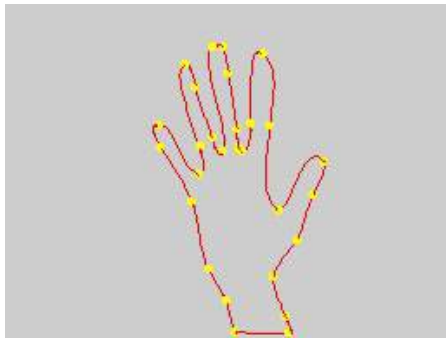


Fig. 7. Reconstructed hand contour by parsing the delivered metadata

B. Comparison with the existing MPEG Tools

There are few tools that can be used to describe hand gesture or hand contour in the MPEG such MPEG-7 CSS and MPEG-U Part 2. In this sub-section, we present the review of the existing such tools, and present the comparison results between MPEG-7 CSS and Bézier curve based description of hand contour in the context of IoMT.

MPEG-U Part 2 specifies Advanced User Interaction (AUI) interface to enhance interaction between scene descriptions and system resources [6]. In the data format of the AUI interface specified in MPEG-U Part 2 as shown in Table 1, a set of hand postures and gesture patterns have been specified to support the intuitive hand-based interaction for scene description.

TABLE I
DATA FORMAT OF AUI INTERFACE SPECIFIED BY MPEG-U PART 2

Pattern	Type
Geometric	Point, Line, Rect, Arc, Circle
Symbolic	Victory, Heart, Rock, Scissors, Paper, Okay
Hand Posture	OpenPalm, Fist, Pointing, Thumb-Up, Thumb-Down, Grap
Hand Gesture	Push, Pull, Slap, Slap_right, Slap_top, Slap_bottom, Circle_clockwise, Circle_anti-clockwise, waving, check
Touch	Tap, Double Tab, Press, Dragg, Rotate, Flick
Composition	

MPEG-7 specifies a set of visual descriptors to describe various visual features mainly focusing on content-based indexing and retrieval applications [7]. In MPEG-7, the shape descriptor of the curvature scale space (CSS) is directly related to the description of hand contour and hand motion trajectory, respectively

The MPEG-7 CSS descriptor which is a contour based shape descriptor treats shape boundary as a 1D signal, and analyzes this 1D signal in a scale space [7], [8]. By examining zero crossings of curvature at different scales, the concavities/convexities of shape contour are found. These concavities/convexities are useful for shape description because they represent the perceptual features of shape contour.

The existing MPEG-U Part 2 and MPEG-7 CSS may not be enough for describing hand gesture/contour in the context of IoMT as follows. An environment of hand gesture recognition in smart glasses is slightly different with that is considered in MPEG-U in terms of camera view and camera location. A smart glass is worn on a user's face and hand posture is acquisitioned within a short distance range while it is assumed that the viewpoint of the camera is located on the user's front torso in MPEG-U. It is likely that more rich and elaborate gestures are used in smart glasses to support various hand gesture based use cases. However, a set of patterns of hand posture and hand gesture defined in MPEG-U Part 2 are very limited.

The MPEG-7 CSS is a shape descriptor mainly focusing on the applications of content-based indexing and retrieval, therefore it has a few limitations in in the context of IoMT as follows. The characteristic of rotation invariant representation of the CSS would be a shortcoming since hand postures with different angle could have different meanings in the description of hand contour. In addition, the detected hand contour should be reconstructed from the description to be recognized, which is not supported by CSS. As shown in the comparison results of Table 2, the Bézier curve based description meets all functionalities required in the hand gesture based wearable applications in IoMT.

TABLE II
COMPARISON BETWEEN MPEG-7 CSS, MPEG-U PART 2, AND BÉZIER CURVE BASED DESCRIPTION TOOLS IN TERMS OF FUNCTIONALITIES

Key Functionalities	MPEG-7 CSS	MPEG-U Part 2	Beizer Curve
Differentiation of rotation	X	O	O
Applicability of diverse recognition methods	X	O	O
Extensibility of gesture types	O	X	O
Reconstructability	X	–	O

In terms of hand contour description, we present the details on the experiment of comparison between MPEG-7 CSS and Bézier curve based description in Table 3. The results shows that both descriptors give similar description performance in terms of complexity and efficiency (description size). However, only the Bézier curve based descriptor allows the reconstruction of contour from the description as well as scalable description.

