Chapter 7
Smart Home Appliance Control via Hand Gesture Recognition Using a Depth Camera

Dong-Luong Dinh and Tae-Seong Kim

Abstract The user-friendly and -intuitive interface for household appliances is considered as one of the highly promising fields for researches in the area of smart home and environment. Instead of traditional interface methodologies such as keyboard, mouse, touchscreen, or remote control, users in smart home or environment can control smart appliances via their hand gestures. This chapter presents a novel hand gesture interface system via a single depth imaging sensor to control smart appliances in smart home and environment. To control the appliances with hand gestures, our system recognizes the hand parts from depth hand silhouettes and generates control commands. In our methodology, we first create a database of synthetic hand depth silhouettes and their corresponding hand parts-labelled maps, and then train a random forests (RFs) classifier with the database. Via the trained RFs, our system recognizes the hand parts from depth silhouettes. Finally based on the information of the recognized hand parts, control commands are generated according to our predefined logics. With our interface system, users can control smart appliances which could be TV, radio, air conditioner, or robots with their hand gestures.

Keywords Hand gesture recognition · Depth imaging sensor · Smart appliances · Smart home · Smart environment

D.-L. Dinh
Department of Information Technology,
Nha Trang University, Nha Trang, Vietnam
e-mail: luongdd@ntu.edu.vn

T.-S. Kim (✉)
Department of Biomedical Engineering,
Kyung Hee University, Yongin, South Korea
e-mail: tskim@khu.ac.kr

J. Littlewood et al. (eds.), *Smart Energy Control Systems for Sustainable Buildings*,
Smart Innovation, Systems and Technologies 67,
DOI 10.1007/978-3-319-52076-6_7

7.1 Introduction

Home is an important place of living for people especially the elderly and disabled. Home environment not only impacts on the quality of life, but it is also a place where people spend a large amount of their time. Recently, applying advanced technologies in various fields of architectural, electrical, automatic control, computer, and biomedical engineering to home is getting a lot of attentions from researchers to create smart home. One of the important technologies for smart home is how to control home environments. For instance, interaction via hand gesture is a more intuitive, natural, and intelligent way to interact with household appliances than the traditional interface methodologies using keyboard, mouse, touchscreen, or remote control devices, since users can interface with household appliances with just their hand gestures. Potential applications of such a human interaction based on hand gesture recognition include home entertainments [1], home appliances control [2], home healthcare systems [3, 4], etc. Among these applications, smart control for smart home appliances is one of the important applications for its daily usage: household appliances such as TV, radio, fans, and doors can be controlled by just hand gestures such as changing channels, temperature, and volume [5, 6].

Techniques for hand gesture recognition can be technically divided into two approaches: one is sensor-based and the other vision-based approaches [7] for static hand gestures which are typically represented by the orientation and shape of the hand pose in space at an instant of time without any movement and for dynamic hand gestures which include movement.

In the first approach, inertial sensors such as accelerometers or gyroscopes are typically used which are attached to hand parts to track and estimate their positions, velocity, and acceleration. Then motion features are extracted from measured signals and used for hand gesture recognition. In [8], the features were extracted from the angular velocity and acceleration of triaxial sensors and analysed by fisher discriminant analysis for 3D character gesture recognition. In [9, 10], the changes of accelerations in three perpendicular directions due to different gesture motions were used in real-time as features and then template matching and Hidden Markov Model (HMM) were employed to achieve gesture recognition. These studies had shown some success in recognizing hand sign gestures. However, most studies are focused on recognizing dynamic hand gestures based on motion features acquired through sensor signals, while static hand gesture or pose recognition are still remaining challenges due to motion sensor characteristics. Some applications of the gesture recognition technology in smart home environment have been developed in [11–13] where they used the sensor-based approach for dynamic hand gesture recognition to control home appliances such as radio, TV, and lightings. In these applications, the requirement of the sensor devices as a remote controller makes this kind approach unnatural and inconvenient in spite of its high sensitivity.

