International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.9 (2015), pp.219-228
http://dx.doi.org/10.14257/ijsip.2015.8.9.23

Face and Gesture Based Human Computer Interaction

Yo-Jen Tu[1], Chung-Chieh Kao[1], Huei-Yung Lin[1] and Chin-Chen Chang[2]

[1]*Department of Electrical Engineering and Advanced Institute of Manufacturing with High-tech Innovations*
National Chung Cheng University, Chiayi 621, Taiwan
[2]*Department of Computer Science and Information Engineering*
National United University, Miaoli 360, Taiwan
E-mail: ccchang@nuu.edu.tw (corresponding author)

Abstract

In this paper, we present a face and gesture based human computer interaction (HCI) system. We combine head pose and hand gesture to control the system. We can identify the positions of the eyes and mouth, and use the face center to estimate the pose of the head. Moreover, we introduce a technique for automatic gesture area segmentation and orientation normalization of the hand gesture. The user does not need to keep gestures in upright position and the system segments and normalizes the gestures automatically. The experimental results show that the proposed approach is accurate with gesture recognition rate of 93.6%. Also, the user can control multiple devices, including robots simultaneously through a wireless network.

Keywords: *Human computer interaction, Skin color, Face detection, Gesture recognition*

1. Introduction

Human computer interaction (HCI) [1, 2, 11, 14, 18] relies on multiple modalities such as speech, faces or gestures. Faces and gestures are one of the main nonverbal communication mechanisms between humans and computers. Therefore, a real-time processing of faces and gestures is important for HCI. Moreover, in recent years the field of computer vision has been progressed rapidly and the efforts have been made to apply research results in the real-world scenarios. When applying research findings, hardware cost becomes an important issue.

The HCI system can be used towards robot tour guidance, recreational, home and health-care applications. In museums, the traditional keyboard and mouse setup can be replaced with a robot tour guidance system. The robot can detect which exhibitions the visitors are interested in and introduce them directly. This not only makes exhibitions more interesting, but also reduces the tour guidance personnel training cost for the museums. For recreational usage, users can substitute wired controllers with hand gestures and enjoy the hands-free control of electronic devices. In household uses, we can combine head movement with simple hand gestures to control air conditioners, lighting, and other home appliances. It may also be used to aid patients in all kinds of situations when their body mobility is limited.

In this paper, we use a video camera and a PC to develop a face and gesture based HCI system. The proposed HCI system not only can detect face features in head-tilted situations, but also can recognize hand gestures correctly anywhere in the whole image. It is also robust to busy backgrounds and different clothing situations, extracting hand regions, and recognizing hand gestures efficiently using a trained neural network. In applications, we apply the proposed HCI system to a real-life scenario. We give

ISSN: 2005-4254 IJSIP

commands wirelessly to trigger the head movement of the robot. Figure 1 shows the diagram of the proposed HCI system.

The rest of this paper is organized as follows. Section 2 reviews related works. In Section 3, the proposed HCI system is introduced. Section 4 describes the experimental results. Lastly, Section 5 briefly describes conclusions.

