Kinect-based Gesture Command Recognition for Vending Machine Operations Using an Android Platform as the Machine Simulator

*Ing-Jr Ding, Rui-Zhi Lin and Zong-Gui Wu

***Abstract*—In this paper, a Kinect-based gesture recognition scheme for the application of operating a vending machine is presented where the popular smart phone device with the android platform is used as a simulator of the vending machine. Compared to the conventional human-computer interface (HCI) of vending machines by push-button operations, the developed gesture recognition scheme that employs the gesture action made by a personal operator as the operational command is a new type of HCI to control the vending machine. Presented Kinect-based gesture recognition for vending machine control sufficiently takes use of the released Kinect software development kit (Kinect SDK) to develop the system where the human skeleton information to represent the corresponding operator's gesture action is extracted and then recognized. For further increasing the recognition accuracy of gesture command recognition, a time interval between the current recognition decision and the next recognition decision is investigated and analyzed. A proper value of decision time-interval will then be provided to the gesture recognition system for avoiding the imperfect recognition accuracy that results from an arbitrary and improper setting of the decision time-interval. Experimental results showed that presented Kinect-based gesture command recognition for vending machine operations is feasible and also competitive on recognition performances. On the use of the released Kinect SDK for establishing the gesture recognition system, gesture recognition with an appropriately-designed time interval of decision making will perform apparently much better than that without any investigation information available about decision time-interval settings.**

***Keywords*—Kinect, gesture recognition, decision time-interval, vending machine, Android**

I. Introduction

Nowadays, the user interface (UI) of the vending machines on the market belongs to the category of the touching style where the hand-touching button is provided for each vending item. Such the conventional UI for the vending machine could encounter some disadvantages such as the unclean touching button and the insensitive button functionality. With the appearance of the Kinect device [1, 2], the touching button UI can be replaced with the human gesture recognition UI. The user interface design by human gesture recognition is to use the active person's gestures to control the target device, which belongs the category of the contactless style and is extremely appropriate for the vending machine operation application.

In fact, the Kinect device made by the Microsoft company [1, 2] speeds up the development of human gesture recognition. The Kinect sensor and its software development kit (Kinect SDK) can be efficiently used to simplify the conventional gesture recognition technique. Recently, studies on Kinect-based gesture recognition have been frequently seen [3-6], and Kinect-based gesture recognition applications mainly include robot automation, health-care, entertainment and surveillance. For the vending machine operation application using Kinect-based gesture recognition, extremely rare studies are done.

Although gesture recognition with Kinect SDK have been widely used in lots of application fields, the recognition performance on gesture recognition is still dissatisfactory due to the restriction of the unchangeable Kinect SDK. This paper proposes Kinect SDK-based gesture recognition with a flexible decision time-interval scheme for further enhancing the recognition accuracy of Kinect SDK-based gesture recognition. The presented method is performed on the application of vending machine control simulations. On the condition of gesture recognition using the restricted Kinect SDK, the presented decision time-interval scheme can still significantly increase the recognition accuracy, which will be detail in the following.

II. Kinect-based Gesture Command Recognition Systems

In general, the gesture command recognition system using the Kinect device could be mainly divided into two design categorizations, gesture recognition without the utilization of the Kinect software development kit and gesture recognition using the complete Kinect software development kit. This study aims at presenting an innovative idea to use Kinect for easily performing gesture command control on vending machine operations, and therefore the complete Kinect SDK is employed for carrying out gesture command recognition. Note that a flexible decision time-interval scheme for enhancing gesture recognition that mainly employs the Kinect SDK is proposed, which will be presented as follows.

A. *Gesture Recognition Computations with/without Kinect SDK*

Compared with the conventional gesture recognition technique without the support of the Kinect device, Kinect-based gesture recognition is conceptually simple and easy to construct. With the Kinect SDK released by the Microsoft company, a Kinect-based gesture command recognition system will be able to be established quickly. In fact, the main characteristics of the Kinect incorporated with the related SDK is the extraction of the human skeleton information. The human skeleton information obtained by

*Corresponding author: Ing-Jr Ding

Department of Electrical Engineering, National Formosa University
Taiwan

the Kinect device contains primarily the (*x*, *y*, *z*)-location data of all kernel joints in a human body.