In the second approach, vision information from cameras is typically used for hand gesture recognition such as colour, shape, or depth. For static hand gesture or pose recognition which is done by extracting some geometric features such as fingertips, finger directions, and hand contours, or some non-geometric features such as such skin colour, silhouette, and texture. For instance, in [14], a static hand gesture recognition system was presented for nine different hand poses. Orientation and contour features were extracted for the hand gesture recognition. In [15], a real-time hand parts tracking technique was presented by using a cloth glove with various colours imprinted on it. The colour code and position features on the glove were used to track the hand parts for hand pose recognition. Meanwhile, for dynamic hand gesture recognition which is recognized by analyzing a consecutive sequence of recognized static hand poses. For instance, in [16], the key points were extracted as features from hand silhouettes using the Scale Invariance Feature Transform (SIFT) method. Their static hand gesture recognition method used bag-of-features and multiclass Support Vector Machine (SVM), and then a grammar was built to generate gesture commands from dynamic hand gestures to control an application or video game via hand gestures. Since the quality of image captured from RGB cameras is sensitive to the user environment such as noise, lighting conditions, and cluttered backgrounds. These studies have shown limited success in hand gesture recognition. Recently, with an introduction of new depth cameras, some studies used depth images for hand gesture recognition. For instance, in [17–19], the geodesic distances of depth map were utilized and considered as extracted features for hand gesture classification. In [20–23], a novel technique was presented which is one of the most popular and widely used methodologies for hand pose recognition, by directly recognizing hand parts: that is all pixels of given hand depth silhouette were assigned labels which were utilized to recognize hand parts. Based on the recognized parts, hand gestures were detected and recognized. Some applications to control home appliances in smart home using these techniques have been introduced. For example, in [24], an assisting system for the elderly and handicapped was developed to open or close household appliances such as TV, lamps, and curtains by hand gestures where hand poses was captured from three cameras and then recognized by getting their position and direction features. In [25], the authors used some extracted features from depth images which reflect the hand contour information to recognize some static hand poses. Then, dynamic hand gestures were recognized by considering a consecutive sequence of these static poses. Seven dynamic hand gestures were recognized and used for household appliances control.

In this chapter, we present a novel hand gesture recognition and Human Computer Interaction (HCI) system which recognizes each hand part in a hand depth silhouette and generate commands to control smart appliances in smart home environments. The main advantage of our proposed approach is that the state of each finger is directly identified by recognizing the hand parts and then hand gestures are recognized based on the state of each finger. We have tested and validated our system on real data. Our experimental tests achieved 98.50% in recognition of hand gestures with five subjects. Finally we have implemented and tested our HCI system through which one can control home appliances: smart home appliances can be turned on and off; channels and volumes can be changed with just simple hand gestures.

7.2 Hand Gesture-Based Interface System

The setting of our hand gesture recognition and HCI system for appliances control in smart home environments is shown in Fig. 7.1. The system consists of two main parts: a depth camera which is used to get hand depth silhouettes and an appliances control interface which is used to give instructions to appliances. To make a user friendly interface, our hand gesture interface system allows users to interface with the appliances by understanding dynamic hand gestures which are recognized from the hand poses and their movements as described in Tables 7.1 and 7.2.

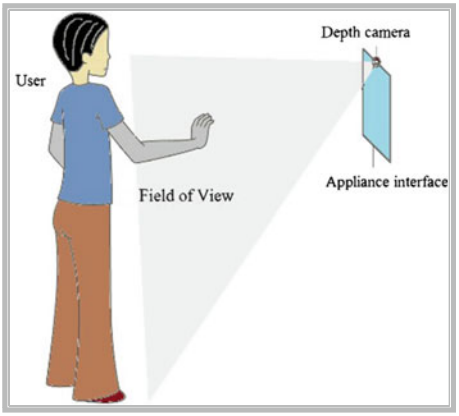


Fig. 7.1 The setting of our proposed hand gesture-based interface system

Table 7.1 Four basic types of hand poses

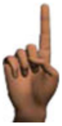
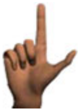
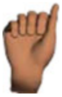
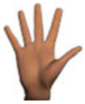
Pose ID	1	2	3	4
Function	Ready	Selection/enter	Exit	Move menu
Hand Poses				

Table 7.2 Types of static and dynamic hand gestures

Gesture types	Functionality of the system	Gesture types	Functionality of the system
1	Recognize Pose 1 to ready the interface system	4	Recognize and track Pose 1 as a hand mouse
2	Recognize Pose 2 to make a selection (i.e., enter)	5	Recognize and track Pose 4 to move the menu right or left
3	Recognize Pose 3 to exit the system	6	Recognize and track Pose 4 to turn up or down volume