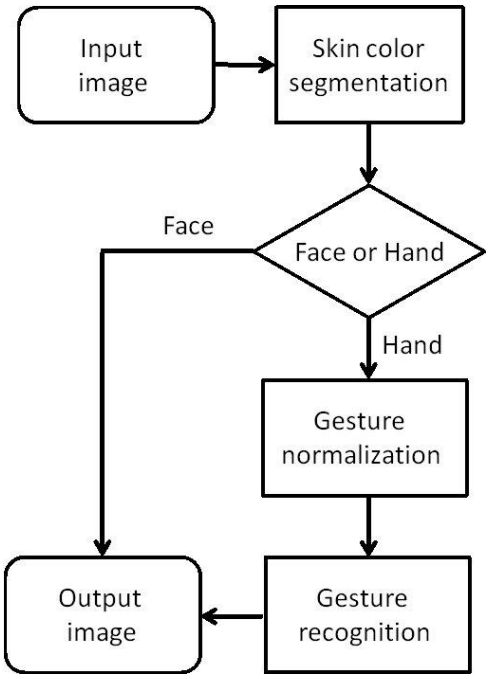

Figure 1. Diagram of the Proposed HCI System

2. Related Works

For face detection, Rowlet et al. [10] presented a neural network-based face detection system. Unlike some works which are limited to detect upright and frontal faces, this system can detect the faces at any degree of rotation in the image plane. Their system uses multiple networks. The first is a router network, which processes each input window to determine its orientation. This information is then used to prepare the window for one or more detector networks. Lee et al. [7] proposed a face detection approach based on the local image region and direct pixel-intensity distributions. The above two works only allow one degree of freedom, which is not suitable for the estimation of head tilt positions.

For head position estimation, Chen et al. [3] presented an approach to estimate the 3D pose of human heads using a single image. Their method only makes use of the information about the skin and hair region of the heads. Yamada et al. [17] proposed a real-time head pose estimation system using an image matching technique. Their system consists of a training stage and a recognition stage for head pose estimation.

For hand gesture recognition, Wachs *et al.* [15] proposed a methodology using a neighborhood-search algorithm for tuning the system parameters. Their system is limited if it is used as a part of HCI systems because it cannot detect the hand gesture locations in the image automatically. The user must restrict the hand gestures in a certain area. Kim *et al.* [6] analyzed the hand gestures with four processes: detecting the hand in bimanual movements, splitting of a meaningful gesture region from an image stream, extracting features and recognizing the

gesture. The user needs to wear long sleeve clothing, exposing only palms and hands to allow the hand gesture recognition to function correctly and properly.

3. The Human Computer Interaction System

In this paper, we develop a face and gesture based human computer interaction system using a single video camera and a PC. We combine the results of both face detection and hand gesture recognition, and show them on the screen to verify the correctness of the detection and recognition results.

3.1. Skin Color Segmentation

The first step is skin color segmentation, which is an important preprocessing stage. We analyze a few skin color detection methods. Color spaces such as NCC r-g [13], RGB [12], YCbCr [5], hue-saturation-value (HSV) [4] color spaces have their pros and cons. We group them into categories based on which color space is used to accomplish skin color detection and select the most suitable one to apply to our HCI system.

The experiments show that the NCC r-g color space gives the best result. This combined skin locus and color space keeps the calculation cost low, and also copes well with the skin color change due to varying lighting conditions. After this analysis we decide to use the NCC r-g color space skin locus model in our HCI system.

For skin color segmentation, first we label the areas of an image using skin colors, which act as candidates for the face or hand. Second, connected components are discovered from these image areas. Third, we set a threshold for the connected components to remove noise and eliminate the areas which are too small to be candidates for the face or hand.

3.2. Face Detection

For face detection, we declare search areas for the eyes and mouth from the remaining succeeded candidates. We search for the eyes using the black and white color feature characteristics. We find the mouth using the distinct redder color tone of the lips compared to face skin. After retrieving the eyes and mouth, we use a simple isosceles triangle geometric shape to find a best match and output the resulting triangle model for the detection of the face.

When developing a real-time face detection system, reducing computational cost is a critical issue. In normal situations, a person's face features locate in fixed relative positions. Thus, we can use this characteristic to define the search windows and search range when searching for them. Figure 2 shows the diagram of the search windows for the eyes and mouth.