V. HAND GESTURE RECOGNITION

Fig. 8 shows the overall procedure of hand gesture recognition. Delivered metadata describing the detected contour is parsed into a list of Bezier points from which the detected contour is reconstructed. In order to recognize the hand gesture from the reconstructed hand contour, as an initial step, we generate a reference mask from the contour of open palm of a user. A reference mask gives initial values of features that are extracted and used as reference values for the subsequent recognition step such as convexity defects, degree of rotation, and gravity center.

TABLE III
RESULTS ON THE COMPARISON BETWEEN MPEG-7 CSS AND BÉZIER CURVE
BASED DESCRIPTION

Items	Bézier curve based descriptor	MPEG-7 CSS descriptor	Comment
Applicability	Contour of the object Moving trajectory of the object	Contour of Object	
Scalability	Scalable in terms of description size and contour accuracy	Not available	Possible to trade off contour accuracy and description size
Reconstruction	Available	Not available	
Computational Performance	Average ~ 5msec	Average ~ 20msec	Computational time for generating descriptor from hand contour
Description Size	Hand contour: Average 95Byte Hand trajectory: Average 36Byte	Average 15Bytes	Both cases are very light weight
Main Applications	General contour description, transmission and reconstruction	Shape similarity matching and indexing	

Using convex hull, we can obtain some defects which allow us to figure out the position of each finger, and the gradient moment is used to obtain the center of gravity and degree of rotation. The gradient of moment $\mu_{p,q}$ and the degree of rotation θ are obtained from the palm region by using Eg. (2) and Eg. (3), respectively. In order to obtain palm region, the finger region is removed from the hand region by morphological operations. x and y mean a coordinate of pixels, $\bar{x}$ and $\bar{y}$ mean a mean of x and y in the palm region, respectively.

$$\mu_{p,q} = \sum_{(x,y) \in R} (x - \bar{x})^p (y - \bar{y})^q \quad (2)$$

$$\theta = \frac{1}{2} \tan^{-1} \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \quad (3)$$

In the recognition step, the reference mask is superimposed on the reconstructed hand contour for each frame depending on the rotation of degree. In this way, each finger's state of open or close is identified by comparing the initial defects and the current defects. Fig. 9 shows the details on the recognition algorithm, in which states of each finger as well as the degree of rotation are identified based on the initial features of the reference mask.

The details on the algorithm for hand gesture recognition are given in Fig. 9. When the reconstructed hand contour is inputted for recognition of hand gesture. First of all, 7 defects which are used for recognition of open finger position are founded by using convexity defect algorithm. To match between the reference mask and reconstructed hand contour, after then, the degree of rotation of reconstructed hand contour should be calculated by using the gradient of moment. In this way, the reference mask is able to be mapped into the hand gesture contour according to its rotation.

In order to figure out whether the finger's state is open or close, an ellipse is drawn in the region finger, and if there are more pixels in the finger region than the threshold value inside of the ellipse, it is regarded that the corresponding finger is spread. It can be also recognized which finger is

spread by checking the defects inside of ellipse. Finally, the gesture of 'OK' sign can be recognized by checking the existence of a closed contour inside of the hand contour.

As the result of the recognition, the command corresponding to the recognized hand gesture is generated by mapping the recognized hand gesture into one of the set of commands predefined in a given application.

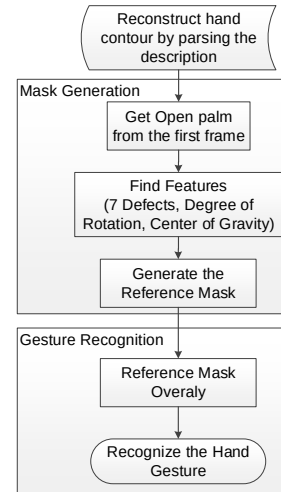


Fig. 8. Overall procedure of hand gesture recognition from reconstructed hand contour by parsing the delivered metadata