For the categorization technique of gesture recognition computations without the utilization of the Kinect software development kit, the Kinect-extracted human skeleton data is then efficiently utilized for the self-developed feature extraction calculations and the corresponding gesture pattern recognition estimations. In the work of [3], the gesture recognition computation is performed using an eigenspace approach where the principal component analysis to the three dimensional location data of all kernel joints is done for deriving the eigenspace of Kinect-extracted human action gestures. A hidden Markov model (HMM) approach for Kinect-based gesture recognition is proposed in [4] where the recognition accuracy of the HMM gesture model could be improved by properly adjusting HMM model parameters. In addition, in [5], a feature design scheme is developed for the computation of Kinect-based dynamic time warping gesture recognition. All those gesture recognition computations without the utilization of the Kinect software development kit in [3-5] will be able to improve the recognition performance of Kinect-based gesture recognition by the self-developed gesture recognition method, the gesture model adjustment scheme, or the accurate gesture feature for recognition.

For the categorization technique of gesture recognition computations with the Kinect SDK, however, such those improvement strategies presented in [3-5] for increasing the accuracy on gesture recognition cannot be employed due to an absolute restriction on developments of recognition system. Gesture recognition computations with the Kinect SDK use a direct library-style tool for rapidly establishing the gesture recognition system, which belongs to the system integration method. Unlike those system development methods in [3-5], such the system integration method by only the Kinect SDK tools can’t change the original source design of gesture recognition computations. A feasible way to enhance the gesture recognition system where gesture recognition is computed directly by only the Kinect SDK is to consider a flexible decision time-interval scheme, which will be detailed in the following section.

B. *A Flexible Decision Time-Interval Scheme for Enhancing Gesture Recognition with the Kinect SDK*

The section will present the proposed flexible decision time-interval scheme for further improving the performance of gesture recognition with the use of the Kinect SDK. Conventionally, gesture recognition with the Kinect SDK does not properly consider the value of the time-interval between the current recognition and the next recognition. Too compact arrangements for each gesture recognition calculation will disorder the gesture recognition system and lead to decrease the recognition performance of gesture recognition. Therefore, a time-interval design, denoting a pause to the computation of gesture recognition, will be absolutely necessary for the Kinect SDK-gesture recognition.

Figure 1 illustrates a time-interval scheme embedded inside the overall gesture recognition process. Observed from Fig. 1, three types of time-interval durations, the short time duration, the middle time duration or the large time duration will be able to be considered when developing gesture recognition. The estimate of the value of the time-interval between the current recognition and the next recognition will have a crucial impact on gesture recognition performance. A too short time duration of the decision time-interval embedded two continuous gesture recognition decisions will lead to the occurrence of a series of incorrect gesture recognition results due to certain gesture action with incomplete gesture data recognized. In such the situation of extremely short time interval, it will be possible to perform each gesture recognition calculation on the incomplete gesture action data. Conversely, a too large time duration for the decision time-interval is still not proper, which will cause a too long time duration for making a gesture recognition decision. In the case of large values of the decision time-interval, the recognition system will not meet the requirement of the real-time recognition. Therefore, only a proper value for the decision time-interval will be effective to increase the performance of gesture recognition. In this work, a brute force-like method is used to find the optimal decision time-interval. In the presented scheme, all the decision time-intervals with the value distributed from the extremely small case to the extremely large case will be evaluated the effectiveness to the performance of gesture recognition. A training database including all classes of gesture command action data acquired from numerous active players will be established in advance. The optimal value for the decision time-interval is then estimated using the built gesture database. Figure 2 depicts the flowchart of the presented flexible decision time-interval scheme for enhancing gesture recognition with the Kinect SDK. An initial value is firstly set for the decision time-interval, and in general, such the initial value is extremely small. A setting of the time-interval will be used to evaluate all gesture command data in the gesture database. After finishing an iterative evaluation using the set value of the time-interval, the current time-interval will be increased by an increment, and the time-interval with the updated value will be employed for evaluating all gesture command data again. As could be seen in Fig. 2, the presented method for estimating the optimal time-interval is an iterative approach.

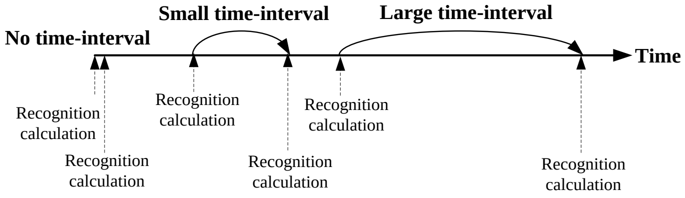

Figure 1. Kinect-SDK gesture recognition with no any time-intervals, the small time-interval and the large time-interval

For Kinec SDK-gesture recognition calculations, there are generally two design categories, feature-based and model-based calculations. The feature-based approach uses the characteristics of the extracted feature of the obtained active gesture data for classifying each different class of gesture commands. In mode-based Kinect SDK-gesture recognition, a gesture recognition model is required to be trained in advance using a proper Kinect SDK tool. For example, the Kinect SDK dynamic time warping gesture recognition system can be established fast using the corresponding DTW SDK [7]. In this work, the feature-based Kinect SDK recognition calculation is employed.

