2014 International Conference on Robotics and
Emerging Allied Technologies in Engineering (iCREATE)
Islamabad, Pakistan, April 22-24, 2014

Dynamic Time Wrapping based Gesture Recognition

Saad Masood*, Majid Parvez Qureshi*, M. Bilal Shah*, Salman Ashraf*, Zahid Halim* and Ghulam Abbas*

*Faculty of Computer Science,
Ghulam Ishaq Khan Institute of Engineering Sciences and Technology,
Topi, Pakistan
{saadmasoodsaadi, mpq1990, mbilalshah1, salman.ashraf242}@gmail.com {zahid.halim, abbasg}@giki.edu.pk

Abstract—**Sign language provides hearing and speech impaired people with an interface to communicate with society. Unfortunately most people do not understand sign language. For this, image processing and pattern recognition can provide with a vital tool to detect and translate sign language into vocal language. This work presents a method for detecting, understanding and translating sign language gestures to vocal language. Microsoft Kinect is the primary tool used to capture video stream of the user. This is achieved by getting skeleton frame from Kinect and then extracting joints of interest. The data obtained are normalized and a linked list of skeleton frame is maintained. The proposed method is capable of successfully detecting all gestures that do not involve finger movements. The proposed system has an accuracy of 91%.**

Keywords—Feature extraction; Image recognition; Image sequence analysis; Multimodal sensors; Pattern recognition.

I. INTRODUCTION

The most common form of interaction between humans is vocal language. Language provides us with a tool to communicate. Spoken language is something that has allowed humans to have an upper hand from other living species thus giving us the ability to communicate and share ideas and thoughts directly. Unfortunately some people have not been blessed with the ability to speak or hear. They require some alternative method to communicate. Sign language is the primary alternative to spoken language. Sign language uses manual movements and body language to communicate thoughts and meanings with others. The basic component of sign language includes: hand movements, arm movements and also uses the facial expressions to communicate certain feelings. Every region in the world has a unique spoken language and similarly every region has a unique sign language. Sign language varies from culture to culture and region to region. Learning sign language is not an easy job and only trained persons are able to communicate using sign language. People with speech and hear impairing have difficulty to communicate with normal individuals via sign language.

The proposed system focuses on the problem explained above and uses image processing techniques to solve it. This work discusses and implements a system that facilitates the users by translating their gestures of sign language into spoken language. The basic tool used is Microsoft Kinect 360™ camera. Microsoft Kinect has the ability to directly provide depth images of body joints. Kinect uses an infrared camera so the problems of lightning conditions are minimized. Previously, work has been done in this area, to the best of our knowledge most of it is either focused on few limited gestures or ignores those sign language gestures that involve hand/fingers.

The proposed system has the ability to detect gestures without finger movements with an accuracy of 91%. A feature that is included in the proposed system is the ability of user to define and add his/her custom made gestures. This allows the system to support sign language of any region and culture. Though the focus of this paper is on gestures related to sign language but overall scope of the proposed approach is not limited to these. It can be extended to gesture recognition of any sort and be used in other systems too where gesture recognition is appropriate. The system is able to detect gestures that are performed between head and hip bone based upon the fact that gestures of sign language are performed between these areas.

The objective is achieved by using a number of computing techniques aided by Microsoft Kinect SDK. Microsoft Kinect has two modes (1) Recording mode and (2) Translation mode. In Recoding mode, the Kinect stream is captured and coordinates that respond to joint of interest are extracted from it. These are the Cartesian coordinates. Size of users would definitely vary so it is imperative to normalize the data so that the users' size does not have an impact when comparing gestures. Normalization is done by firstly shifting the origin from Kinect to human body spine. Secondly the coordinates are shifted to spherical coordinate system. The distances of joint of interest from origin are divided by distance of origin to head and hence normalization of data is achieved. Next the normalized skeleton frames are stored in a linked list and once the gesture is completed, it is written on to the memory in a gestures dictionary. In translation mode, all of the procedure (listed earlier) is similar to that of recording mode. The only difference is where the gesture ends. Unlike recording mode where the gestures are stored in a gesture dictionary, in translation mode the gesture currently stored in a linked list is compared to the gestures in the dictionary. This is achieved through Dynamic Time Wrapping algorithm [1].

Rest of the paper is organized as follows. Section 2 presents the related work. Section 3 contains the proposed methodology of our system. Section 4 lists the experimentation and Section 5 concludes the paper with some of the future directions.