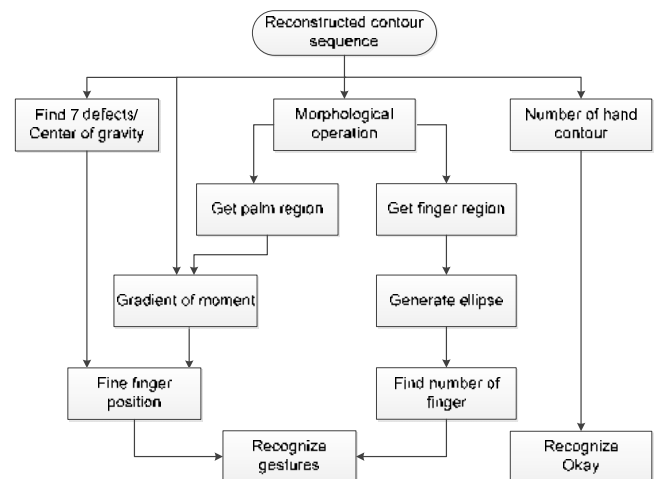


Fig. 9. Details on the procedure of the hand gesture recognition

By the way, the rotation of reference mask may be incorrectly calculated by the center of gravity when few fingers are opened or no finger is opened, which results in directionally biased form as shown in Fig. 10 (b) and (c). To recognize a rotated hand correctly by overcoming this problem, the rotational degree of reference mask should be revised according to the rotation of given hand contour.

To address this issue, the rotational angle of the reference mask is obtained more accurately by using Eq. (4). In other words, we consider an additional factor of palm rotational angle as well as both rotational angles of the incoming hand gesture and the reference mask. We use a morphological method to obtain either palm region or finger regions only. Then, the palm region is used for the calculation of rotation, and mask overlay with the consideration of the rotation.

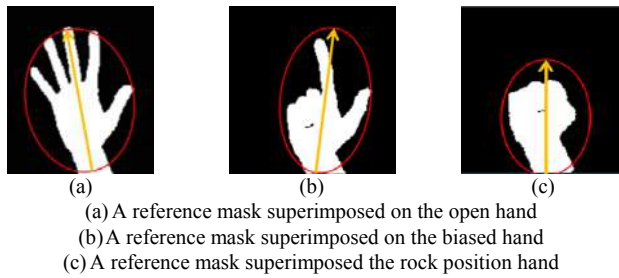


Fig. 10. An example of reference mask rotated depending on hand poses

As shown in Eg. (4), the value of R_{ref_mask} , the angle of reference mask that should be rotated when it is superimposed on the hand contour is compensated with the difference of rotational angle of palm and gesture.

$$R_{ref\_mask} = (A_{gesture} - A_{mask} + (A_{palm} - A_{gesture})) \cdot \frac{\pi}{180} \quad (4)$$

$A_{gesture}$: rotational angle of the hand gesture in the current frame

A_{mask} : rotational angle of the reference mask

A_{palm} : rotational angle of finger removed palm in the current frame

The obtained finger region is used for identifying whether a finger is open or not. In other words, we apply an ellipse to each finger regions, and determine the existence of finger according to the number of pixel lies inside the region overlapped ellipse. If the number of pixel is larger than threshold, then the existence of finger is true, which is marked '1'.

Finally, a post processing is applied on the recognition results of consecutive frames in temporal domain to reduce recognition error in each frame. The recognition error is occurred by not only in the process of changing hand gestures but also by fine shaking of hand. By accumulating the result of each frame, more reliable recognition result can be obtained for a given duration in the temporal post processing.

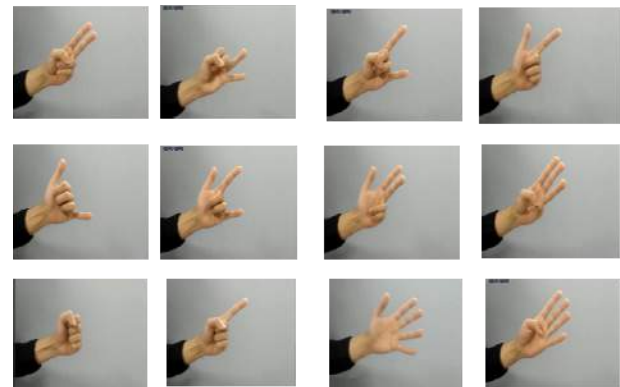
VI. EXPERIMENTAL RESULTS

In the experiment, a set of hand gestures to be recognized is given in Fig. 11, which includes diverse types of hand gesture with different combination of open fingers, different hand postures with the same number of fingers, and 'OK' sign. Using the seven defect points, it is possible to recognize which fingers are open as well as the number of open fingers. The proposed method uses the number of finger contour to figure out the number of open finger, and the position of open finger is recognized by using ellipse and convexity defects.