Feature-based Kinect SDK recognition calculation in this paper mainly employs the skeletal tracking and the 2-dimentional (X, Y) position data of the kernel joints. In this work, there are totally ten gesture commands made, and

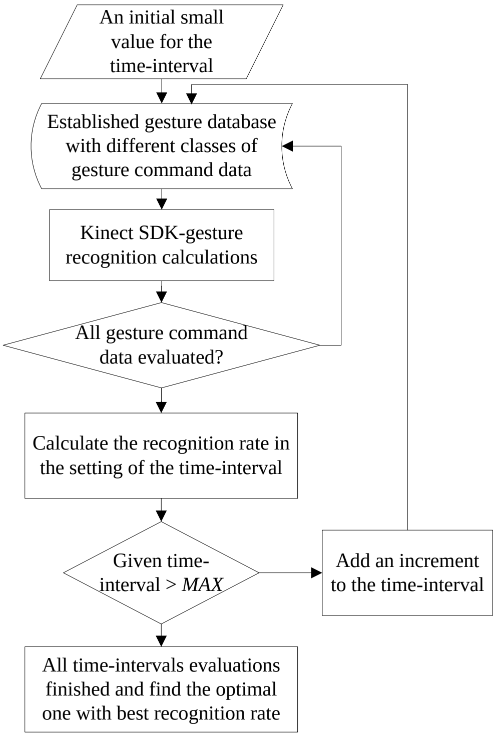
An initial small value for the time-interval
Established gesture database with different classes of gesture command data
Kinect SDK-gesture recognition calculations
All gesture command data evaluated?
Calculate the recognition rate in the setting of the time-interval
Given time-interval > *MAX*
Add an increment to the time-interval
All time-intervals evaluations finished and find the optimal one with best recognition rate

Figure 2. A flowchart of presented flexible decision time-interval scheme for enhancing gesture recognition with the Kinect SDK

each of all gesture commands is indicated by an active gesture operation. These ten gesture commands are “Waving the right hand,” “Waving the left-hand,” “Lifting the right hand over the head,” “Lifting the left hand over the head,” “Putting the right hand on the right shoulder,” “Putting the left hand on the left shoulder,” “Waving the left leg,” “Waving the right leg,” “Putting the right hand on the right waist,” and “Putting the left hand on the left waist.” Figure 3 shows the gesture command of “Waving the left-hand,” and nine colored points, one blue point and eight red points, denote the kernel human joints. These colored points 1, 2, 3, 4, 5, 6, 7, 8 and 9 represent “Head-joint,” “Left shoulder-joint,” “Right shoulder-joint,” “Left hand-joint,” “Right hand-joint,” “Left-waist joint,” “Right-waist joint,” “Left-leg joint,” and “Right leg-joint.” Feature-based Kinect SDK-gesture recognition used in this work mainly employs the two-dimensional (X, Y) spatial position information of these nine kernel human joints to perform the classification task of the above-mentioned ten gesture command operations. A If-then rule-based classification scheme to classify these ten gesture commands using the Kinect SDK-extracted (X, Y) joint position features is presented as follows.

Command 1: If P5.X is larger than (P1.X + 0.5), then the gesture command is “Waving the right hand.”

Command 2: If P4.X is larger than (P1.X - 0.5), then the gesture command is “Waving the left hand.”

Command 3: If P5.Y is larger than P1.Y, then the gesture command is “Lifting the right hand over the head.”

Command 4: If P4.Y is larger than P1.Y, then the gesture command is “Lifting the left hand over the head.”

Command 5: If P5.Y is larger than P3.Y, then the gesture command is “Putting the right hand on the right shoulder.”

Command 6: If P4.Y is larger than P3.Y, then the gesture command is “Putting the left hand on the left shoulder.”

Command 7: If P8.X is larger than (P1.X - 0.1), then the gesture command is “Waving the left leg.”