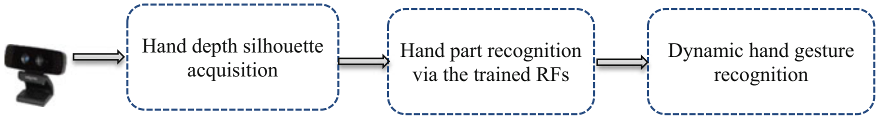

Hand depth silhouette acquisition
Hand part recognition via the trained RFs
Dynamic hand gesture recognition

Fig. 7.2 The framework of our hand gesture recognition system

7.3 Methodology

Overall process of our proposed system for hand gesture recognition, shown in Fig. 7.2, consists of two main parts: in the first part of hand parts recognition, a synthetic hand database (DB), which contains more than thousands of pairs of depth maps and their corresponding hand parts-labelled maps, was generated. Then, the DB was used in training Random Forests (RFs). In the recognition stage, a depth image was first captured from a depth camera and then a hand depth silhouette was extracted by removing the background. Next, the hand parts of a depth silhouette were recognized using the trained RFs. In the second part of hand gesture recognition, a set of features was extracted from the labelled hand parts. Finally, based on the extracted features, hand gestures were recognized by our rules, generating interface commands.

7.3.1 Hand Depth Silhouette Acquisition

In our work, we used a creative interactive gesture camera [26]. This device is capable of close-range depth data acquisition. The depth imaging parameters were set with the image size of 240 × 320, and frame rate of 30 fps. The hand parts were captured in the field of view of 70°.

To detect the hand area and remove background, we used an adaptive depth threshold. The value of threshold was determined based on a specific distance from a depth camera to the hand parts. Hand depth silhouettes were extracted with the background removal methodology mentioned in [26]. The detected and segmented hand is shown in Fig. 7.3.

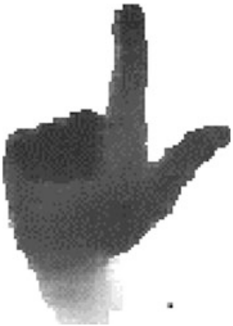


Fig. 7.3 A segmented hand depth silhouette

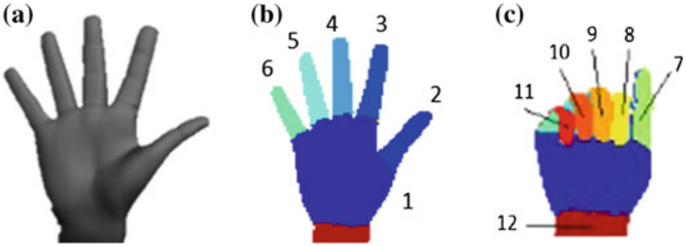


Fig. 7.4 Hand model: **a** 3D hand model in 3Ds-max and **b**, **c** hand parts with twelve labels

7.3.2 Hand Parts Recognition

a. Synthetic hand DB generation

To recognize hand parts from a hand depth silhouette via RFs, the synthetic hand DB, which contains pairs of depth images and their corresponding hand part-labelled maps, is needed to train RFs. We created the DB with a synthetic hand model using 3Ds Max, a commercial 3D graphic package [27]. To identify hand parts, twelve labels were assigned to each hand model as shown in Fig. 7.4 and Table 7.3. The five fingers including the thumb, index, middle, ring, and pinkie fingers, were represented by ten corresponding labels including the five front and five back sides of the fingers. The front parts were coded with the indices of 2, 3, 4, 5 and 6. Likewise, the five back sides were coded with the indices of 7, 8, 9, 10 and 11, respectively. The images in the DB had a size of 320 × 240 with 16-bit depth values.

b. Depth feature extraction

To extract depth features f from pixel p of depth silhouette I as described in [28], we computed a set of depth features of the pixel p based on difference between the depth values of a neighbourhood pixel pair in depth silhouette I. The positions of the pixel pairs were randomly selected on the depth silhouette and they had a relation with the position of the considered pixel p by two terms o_1 and o_2 of the coordinates of x and y, respectively. Depth features, f, are computed as follows:

Table 7.3 A list of the named and labelled parts in hand model

Label no.	Hand parts	Label no.	Hand parts
1	Palm	7	Thumb (back)
2	Thumb (front)	8	Index (back)
3	Index (front)	9	Middle (back)
4	Middle (front)	10	Ring (back)
5	Ring (front)	11	Pinkie (back)
6	Pinkie (front)	12	Wrist

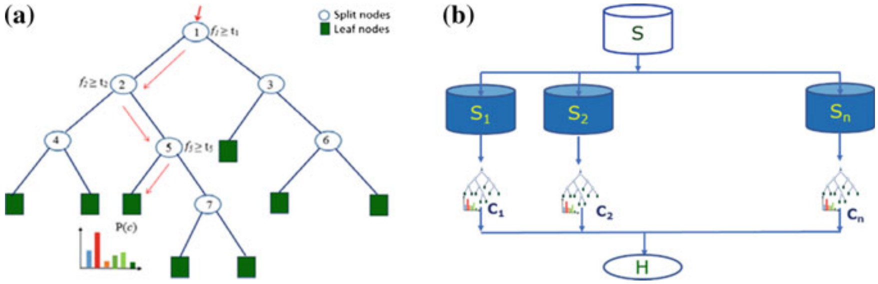


Fig. 7.5 RFs for pixel-based classification: **a** a binary decision tree and **b** ensemble of n decision trees for training RFs

$$f(I,p) = I_d\left(x_p + \frac{o_{1x}}{I_d(x_p, y_p)}, y_p + \frac{o_{1y}}{I_d(x_p, y_p)}\right) - I_d\left(x_p + \frac{o_{2x}}{I_d(x_p, y_p)}, y_p + \frac{o_{2y}}{I_d(x_p, y_p)}\right), \quad (7.1)$$

where I_d (x_P, y_P) is the depth value at the coordinates of (x_P, y_P) and (o_1, o_2) is an offset pair. The maximum offset value of o_1, o_2 pairs was 30 pixels that corresponds to 0.6 meters which was the distance of the subject to the camera. The normalization of the offset by $\frac{1}{I_d(x_p, y_p)}$ ensured that the features are distance invariant.

c. Hand part recognition via RFs

In our work, we utilized RFs for hand parts recognition. RFs are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently with the same sample distribution for all trees in the forest [29]. These concepts are illustrated in Fig. 7.5. Figure 7.5a presents a single decision tree learning process as a tree predictor. The use of a multitude of decision trees for training and testing RFs on the same DB S is described in Fig. 7.5b. The sample sets $\{S_i\}_{i=1}^{n}$ are drawn randomly from the training data S by bootstrap algorithm [29].

In training, we used an ensemble of 21 decision trees. The maximum depth of trees was 20. Each tree in RFs was trained with different pixels sampled randomly from the DB. A subset of 500 training sample pixels was drawn randomly from

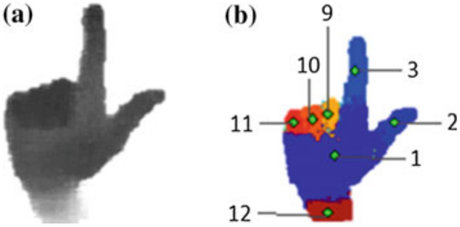


Fig. 7.6 An example of labelled hand parts recognition: **a** a hand depth silhouette and **b** the labelled hand parts

each synthetic depth silhouette. A sample pixel was extracted as in Eq. (7.1), to obtain 800 candidate features. At each splitting node in the tree, a subset of 28 candidate features was considered. For pixel classification, each pixel p of a tested depth silhouette was extracted to obtain the candidate features. For each tree, starting from the root node, if the value of the splitting function was less than a threshold of the node, p went to left and otherwise p went to right. The optimal threshold for splitting the node was determined by maximizing the information gain in the training process. The probability distribution over 12 hand parts was computed at the leaf nodes in each tree. Final decision to label each depth pixel for a specific hand part was based on the voting result of all trees in the RFs.

To recognize hand parts of each hand depth silhouette, all pixels of each hand depth silhouette were classified by the trained RFs to assign a corresponding label out of the 12 indices. A centroid point was withdrawn from each recognized hand part, representing each hand part as illustrated in Fig. 7.6.