Example results of recognition are as shown in Fig. 12. As a result of hand gesture recognition, the position of each finger is obtained as '1' (open) or '0' closed. The binary code representing the position of five fingers are compared with those of the predefined gestures. Finally, the gesture image corresponding to the matched gesture is displayed as a final result as shown in Fig. 13. In addition, the gesture of 'OK' sign can be recognized by checking the existence of a closed contour inside the contour.

Some examples cases of final recognition results are shown in Fig. 13. The left and right images are the stereo images acquired by the stereo camera, and the centered small image shows the final recognition result. The example results for the same number of fingers with different postures are shown in Fig. 13 (a). Fig. 13 (b) shows the recognition result

for the 'OK' sign, and the recognition result depending on the degree of rotation of the hand is shown in Fig. 15 (c). In this way, any hand posture can be identified and mapped into the corresponding gesture based user's command in wearable applications



(a) Hand gestures with different postures of hand

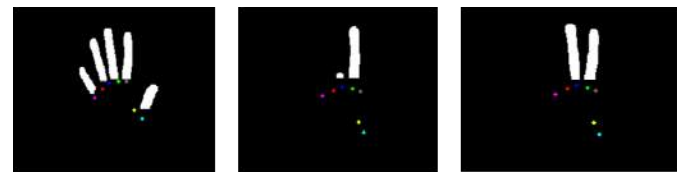


(b) OK sign



(c) Hand gesture with different number of finger

Fig. 11. A set of hand gestures to be recognized in the experiment



(a) Recognition results of each finger's position

Finger	Bin	Finger	Bin	Finger	Bin
THUMB	1	THUMB	0	THUMB	0
INDEX	1	INDEX	1	INDEX	1
MIDDLE	1	MIDDLE	0	MIDDLE	1
RING	1	RING	0	RING	0
LITTLE	1	LITTLE	0	LITTLE	0

(b) Binary code related with (a) for representing each finger's position

Fig. 12. Example results of hand gesture recognition

In the experiments, to measure the recognition accuracy of the proposed method, we use a pre-recorded video in addition to real-time video capturing hand gesture. The experimental conditions are summarized in Table 4, and the recognition accuracy is measure by using Eq. (5).

$$Recog\_accuracy(\%) = \frac{num\_of\_accurately\_recognized\_frame}{num\_of\_total\_frame} \quad (5)$$

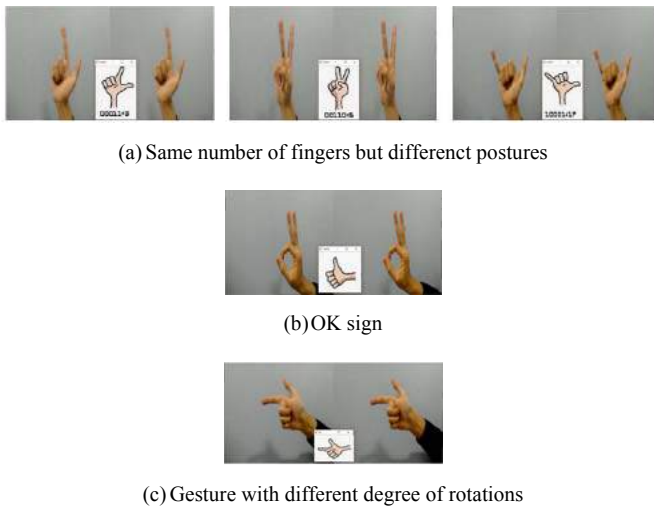


Fig. 13. Some final results of hand gesture recognition

TABLE IV
EXPERIMENTAL CONDITIONS

PC	Processor: Inter® Core™ i7-6700 CPU @3.40 GHz
	Memory: 16.0GB
	OS: Windows 10, x64
Input stereo image sequence	Frame rate: 6 frames/sec Resolution: 1280 * 720 Number of gesture type: 12 types Number of Frame: 252 frames

Table 5 shows the performance of proposed method in terms of recognition accuracy measured by Eq. (5). The comparison results are for both cases of pre-recorded video and real-time video with and without the temporal post-processing.

Pre-recoded video are captured in an ideal environment without illumination change and no movement of hand location except hand gestures themselves. In such ideal environment, we obtained the accuracy of 97.2% in a recognition of finger numbers and 95.0% of recognition of finger position, respectively. However, in experiments on real-time video captured by the head mounted camera, illumination and location of hand are not stable due to a little motion of head. As a result, the recognition accuracies are decreased to 95.2% and 89.9%, respectively.