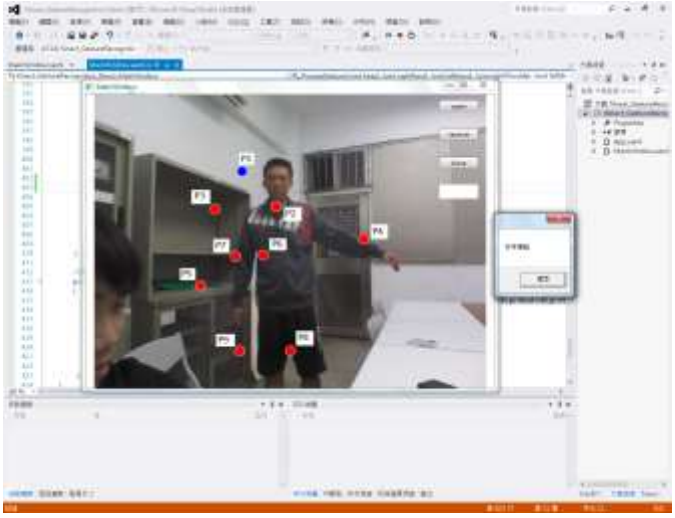

Figure 3. Feature-based Kinect SDK-gesture recognition using the spatial position information of kernel human joints (the gesture command “Waving the left-hand” operated by an action player)

Command 8: If P9.X is larger than (P1.X + 0.1), then the gesture command is “Waving the right leg.”

Command 9: If (P7.X + 0.1) is larger than P5.X, then the gesture command is “Putting the right hand on the right waist.”

Command 10: If (P6.X + 0.1) is larger than P4.X, then the gesture command is “Putting the left hand on the left waist.”

Feature-based Kinect SDK-gesture recognition by these 10 recognition rules can be further combined with the decision time-interval scheme for enhancing the system performance. For example, when Kinect SDK-gesture recognition calculation recognizes the operative player’s gesture action as the gesture command “Waving the left-hand,” an time-interval with 5-seconds to pause the overall gesture recognition system can be incorporated by a simple library in the Windows platform, *Sleep(5)*.

III. An Android Application for Simulating Vending Machines by Gesture Recognition Operations

The Kinect-based gesture command recognition system mentioned in Section 2 is used for the application of a vending machine control simulation. Figure 4 shows an integration that Kinect-captured gesture command made by the user for controlling the android application of vending machines is recognized and then sent to the smart phone device with the Android platform through the Bluetooth wireless transmission. The APP of the vending machine control simulation is developed in the smart phone with the Android platform. The released Android software development kits are used in this investigation [8, 9] to build up the Bluetooth function on the target phone device quickly. In this study, for simplifying the transmission task, a text symbol table composed of ten symbols, each of which denotes the corresponding gesture command, is made. After finishing the recognition task using the Kinect SDK-gesture recognition with a decision time-interval scheme, the symbol in the text symbol table to represent the label of the recognized gesture command will be then found out. The text symbol will then be sent to the smart phone device to

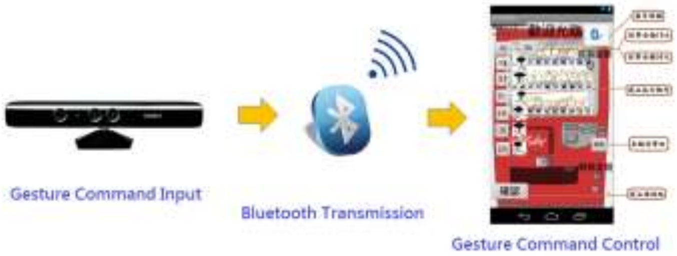

Figure 4. Kinect-captured gesture command made by the user for controlling the android application of vending machines through the Bluetooth wireless transmission.

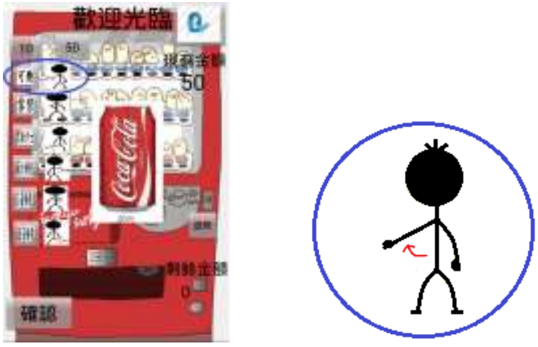

Figure 5. The android application of vending machine simulations where each vending item is indicated by a specific gesture action.

simulate the corresponding vending machine control operation through the Bluetooth wireless network. An interpretation work to translate the received symbol into the practical vending machine operation will also be done on the APP of the smart phone device.