II. Related Work

This section covers the recent and few basic literatures that are related to the problem statement at hand. Initially few of the basics of sign language are explained followed by the techniques used to make sign language understandable by machine.

Human born with deafness or speech impairing uses sign language to interact with normal people. Sign language and the spoken language are mostly similar to each other. Both have their own grammar and vocabulary. The difference between the two is that sign language is performed using hands whereas spoken language is with the help of tongue. To understand sign language no voice and listening is needed, so it is used by deaf people.

Sign language not only requires hands but also movement of arms, specific shape of fingers, facial expressions, body and head. Most importantly sign language is not universal. Different counties have their own signs e.g. Pakistani sign language contain nearly 4000 signs for different words. Different regions, towns, and cities can also have their own "local sign language". Each sign, due to the involvement of different body parts convey the meaning slowly than the words, but short sign conveys more meaning than a single short word [2-6].

Juan Pablo Bonet mentions in [4], that every letter has some sign and in order to say a word, the whole sign for a particular word was the combination of every sign of the particular letter. This means that this sign language was not efficient at all, it only makes communication possible. Proper Sign language was first introduced in 1960 under the instruction of Harry Bornstein of Gallaudet College.

Manual component involves the hand whereas involvement of rest of body parts in the representation of signs can be described as non-manual component. As we know that hand is one of the components in sign languages from which signs can be made. Two signs can be different using hand in term of hand location, orientation, shape and movement. Single handed signs can be in a state of motion or can be represented using static or rest position of hand. Double handed signs involve the domination of one hand over the other or both hand share equal priority [2],[7-9].

As in spoken language there is past, present and future tenses. This depends on the same sign performed with differences in repetition of the sign, performing same sign with different facial expression or movement of different body parts. To convey the verbal aspects, sign is performed with repetitions. Mixture of two signs is used to convey a full sentence [2]. Sign language also has syntax like question is represented by raising the eyebrows [8-9].

Work in [10] mentions uses of Kinect to implement the gesture recognition system for media player. To capture the hand motion, 3d vector and to detect the hand gesture, Hidden Markov Model [11] was used. The system was limited to perform and detect hand gesture which does not involve different alignment of fingertips. System was able to detect up to down or left to right movement of hands. Raheja et al.'s [12] system was able to grab finger tips and to identify the center of palm. Segmentation was used to separate the hand from video frames. After segmentation palm was extracted from hand so that system can get the finger. Then by obtaining minimum depth finger tips were found.

Detection and Normalization can be used to detect hand gesture. After normalization Dynamic Time Wrapping can be applied to identify gestures. Color images and depth images can be used to detect hand. Generic skin color histogram is used to calculate the value of skin color. By assigning some values to the pixel containing skin color in a frame to calculate the probability of having skin there. Color images alone cannot identify the hand but hand is one of the components in the values, so by using color images with depth images using Kinect will help to detect the hand alone. Normalization involves making a set by getting 2D points of hand region. This point is used to calculate the Minimum Enclosing Circle (MEC) of the set to get the center of the hand. By drawing a circle at MEC center of twice radius of MEC circle, features will be normalized [13].

Older people living on their own can help themselves in many situation but they do need somebody to take care for them in critical situation like getting pain in the chest. They hire live-in nurse to care for them. A system in [6] is designed to help them and also to help doctor to measure the severest of pain. Since Kinect can detect the body and after getting the body depth images system tracks the movement of hand or body to guess the position where the pain occurs. This information is transmitted to visualization application where there is rating to get the severity pain position.

To locate the hand motion, an improved Camshift tracking is proposed in [14] combined with the information extracted using RGB color information and depth images captured by Kinect. Then Hidden Markov Model and FNN are used to identify gestures. Tracking error will occur in frame where there are similar colors, due to color image information used by CamShift algorithm. Orientation feature was given the highest priority over two other basic feature i.e. velocity and location. Difference between two simultaneous point on hand gesture path denotes the orientation.

Swiss Ranger SR4000 camera is a range and intensity capturing camera using integrated sensor at the same time. Unlike Kinect, this camera does depend on light condition. The only goal is counting the number of raised fingers and to identify them by grabbing object using only hand gesture. Extracting the information from hand require segmentation which involves range based segmentation, integration time model, noise removal, connected component analysis, and hand tracking. To draw line on the outer point of the hand, first center of the hand that is palm was determined by finding the center of gravity of segmented data. Next, using the version of convex hull the outer point is lined. In convex hull a point on the left side is chosen and then this point is used to get the neighboring points. The point on the clockwise side is chosen and point on anti-clockwise is neglected by drawing line from that point to last clockwise point. Grabbing object using hand gesture in virtual environment involves recognizing the gesture and then using visualization application to show the results. This process involves segmentation of hand, determining the position of the object and the orientation of the object. The center of the object and center of hand is at same coordinates in the model. Three rotation angles ω, ϕ, and K are used to locate the position and orientation of the object. [12],[15-17].