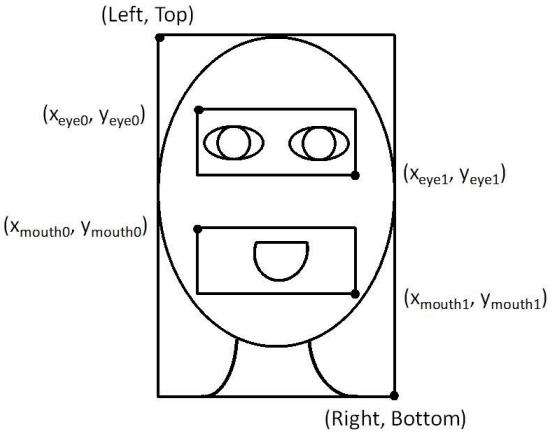

Figure 2. Search Window for the Eyes and Mouth

The search window for the eyes is defined by

$$x_{eye0} = Left + \frac{Right - Left}{10} \text{ and } y_{eye0} = Top + \frac{Bottom - Top}{5},$$

$$x_{eye1} = Right - \frac{Right - Left}{10} \text{ and } y_{eye1} = Top + \frac{Bottom - Top}{2}.$$

The search window for the mouth defined by

$$x_{mouth0} = Left + \frac{Right - Left}{5} \text{ and } y_{mouth0} = Top + \frac{Bottom - Top}{2},$$

$$x_{mouth1} = Left - \frac{Right - Left}{5} \text{ and } y_{mouth1} = Top + \frac{3(Bottom - Top)}{4}.$$

The eyes tend to have a darker tone and have a more distinct characteristic than other face features. If we express this characteristic in color space, the three channels in the RGB color space tend to have similar intensity values. We can distinct the eyes with other face features using this observation. We can extract eyes from other areas of the face by

$$|R - G| + |G - B| + |B - R| < T_{eye},$$

where T_{eye} is the parameter for extracting the eyes.

The RGB channel intensities have values ranging from 0~255. After performing multiple evaluations, setting T_{eye} to 100 is a suitable threshold for extracting the eyes. We can locate the eyes in an image. Moreover, we propose a new method to locate the mouth. Using the previous analyzed skin color information, we dynamically adjust the mouth-extraction threshold values. The specific steps are as follows:

1. From the previous skin color pixels of the image, we calculate the average of *R* and *G* color channel intensities by

$$R_{avg} = \frac{1}{n}\sum_{i=1}^{n} R_i \text{ and } G_{avg} = \frac{1}{n}\sum_{i=1}^{n} G_i,$$

where *n* indicates the total number of skin color pixels in a frame.

2. People's mouth color tends to have a redder tone compared to the rest of the skin. The R/G channel intensity ratio is higher than all other skin color pixels. This can be expressed by

$$1.2 \times \frac{R_{avg}}{G_{avg}} < \frac{R_{mouth}}{G_{mouth}} < 1.5 \times \frac{R_{avg}}{G_{avg}}.$$

Although some unwanted small areas are extracted using the above method, we can locate the exact position of the mouth by retrieving the largest extracted redder tone skin area and calculating the center of gravity of that specific area.

When we extract the eyes from other face features, sometimes eyebrows are extracted accidently at the same time. We define three rules to avoid this problem. Using accurate mouth position data acquired from the previous step, we use the rules below to find the best match for the eyes while eliminating the eyebrows. Figure 3 shows an illustration.

1. The width between eyes D_{eye} is limited by

$$\frac{width}{4} < D_{eye} < \frac{3width}{4} \text{ and } D_{eye} = \sqrt{(x_{eyeR} - x_{eyeL})^2 + (y_{eyeR} - y_{eyeL})^2},$$

where *width* is the width of the face.

2. The distances between the eyes and mouth D_{mouth} are also limited by

$$\frac{width}{3} < D_{mouth} < \frac{3width}{4} \text{ and}$$

$$D_{mouth} = \sqrt{(x_{mouth} - \frac{x_{eyeR} + x_{eyeL}}{2})^2 + (y_{mouth} - \frac{y_{eyeR} + y_{eyeL}}{2})^2} .$$

3. The mouth must lie between the eyes by

$x_{eyeR} < x_{mouth} < x_{eyeL}$.

Using the above rules, we eliminate eyebrows and other unwanted areas. Hence, we can retrieve the correct eye positions.