With the temporal post-processing, the recognition accuracy of the number of finger and finger position using pre-recorded video are 100.0% and 98.6%, respectively. In the case of real-time video, the accuracies are given by 97.0% and 93.0%, respectively. This results mean that the proposed temporal post processing significantly enhances the performance of recognition accuracy higher.

TABLE V
EXPERIMENTAL RESULTS OF RECOGNITION ACCURACY

Test condition	Recognition accuracy			
	Pre-recorded video		Real-time video	
	Finger number	Finger position	Finger number	Finger position
Without post processing	97.2	95.0	95.2	89.9
With post processing	100.0	98.6	97.0	93.0

VII. CONCLUSION

In this paper, we presented a hand gesture detection and recognition method for gesture-based smart glasses applications in the context of MPEG IoMT, which aims to support efficient media consumption in IoT and wearable environments. In addition, we presented a method of representation of hand contours using Bézier curves to provide an interoperable interface between processing units each of which perform gesture detection and gesture recognition, respectively, in an IoMT framework.

Experimental results showed that the proposed methods of detection and recognition of hand gestures could be effectively applied to wearable applications such as smart glasses in real time. In addition, we have found that the Bézier curve based descriptor is more appropriate to support hand gesture based use cases than the MPEG-7 CSS and MPEG-U Part 2 based on the comparison analysis.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by Korean Government (MSIT) (R0127-15-1015). This paper is an extension of the ICACT 2017 conference paper.

REFERENCES

- [1] M. Mitrea, *Working draft 2.0 of ISO/IEC 23093-1 IoMT Architecture*, ISO/IEC JTC1/SC29/WG11 N17094, Torino, July 2017.
- [2] H. Cheng, L. Yang, Z. Liu, "Survey on 3D Hand Gesture Recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1659-1673, Sep. 2016.
- [3] M. Mitrea, S.-K. Kim, and S. Chun, *Use cases for Internet of Media-Things and Wearables*, ISO/IEC JTC1/SC29/WG11 N16533, Chengdu, China, Oct 2016.
- [4] A. Yang, S. Chun, H. Ko, J. G. Kim, *Hand Gesture Description for Wearable Applications in M-IoTW*, ISO/IEC JTC1/SC29/WG11 M38526, Geneva, Swiss, May. 2016.
- [5] A. Yang, S. Chun, and J.-G. Kim, "Detection and Recognition of Hand Gesture for Wearable Applications in IoMTW," In *Proc. ICACT2017*, Feb. 2017, pp. 598 – 601.
- [6] *Information technology – Rich media user interfaces – Part 2: Advanced user interaction (AUI) interfaces*, ISO/IEC 23007-2:2012, Oct. 2012.
- [7] *Information technology – Multimedia content description interface – Part 3 Visual*, ISO/IEC 15938-3:2002, Oct. 2001.
- [8] D. Zhang, G. Lu, "Evaluation of MPEG-7 shape descriptors against other descriptors," *Multimedia Systems*, vol. 9, no. 1, pp. 15-30, July 2003.



Anna Yang received the MS. degree from Korea Aerospace University, Korea, in 2017. She is currently working toward the Ph.D degree in the Department of Electronics and Information Engineering, Korea Aerospace University, Goyang-city, Korea. Her current research interests include IoT/wearable media applications, and MPEG standards.



Jae-Gon Kim received the B.S. degree in electronics engineering from Kyungpook National University, Daegu, Korea, in 1990, the M.S. and Ph.D degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1992 and 2005, respectively. From 1992 to 2007, he was with Electronics and Telecommunications Research Institute (ETRI), where he was involved in the development of digital broadcasting media services, MPEG-7/7/21 standards and related applications, and convergence media

technologies. From 2001 to 2002, he was a Staff Associate at the Department of Electrical Engineering, Columbia University, New York, USA. Since 2007, he has been with the Korea Aerospace University, Goyang-si, Gyeonggi-do, Korea, where he is currently a professor in the School of Electronics and Information Engineering. His research interests include digital video coding, video signaling processing, digital broadcasting media, and multimedia applications.



Sung Moon Chun received the B. S. degree from Sungkyunkwan University, Korea, in 1990. From 1990 to 2000, he was with Hyundai Electronics, where he was involved in the development of HDTV, MPEG-2/4 standards and related video codec technologies. From 2002 to 2012, he was with ECT Inc, where he was involved in the development of Semiconductor and related stereo video technologies. He is currently CTO in Insignal Inc. His research interests include video

compression, virtual reality, and wearable applications.