III. METHODOLOGY

The proposed system consists of a series of steps for the detection of any particular gesture. Fig. 1 lists the flow of these activities. Microsoft Kinect is used to get stream of input frames form the user. This stream is then processed and stored in memory for purpose of gesture recognition. This whole process is divided into number of steps which we later in this paper discuss in detail. The individual steps are (1) receiving joint of interest from Kinect video stream (2) normalizing the skeleton frame data (3) building a linked list of the normalized data and (4) storing or detecting the gesture.

There are two modes in this process. The first is recording mode and the second is translation mode. In recording mode the user is able to add gestures to the dictionary whereas in the translation mode the user performs a gesture and that gesture is compared to gestures stored in dictionary. Most of the sign language gestures are above the body's hip bone and below head, so in order to start and end a sign language gesture the user has to perform a specific defined gesture. Our system requires the user to place both hands below hip bone with distance between hands below 0.5 m to be able to start or end a sign language gesture. We call this gesture "Recording Translation Gesture" or "RTG". After the RTG is performed the following sequences of steps follow:

A. *Joints of interest*

Kinect is able to detect and track human body joints. Microsoft Kinect SDK provides with functions to get the Cartesian coordinate of the joints. We use these coordinates to store the movements performed by the users in order to record the gesture performed. We can get 20 joints of human body with Microsoft Kinect SDK. In our case we will be using only those joints which are required for detecting sign language gestures. For RTG gesture we require hip joint and center joint. For sign language gesture, eight joints are required which are head (H), spine (S), elbow right (ER), elbow left (EL), hand right (HR), hand left (HL), wrist right (WR) and wrist left (WL) we will be storing and tracking coordinates of these joints and then normalize these.

B. *Normalization*

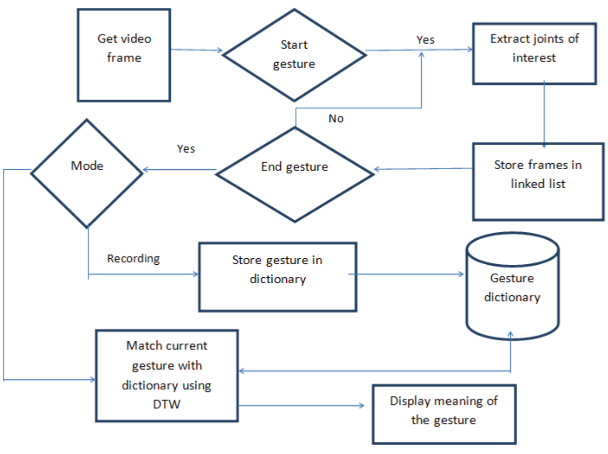

Fig. 1 Methodology for gestures recognition

Every user's height and dimensions will be different. This has a huge impact on the performance of the system. The reason being X, Y and Z coordinates of joints of every user might be different. This can also happen due to varying position of user from Kinect. Ideally a user should be 6 feet from Kinect and straight in front of camera but it is not always the case. The user can be at any angle from Kinect and at any distance. So a need to normalize the data is of necessary to increase accuracy of gesture recognition.

The Cartesian products that are retrieved from Microsoft Kinect SDK are with respect to Kinect, we first shift the origin of the coordinates from Kinect to a point in human body. The joints of interest we have used are: (1) head, (2) right wrist, (3) left wrist, (4) right hand (5) left hand, (6) spine, (7) hip bone, (8) left shoulder, (9) center shoulder and (10) right shoulder. The coordinates when captured are in Cartesian coordinate system. These are then converted to a three dimensional space system known as spherical coordinate system. The reason for this is that the coordinates are easy to normalize. Regardless the size user, only distance from origin will vary unlike all coordinates in Cartesian system and the angles will remain constant. We would now only require normalizing this distance as explained later. This system consists of three attributes (1) distance of point from origin (2) a polar angle and (3) an azimuth angle known as r, ϑ and ҩ respectively. Θ is angle measured with reference to Z axis and ҩ is measured with respect to x-axis in two dimensional X-Y coordinate planes.