7.3.3 Hand Gesture Recognition

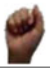
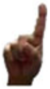
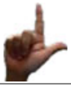
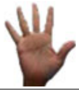
a. Hand poses recognition

From the recognized hand parts, we extracted a set of features. In our labelling, each finger was represented by two different labels: one label for its front side corresponding to the open state of the finger and another for its back side corresponding to the close state of the finger. From the information of the recognized hand parts, we identify the open or close states of each finger. The states of the five labelled fingers were identified and saved as features, namely f_{Thumb}, f_{Index}, f_{Middle}, f_{Ring}, and f_{Pinkie} respectively.

$$f_{Fingers}(i) = \begin{cases} 1 : Open \quad for \quad Label = 2, 3, 4, 5, \ or\, 6 \\ 0 : Close \quad for \quad Label = 7, 8, 9, 10, \ or\, 11 \end{cases} \tag{7.2}$$

For example, as shown in Fig. 7.6b, $\mathrm{f_{Thumb}}$ and $\mathrm{f_{Index}}$ become 1 corresponding to the open state of the fingers. In contrast, $\mathrm{f_{Middle}}$, $\mathrm{f_{Ring}}$, and $\mathrm{f_{Pinkie}}$ become 0 corresponding to the close state of the fingers.

Table 7.4 Recognition rules of the four basic hand poses based on the states of the five fingers

f_{Thumb}	f_{Index}	f_{Middle}	f_{Ring}	f_{Pinky}	Pose type	Hand poses
0	1	0	0	0	1	
1	1	0	0	0	2	
0	0	0	0	0	3	
1	1	1	1	1	4	

0 = Close and 1 = Open

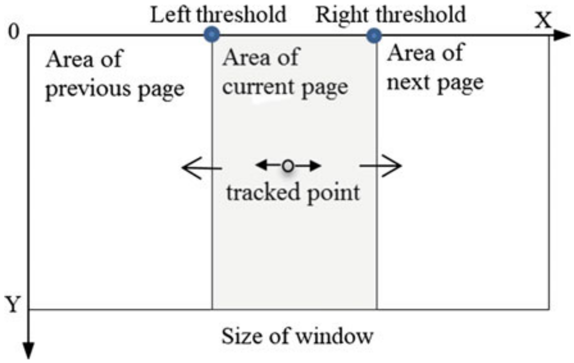


Fig. 7.7 Three divided areas in the screen window to recognize dynamic hand gestures (i.e., Gesture types 5, 6, and 7)

To recognize four basic hand poses, we derived a set of recognition rules. The set of five features from the states of all fingers was used to decode the meaning of the four hand poses. The derived recognition rules are given in Table 7.4.

b. Hand gesture recognition

To understand hand gestures with the recognized hand poses as explained in Table 7.2, after recognizing the hand poses, we tracked their positions. To understand Gestures 1, 2, and 3, our system recognizes Pose 1, 2, and 3, respectively. Then Gesture 4 can be understood by recognizing Pose 1 and then tracking the centroid point position of the index finger in the x and y dimension which gets mapped on the interface screen: it acts as a hand mouse. To understand Gesture 5 which is used to move the menu to the right or left, Pose 4 is recognized and the centroid point position of the palm is tracked in the x dimension between two consecutive frames including the previous and current frames. By dividing the frame window into three sub-areas as presented in Fig. 7.7, if the tracked point of the current frame is moved from the area of current or previous pages to the area of next page, the screen menu slides to the right. Likewise, if the tracked point of the current frame is moved from the area of current or next pages to the area of previous page the screen menu slides to the left. To understand Gesture 6 to turn up or down volume, Pose 4 is recognized and then the centroid point position of the palm is tracked in the y dimension. The difference between the tracked points position of two consecutive frames is used to turn up or down the volume.

7.4 Experimental Results and Demonstrations

7.4.1 Results of Hand Parts Recognition

To evaluate our hand parts recognition quantitatively, we tested on a set of 500 hand depth silhouettes containing various poses over the four hand poses. The average recognition rate of the hand parts was 96.90%. Then, we assessed the hand parts recognition on real data qualitatively, since the ground truth labels are not available. We only performed visual inspections on the recognized hand parts. A representative set of the recognized hand parts are shown in Fig. 7.8.