Figure 5 depicts the android application of vending machine simulations where each vending item is indicated by a specific gesture action. As could be seen in Fig. 5, the action player desires to perform an operation to purchase the drink ‘Cola’ at the vending machine, the player makes the gesture action “Waving the right hand” as indicated by the user interface of the vending machine control APP to complete gesture command control.

IV. Experiments and Results

Kinect SDK-gesture recognition with a decision time-interval scheme for vending machine control simulations was performed in an laboratory office environment. The main hardware devices are a personal computer with the Windows 7 platform, a Kinect sensing device, and a smart phone device with the Android platform. The Kinect sensing device is properly connected to the personal computer, and the released Kinect SDK was also appropriately used for developing the system in this study. The default setting of the image capturing rate in both the Kinect RGB and depth cameras is 30 frames per second.

The gesture database was established from 4 male active users. For each active player, 10 classes of active gestures are requested to be made. Each class of active gesture was made 10 times by the same active user. The gesture database was divided into two parts, the training part and the test part. The training gesture database was used to estimate the optimal value of the decision time-interval incorporated in Kinect-SDK gesture recognition, and the test database was employed for evaluating the recognition performance of the presented method. Note that for system developments of gesture recognition, the utilization of the released Kinect SDK will be helpful to rapidly establish the recognition system of presented feature-based gesture recognition.

Figs 6-8 illustrate these gesture commands “Putting the left hand on the left waist,” “Waving the right hand,” and “Lifting the right hand over the head” made by an action player for purchasing the soft drink, the cookies, and the sarsaparilla, respectively, on the APP of vending machines of the smart device. From Figs. 6-8, it could be seen that the correctly recognized gesture command will be effective for controlling the vending machine simulator. Such the system prototype can be then referenced to develop a practical gesture command-based vending machine. Recognition performances of Kinect SDK-gesture recognition with a decision time-interval of different settings on a application of vending machine control simulations are shown in TABLE I. Observed from TABLE I, Kinect SDK-gesture recognition with a decision time-interval has the better recognition rate than that without a decision time-interval. Among all ten different settings of the decision time-intervals, Kinect SDK-gesture recognition with a decision time-interval of 5 seconds performs best, which achieves the recognition rate of 78.1%. In addition, too small values (1, 2, or 3 sec.) or too large values (8, 9, or 10 sec.) given for the decision time-interval will not be proper, both of which have the dissatisfactory recognition performance. For these 10 designed gesture commands, the 7-th and the 8-th gesture commands have better performances. It could also be seen that each of all ten gesture commands has the satisfactory recognition rate using Kinect SDK-gesture recognition with a proper 5-sec. decision time-interval setting, which obviously demonstrates the superiority of the presented flexible time-interval scheme.

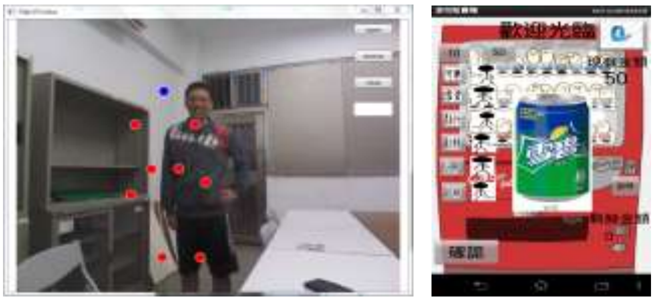

Figure 6. The gesture command “Putting the left hand on the left waist” made by an action player for purchasing the soft drink on the APP of vending machines of the smart device.

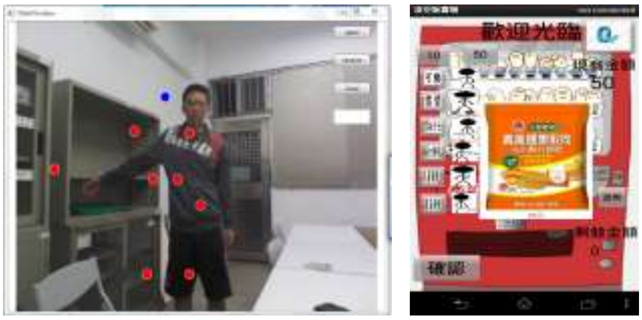

Figure 7. The gesture command “Waving the right hand” made by an action player for purchasing the cookies on the APP of vending machines of the smart device.