The formulas that are used to convert from Cartesian coordinate system [x, y, and z] to spherical coordinate system are listed in equation (1), (2) and (3).

$$r = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

For inclination angle ϑ, the angle with respect to z-axis:

$$\vartheta = \arccos\left(\frac{z}{r}\right) \quad (2)$$

For azimuth angle ҩ, the angle with respect to x-axis in two dimensional x-y plane:

$$\varphi = \arctan\left(\frac{y}{x}\right) \quad (3)$$

To minimize the problem of difference in size of users we normalize the distances of the joints from the origin. As mentioned earlier in spherical coordinate system whatever the size of user is only the distance r will vary whereas the angles ҩ and ϑ will remain constant .This is achieved by dividing all the joints distances by a factor. We have chosen this factor to be the distance of head to origin. Once this is done the normalization (equation 6) of data of a frame is complete.

$$Normalizedist(r_n) = \frac{Each\_joint\_dist\_fromOrigin(r)}{Head\_to\_origin\_dist.} \quad (4)$$

C. *Temporary Storage*

Once the normalization is completed the data is to be stored in memory. A linked list is maintained to store normalized skeleton frames until the gesture is completed. This is done by storing coordinates of joints in private variables in objects of gesture class and forming a linked list of the objects. The ending RTG marks the end of a gesture. When the complete system executes, a dictionary (explained in next section) is loaded in memory in form of a two dimensional linked list of objects. All gestures are linked vertically, whereas each gesture individually is connected to list of objects that contains the values of joints, spherical coordinates and their normalized coordinated for each frame. Linked list has been the choice to accommodate dictionary in memory due to large size of the text file storing gestures.

D. Dictionary

At this instance, the specific mode of system determines what to do next. Depending on whether it is in recording mode or translation mode the next step would execute. In recording mode, once the ending RTG is performed, the normalized skeleton frame linked list is written in the gesture dictionary. The gesture dictionary is a text file with joints coordinates stored and separated by commas and each gesture is enclosed in braces in order to differentiate them. Fig. 2 shows a snapshot of the gesture dictionary. The coordinates of the gesture "Ten" are stored. Joints coordinates from different users are stored here in the dictionary.

If the mode is translation, DTW algorithm is applied to compare the gesture stored in linked list with the gesture stored in gesture dictionary and the result of corresponding gesture is returned. Each gesture is linearly searched for comparison and compared through DTW algorithm. Dynamic time wrapping algorithm is a general algorithm used to compare different elements of different lengths [18-23].

{Ten;1;1.825039;1.61842;0.4799416;1.752347;-
0.03365774;0.4809779;1.604645;3.227344;0.7872145;1.931323
;-
0.9921522;0.7998513;1.81172;4.062463;0.6572626;1.928003;-
0.7747315;0.6745825;1.747543;3.866844;1;1.800531;1.61658;0
.4763202;1.743661;-
0.04746844;0.4956383;1.588604;3.21181;0.7973173;1.926927;
-
0.9780486;0.8204022;1.810149;4.007045;0.6641551;1.922834;
-
0.7688597;0.6883463;1.736221;3.826816;1;1.76366;1.614945;0
.4762893;1.730185;-
0.05508661;0.5016035;1.572025;3.202354;0.7909063;1.926266
;-

Fig. 2 Gesture Dictionary

IV. EXPERIMENTS

A detailed set of experiments are conducted on the proposed system making sure that it is tested under different scenarios with diverse processing data. For experimentation seven gestures have been used, among them three are test gesture that are generic signs e.g. "Four" and "Third Umpire" gestures used in cricket by umpires and a general "ten" gesture. Rest four sign language gestures belong to the Pakistani sign language.

Initially, four individuals were asked to enter these signs separately in test sign language dictionary. Afterwards the software was tested by three different individuals. Testing was done by changing distance form user to Kinect and giving different weight thresholds to elbow and head joints. This was done to test the software's capability to detect gestures and also to find the best combination of distance and weights required to produce best results.