7.4.2 Results of Hand Pose Recognition

To test our proposed hand poses recognition methodology, a set of hand depth silhouettes was acquired from five different subjects. Each subject was asked to

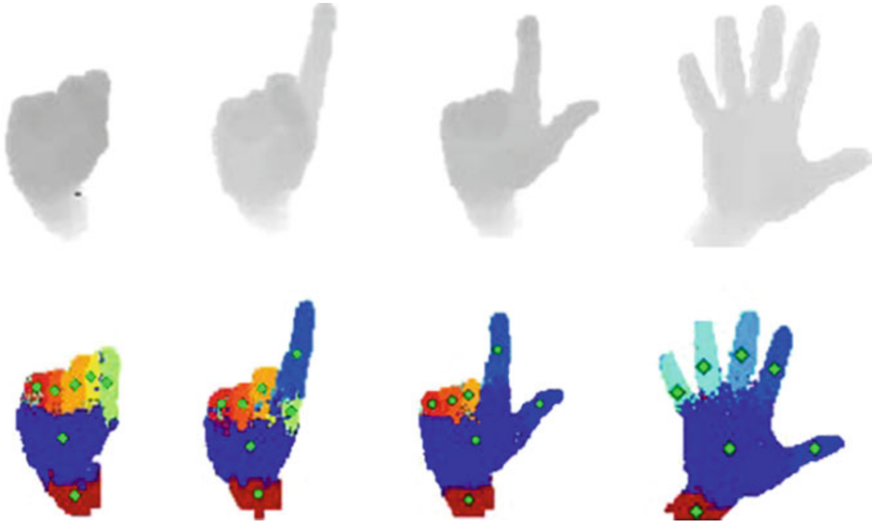


Fig. 7.8 A set of representative results of the recognized hand parts via the trained RFs on real data

make 40 hand poses. Table 7.5 shows the recognition results of the four hand poses in a form of confusion matrix. The mean recognition rate of 98.50% was achieved.

7.4.3 Graphic User Interfaces (GUIs) for Demonstrations

We designed and implemented a GUI interface of our hand gesture interface system as presented in Figs. 7.9 and 7.10. How to use of our system can be explained by the following two examples. In example 1, to open and select any channel for TV, one should use Gesture 1 to open (i.e., activate) the system, use Gesture 4 as a hand mouse to select the TV icon on the GUI screen in Fig. 7.6a, and use Gesture 2 to open the TV. Then the TV channels as shown in Fig. 7.6b opens for a selection of a

Table 7.5 The confusion matrix of the four recognized hand poses using our proposed system

Pose types	1	2	3	4
1	**100**			
2		**100**		
3		2	**98**	
4	1	1	2	**96**

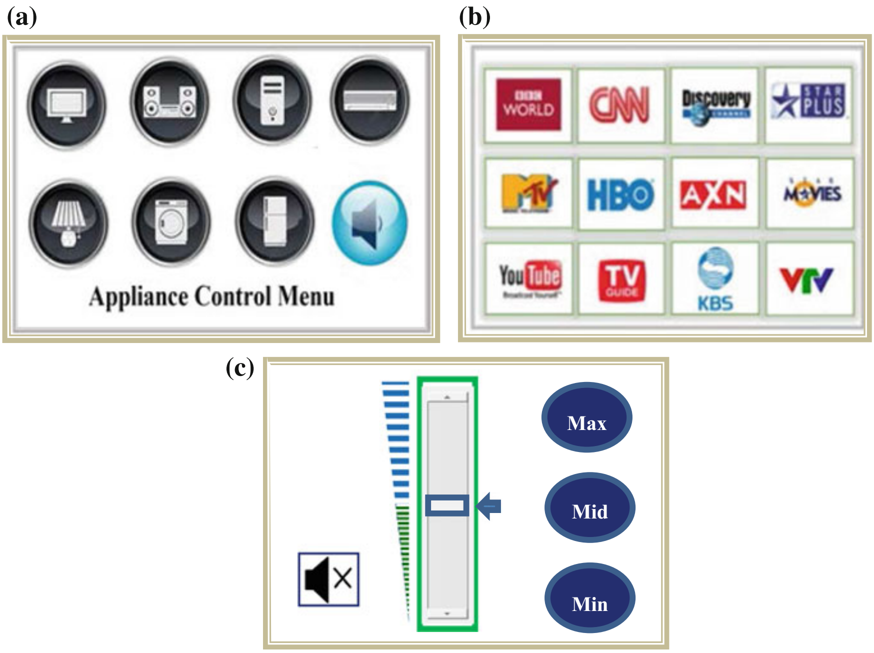


Fig. 7.9 Illustrated gesture-based GUIs of appliance control system. The GUIs for **a** appliances selection, **b** TV channel selection, and **c** volume control