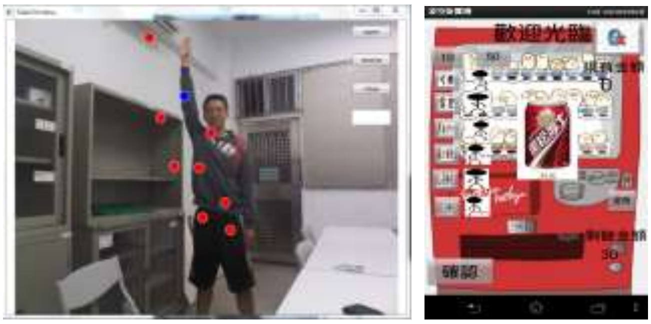

Figure 8. The gesture command "Lifting the right hand over the head" made by an action player for purchasing the sarsaparilla on the APP of vending machines of the smart device.

TABLE I. RECOGNITION PERFORMANCES OF KINECT SDK-GESTURE RECOGNITION WITH A DECISION TIME-INTERVAL OF DIFFERENT SETTINGS ON A APPLICATION OF VENDING MACHINE CONTROL SIMULATIONS

Kinect SDK-gesture recognition with a decision time-interval scheme											
Time-interval settings (sec.)	Recognition rates (%)										
	Gesture commands (10 command in total)										
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	Avg.
1	30	10	10	20	20	30%	60	50	10	20	23.0
2	80	30	20	50	40	40%	70	70	20	40	42.0
3	70	30	10	30	20	60%	80	80	60	30	41.1
4	100	70	80	90	70	70%	100	90	50	80	73.1
5	**100**	**90**	**60**	**70**	**90**	**90%**	**100**	**100**	**80**	**90**	**78.1**
6	80	80	40	60	50	60%	100	100	70	80	66.1
7	90	80	40	30	70	70%	80	100	60	70	62.1
8	70	70	30	40	20	40%	60	70	30	40	43.0
9	40	60	20	20	30	20%	70	30	50	30	35.0
10	20	50	10	20	20	30%	50	40	20	30	26.0

V. Conclusions

In this paper, a Kinect SDK-based gesture recognition with a flexible decision time-interval scheme is proposed for the application of vending machine operations where the APP on the Android platform of the smart phone device is used to be the simulator of the vending machine. That feature-based gesture recognition with the use of the Kinect SDK mainly employs the spatial position feature of the kernel human joints has good recognition performance. Furthermore, the incorporation of the decision time-interval scheme to Kinect SDK-gesture recognition can further increase the recognition rate.

Acknowledgment

This research is partially supported by the Ministry of Science and Technology (MOST) in Taiwan under Grant MOST 103-2218-E-150-004.

References

[1] Z. Zhang, "Microsoft kinect sensor and its effect," IEEE Multimedia, vol. 19, no. 2, pp. 4–10, 2012.

[2] I. Tashev, "Kinect development kit: a toolkit for gesture- and speech based human-machine interaction," IEEE Signal Processing Magazine, vol. 30, no. 5, pp. 129–131, 2013.

[3] I.-J. Ding and C.-W. Chang, "An eigenspace-based method with a user adaptation scheme for human gesture recognition by using Kinect 3D data," Applied Mathematical Modelling, DOI: 10.1016/j.apm.2014.12.054.

[4] I.-J. Ding and C.-W. Chang, "An adaptive hidden Markov model-based gesture recognition approach using Kinect to simplify large-scale video data processing for humanoid robot imitation," Multimedia Tools and Applications, DOI:10.1007/s11042-015-2505-9.

[5] I.-J. Ding and C.-W. Chang, "Feature design scheme for Kinect-based DTW human gesture recognition," Accepted to appear in Multimedia Tools and Applications, June 2015.

[6] I.-J. Ding, C.-W. Chang and C.-J. He, "A Kinect-based gesture command control method for human action imitations of humanoid robots," Proceedings of 2014 International Conference on Fuzzy Theory and Its Applications (iFUZZY2014), pp. 208-211, Kaohsiung, Taiwan, 2014.

[7] "Kinect SDK dynamic time warping (DTW) gesture recognition," https://kinectdtw.codeplex.com/

[8] X. Ducrohet, T. Norbye and K. Chou, "Android studio: an IDE built for Android," Android Developers Blog, Google**,** 2013.

[9] T. Norbye, "Android studio 1.1 preview 1 released," Android Tools, Google**,** 2015.