To test the accuracy of the system three set of

Table 1—System testing with parameter configuration 1

	D=6 ,H=1,E=1			D=6,H=0.5,E=0.2			D=6,H=0.3,E=0.1		
Ten	1	1	1	1	1	1	1	1	1
Four	1	1	1	1	1	1	1	1	1
Third Umpire	1	1	1	1	1	1	1	1	1
Greetings (Asslam-u-Alikum)	1	1	1	1	1	1	1	1	1
Quit	1	1	1	1	1	1	1	1	1
Order (Hukm)	1	1	1	1	1	1	1	1	1
Prize	1	1	1	1	1	1	1	1	1

Table 2—System testing with parameter configuration 2

	D=9 ,H=1,E=1			D=9,H=0.5,E=0.2			D=9,H=0.3,E=0.1		
Ten	1	1	1	1	1	1	1	1	1
Four	1	1	1	1	1	1	1	1	1
Third Umpire	1	1	1	1	1	1	1	1	1
Greetings (Asslam-u-Alikum)	1	1	1	1	1	1	1	1	1
Quit	0	0	1	1	1	1	1	1	1
Order (Hukm)	1	1	1	1	1	1	1	1	1
Prize	1	1	1	1	1	1	0	0	1

Table 3—System testing with parameter configuration 3

	D=12 ,H=1,E=1			D=12,H=0.5,E=0.2			D=12,H=0.3,E=0.1		
Ten	1	1	1	1	1	1	1	1	1
Four	1	1	1	1	1	1	1	1	1
Third Umpire	1	1	1	0	0	0	1	1	1
Greetings (Asslam-u-Alikum)	1	1	1	1	1	1	1	1	1
Quit	1	1	1	1	1	1	1	1	1
Order (Hukm)	0	1	0	1	1	1	1	0	1
Prize	0	0	1	1	1	1	1	1	1

experiments were performed with different configuration of distance from Kinect and weights assigned to hand and elbow. First, the test users were positioned at a distance of 6 feet from Kinect with weight assigned to hand and elbow as 1. The same test was repeated with keeping the distance from kinet constant and changed the weights of hand and elbow to 0.5 and 0.2 first and then 0.3 and 0.1. Each of three users tested the system performing all seven gestures in the dictionary three times. The system output was a Boolean value where 1 was interpreted as a gesture recognized and 0 as not recognized. With aforementioned configurations all the gestures were successfully recognized by the system. Table 1 shows the detailed results of the experiment.

For the second configuration, users were positioned at a distance of 9 feet from Kinect with weight assigned to hand and elbow as 1. The same test was repeated with keeping the distance from kinet constant and changed the weights of hand and elbow to 0.5 and 0.2 first and then 0.3 and 0.1. Each of three users tested the system performing all seven gestures in the dictionary three times. In case any single attempt failed, we interpret it as gesture reorganization failed. With aforementioned configurations all the gestures with the exception of 2 were successfully recognized by the system. Table 2 shows the detailed results of the experiment.

For the third configuration, users were positioned at a distance of 12 feet from Kinect with weight assigned to hand and elbow as 1. The same test was repeated with keeping the distance from kinet constant and changed the weights of hand and elbow to 0.5 and 0.2 first and then 0.3 and 0.1. Each of three users tested the system performing all seven gestures in the dictionary three times. As in case of second configuration if any single attempt failed we interpret it as gesture reorganization failed. With aforementioned configurations all the gestures with the exception of 4 were successfully recognized by the system. Table 3 shows the detailed results of the experiment.

After testing, high accuracy of the system is observed. The system was successful in detecting almost all of the gestures. Table 4 shows the accuracy in percentage of all types of configurations. Overall accuracy of the system is found by taking average of each of the tests resulting in an accuracy of 91 %. It should be noted that as the training data is increased the accuracy of the system can also increase. Further, if the distance from Kinect is decreased from 6 feet or increased than 12 feet the accuracy of the system will decrease in either case.

CONCLUSION

This paper presents an efficient technique used to detect and recognize gestures. The methodology use Microsoft Kinect SDK to get joints of interest from user and after their normalization, stores them in a gesture dictionary. DTW algorithm is used to match gestures with dictionary. The proposed approach is capable of detecting gestures with an accuracy of 91%. By increasing the size of training data the accuracy can be further increased. This system is capable of detecting all gesture which doesn't involve specific finger movements. It can be used by hearing and speech impaired people to communicate with rest of society. In future we plan to extend this software with finger detection. This involves the user to stand very close to Kinect; it couldn't be linked to rest of gesture recognition as by this short distance, 1 to 2 feet, Kinect is unable to detect skeleton and joints. This work used clustering, Graham Scan and contour tracing to detect fingers and hand. Future work needs to be done to combine the two techniques of gesture recognition and finger detection for a complete system capable of detecting any type of gesture.

REFERENCES

[1] Rath T M & Manmatha R, Word image matching using dynamic time warping, In Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, Madison, 18-20 June 2003, pp. II-521.