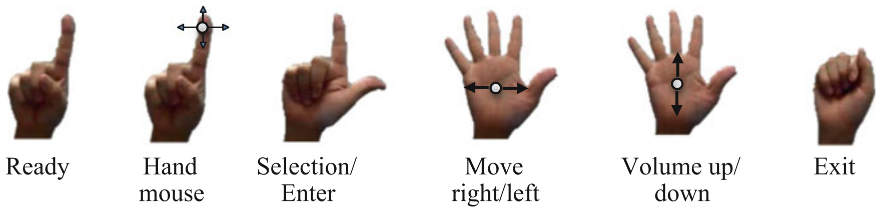


Fig. 7.10 A hand gestures sequence and their tracked points used in our system

channel. To browse the channel menus, one use Gesture 5 to slide the channel pages to the right (i.e., the next page) or left (i.e., the previous page), use Gestures 4 and 2 to select a channel, and finally use Gesture 3 to get back to the main GUI menu as shown in Fig. 7.6a. In example 2, to control volume as well as change temperature, intensity of lightings, or speed of fans, one should use Gesture 1 to open the system, use Gesture 4 as a hand mouse to select the volume icon on the GUI screen, and then use Gesture 2 to open the volume screen as shown in Fig. 7.6c. To change volume, use Gesture 6 to turn up or down volume or use Gestures 4 and 2 to select the fixed minimum (Min), middle (Mid), or maximum (Max) volume levels. To get back to the main GUIs, one can use Gesture 3.

7.5 Conclusions

In this work, we have presented a novel hand gesture recognition system for appliance control in smart home using the labelled hand parts via the trained RFs from a hand depth silhouette. We have achieved the mean recognition rate of 98.50% over the four hand gestures from five subjects. Our proposed hand gesture recognition method should be useful in automation applications for appliance control in smart home environment.

Acknowledgements This work was supported by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as Global Frontier Project (CISS-2011-0031863). This work was supported by International Collaborative Research and Development Programme (funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea) (N0002252).

References

1. Reifinger, S., Wallhoff, F., Ablassmeier, M., Poitschke, T., Rigoll, G.: Static and dynamic hand-gesture recognition for augmented reality applications. In: Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments, pp. 728–737. Springer, Berlin (2007)
2. Shao, J., Ortmeyer, C., Finch, D.: Smart home appliance control. In: Industry Applications Society Annual Meeting, 2008. IAS'08, IEEE, pp. 1–6 (2008)