[2] Li K F, Lothrop K, Gill E & Lau S, A web-based sign language translator using 3d video processing, In Proc of IEEE International Conference on Network-Based Information Systems, Melbourne, 26-28 September, 2011, pp. 356-361.

[3] Hollis S. Sign Language for Beginners Discover the Art of Sign Language. Print Smarter Bristol and West House, Bournemouth, pp. 20-31, 2011.

[4] Alvi A K et al. Pakistan sign language recognition using statistical template matching, Proceedings of world academy of science, engineering and technology. Vol. 3, pp 1-12, 2005.

[5] Farhadi A, Forsyth D & White R, Transfer learning in sign language, In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, 17-22 June, 2007. pp. 1-8.

[6] Johnson J E & Russell J J, Assessment of regional language varieties in Indian Sign Language, Report submitted to Work Papers of the Summer Institute of Linguistics, University of North Dakota, pp. 90-101, 2005.

[7] Diwakar, Synny & Basu A, A multilingual multimedia Indian sign language dictionary tool, In Proc. of 6th Workshop on Asian Languae Resources, Hyderabad, India, 1-12 January, 2008, pp. 65-72.

[8] Lucas C. Sign Language Research : Theoretical Issues. Gallaudet University Press. Pp. 27-49, 1990.

[9] Fischer S & Siple P. Theoretical Issues in Sign Language Research Volume 1:Linguistics. University of Chicago Press, pp. 55-86, 1990.

[10] Li Y, Hand gesture recognition using Kinect, In Proc. of 3rd IEEE International Conference on Software Engineering and Service Science, Beijing, 22-24 June, 2012, pp. 196-199.

[11] Lang S, Block M & Rojas R, Sign Language Recognition with Kinect, In Proc. of the 11th international conference on Artificial Intelligence and Soft Computing, Zakopane, April 29 - 3 May, 2012 , pp. 394-402.

[12] Raheja J L, Dutta P J, Kalita S & Lovendra S, An Insight into the Algorithms on Real-Time People Tracking and Counting System, International Journal of Computer Applications, Vol. 46, No. (5), pp. 1-6, 2012.

[13] Doliotis P, Stefan A & McMurrough C, Eckhard D & Athitsos V, Comparing gesture recognition accuracy using color and depth information, In Proc. of the 4th ACM International Conference on Pervasive Technologies Related to Assistive Environments, Greece, 25-27 May 2011, pp. 20.

[14] Xu W & Lee E, Gesture Recognition Based on 2D and 3D Feature by using Kinect Device, In Proc. of International Conference on Infromation and Security Assurance, Shanghai, 28-30 April , 2012, Vol. 6, No. 1, pp. 77-80.

[15] Lahamy E & Lichti D, Real-time hand gesture recognition using range cameras, In Proc. of the Canadian Geomatics Conference, Calgary, Calgary, 15-18 June, 2010, pp. 1-6.

[16] Appenrodt J, Al-Hamadi A & Michaelis B, Data gathering for gesture recognition systems based on singlecolor-, stereo color-and thermal cameras, International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 3, No. 1, pp. 37-50, 2010.

[17] Hu G, A Generic Gesture Recognition Approach based on Visual Perception, Ph.D. Thesis, Dalhousie University, Canada, pp. 98-130, 2012.

[18] Senin P, Dynamic time warping algorithm review, Information and Computer Science Department University of Hawaii, USA, pp. 1-23, 2008.

[19] Munich M E & Perona P, Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification, In Proc. of 7th IEEE International Conference on Computer Vision, 1999. Vol. 1, pp. 108-115.

[20] Wöllmer M, Al-Hames M, Eyben F, Björn S & Gerhard R, A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams, Neurocomputing, Vol. 73, No. 1, pp. 366-380, 2009.

[21] Niels R & Vuurpijl L, Using Dynamic Time Warping for intuitive handwriting recognition, In Proc of Advances in Graphonomics, Salerno, Italy, 26-29 June, 2005, pp. 217-221.

[22] Salvador S & Chan P, Toward accurate dynamic time warping in linear time and space, Intelligent Data Analysis, Vol. 11, Vol. 5, pp. 561-580, 2007.

[23] Yang C, Jang Y, Beh J, Han D & Ko H, Gesture recognition using depth-based hand tracking for contactless controller application, In Proc. of IEEE International Conference on Consumer Electronics, Los Vegas, 13-16 January, 2012, pp. 297-298