3. Bien, Z.Z., Park, K.H., Jung, J.W., Do, J.H.: Intention reading is essential in human-friendly interfaces for the elderly and the handicapped. IEEE Trans. Ind. Electron. **52**(6), 1500–1505 (2005)
4. Levin-Sagi, M., Pasher, E., Carlsson, V., Klug, T., Ziegert, T., Zinnen, A.: A comprehensive human factors analysis of wearable computers supporting a hospital ward round. In: 2007 4th International Forum on Applied Wearable Computing (IFAWC), pp. 1–12 (2007)
5. Shimada, A., Yamashita, T., Taniguchi, R.I.: Hand gesture based TV control system—towards both user- and machine-friendly gesture applications. In: 2013 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision, (FCV), IEEE, pp. 121–126 (2013)
6. Bhuiyan, M., Picking, R.: Gesture-controlled user interfaces, what have we done and what's next. In: Proceedings of the Fifth Collaborative Research Symposium on Security, E-Learning, Internet and Networking (SEIN 2009), pp. 25–29. Darmstadt, Germany (2009)
7. Murthy, G.R.S., Jadon, R.S.: A review of vision based hand gestures recognition. Int. J. Inf. Technol. Knowl. Manage. **2**(2), 405–410 (2009)
8. Oh, J.K., Cho, S.J., Bang, W.C., Chang, W., Choi, E., Yang, J., Kim, D.Y.: Inertial sensor based recognition of 3D character gestures with an ensemble classifiers. In: Ninth International Workshop on Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004, IEEE, pp. 112–117 (2004)
9. Zhou, S., Shan, Q., Fei, F., Li, W.J., Kwong, C.P., Wu, P.C., Liou, J.Y.: Gesture recognition for interactive controllers using MEMS motion sensors. In: 4th IEEE International Conference on Nano/Micro Engineered and Molecular Systems, 2009. NEMS 2009, IEEE, pp. 935–940 (2009)
10. Xu, R., Zhou, S., Li, W.J.: MEMS accelerometer based nonspecific-user hand gesture recognition. Sens. J. IEEE **12**(5), 1166–1173 (2012)
11. Mäntyjärvi, J., Kela, J., Korpipää, P., Kallio, S.: Enabling fast and effortless customisation in accelerometer based gesture interaction. In: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, pp. 25–31. ACM (2004)
12. Wan, S., Nguyen, H.T.: Human computer interaction using hand gesture. In: 30th Annual International Conference of the IEEE, Engineering in Medicine and Biology Society, 2008. EMBS 2008, IEEE, pp. 2357–2360 (2008)
13. Ng, W.L., Ng, C.K., Noordin, N.K., Ali, B.M.: Gesture based automating household appliances. In: Human-Computer Interaction. Interaction Techniques and Environments, pp. 285–293. Springer, Berlin (2011)
14. Vieriu, R.L., Goras, B., Goras, L.: On HMM static hand gesture recognition. In: 2011 10th International Symposium on Signals, Circuits and Systems (ISSCS), IEEE, pp. 1–4 (2011)
15. Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. In: ACM Transactions on Graphics (TOG), vol. 28, no. 3, p. 63. ACM (2009)
16. Dardas, N.H., Georganas, N.D.: Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Trans. Instrum. Measur. **60**(11), 3592–3607 (2011)
17. Plagemann, C., Ganapathi, V., Koller, D., Thrun, S.: Real-time identification and localization of body parts from depth images. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 3108–3113 (2010)
18. Molina, J., Escudero-Viñolo, M., Signoriello, A., Pardàs, M., Ferrán, C., Bescós, J., Martínez, J.M.: Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models. Mach. Vis. Appl. **24**(1), 187–204 (2013)
19. Liang, H., Yuan, J., Thalmann, D., Zhang, Z.: Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization. Visual Comput **29**(6–8), 837–848 (2013)
20. Kang, B., Rodrigue, M., Hollerer, T., Lim, H.: Poster: real time hand pose recognition with depth sensors for mixed reality interfaces. In: 2013 IEEE Symposium on 3D User Interfaces (3DUI), IEEE, pp. 171–172 (2013)

21. Keskin, C., Kıraç, F., Kara, Y. E., Akarun, L.: Real time hand pose estimation using depth sensors. In: Consumer Depth Cameras for Computer Vision, pp. 119–137. Springer, London (2013)
22. Luong, D.D., Lee, S., Kim, T.S.: Human computer interface using the recognized finger parts of hand depth silhouette via random forests. In: 2013 13th International Conference on Control, Automation and Systems (ICCAS), IEEE, pp. 905–909 (2013)
23. Zhao, X., Song, Z., Guo, J., Zhao, Y., Zheng, F.: Real-time hand gesture detection and recognition by random forest. In: Communications and Information Processing, pp. 747–755. Springer, Berlin (2012)
24. Bien, Z.Z., Do, J.H., Kim, J.B., Stefanov, D., Park, K.H.: User-friendly interaction/interface control of intelligent home for movement-disabled people. In: Proceedings of the 10th International Conference on Human-Computer Interaction (2003)
25. Wu, C.H., Lin, C.H.: Depth-based hand gesture recognition for home appliance control. In: 2013 IEEE 17th International Symposium on Consumer Electronics (ISCE), IEEE, pp. 279–280 (2013)
26. www.intel.com/software/perceptua
27. Autodesk 3Ds MAX, 2012
28. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Moore, R.: Real-time human pose recognition in parts from single depth images. Commun. ACM **56**(1), 116–124 (2012)
29. Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., Tibshirani, R.: The Elements of Statistical Learning, vol. 2, no. 1, Springer, New York (2009)