In our system, the goal is to control the hardware by mixing the eye movement and hand gestures. We now discuss how to recover orientation (pose) of a moving head. Ohayon et al. [9] proposed a method that requires the construction of a head model using 3D points. Although the accuracy is high, it can only process 4 frames per second due high computational cost. Instead, we propose a low complexity method suitable for real-time systems. First we calculate the centroid of the triangle connecting eyes and the mouth. We then compare it with the detected head's centroid. By doing this, we can estimate the tilt position of the head. In Fig. 3, (x_{head}, y_{head}) is the centroid of the detected head, and (x_{face}, y_{face}) is the centroid of the triangle formed by connecting the eyes and mouth with

$$x_{face} < \frac{x_{eyeR} + x_{eyeL} + x_{mouth}}{3} \text{ and } y_{face} < \frac{y_{eyeR} + y_{eyeL} + y_{mouth}}{3} .$$

We calculate the projection distances D_x and D_y to the x and y-axis of points (x_{head}, y_{head}) and (x_{face}, y_{face}), respectively, by

$D_x = x_{head} - x_{face}$ and $D_y = y_{head} - y_{face}$.

Using the projection distances, we can estimate the tilt position of a person's head. For example, if D_x is a positive value, the head tilts to the right. The head tilts to the left if the value is negative. D_y indicates if the person tilts his/her head upward or downward.

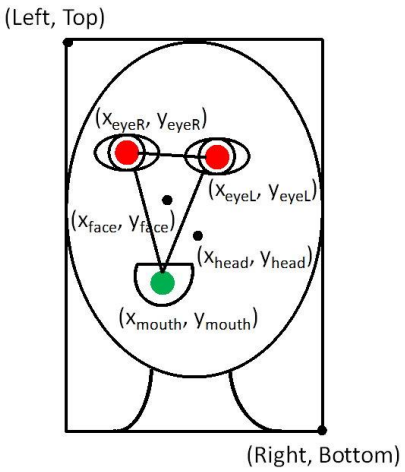

Figure 3. Centroids of Head and Face

3.3. Hand Gesture Recognition

For hand gestures, we eliminate the arm and elbow first, then search for the long-axis of the hand, and normalize the hand to a certain fixed angle. We input the normalized hand to a trained and weighted neural network for hand gesture recognition.

We apply a particular type of neural network model, known as a feed-forward back-propagation neural network [8]. This neural model is easy to understand, and can be easily implemented in image processing tasks. With traditional techniques, one must understand the inputs of the algorithms and the outputs for correct implementation. For a

neural network, you do not have to know these details at all. You simply show the relation of the output associated with the given input. With an adequate amount of training, the network mimics the function that you are demonstrating. With a neural network, it is possible to apply some inputs irrelevant to the solution. During the training process, the network learns to ignore any inputs that do not contribute to the output. If some critical inputs are left in the training process, the network fails to result in a correct solution.

3.3.1. Back-propagation Neural Network: A back-propagation neural network is a kind of feed-forward network. In a 3-layer feed-forward network, the information moves in only one direction, forward, from the input layer (X_1, X_2, X_3, X_4, ...), through the hidden layer (H_1, H_2, H_3, H_4, ...), and to the output layer (Y_1, Y_2, Y_3, Y_4, ...).

Below are specific settings used in our neural network.

1. Network layer: We use the basic 3-layer neural network model for our real-time system. Although increasing layers in a neural network system has a higher accuracy rate, it also increases the complexity of the network. Adding layers will increase the learning and recalling time in the training process, which is not suitable for real-time systems. To increase accuracy, we add more neurons in the hidden layer instead.
2. Hidden layer neurons: A common approach to decide how many neurons are located in this layer is to double the number of neurons in the input layer. We give 16 neurons in the input layer and 30 neurons in the hidden layer.
3. Learning rate: Since learning is usually done in post-process, it will not affect the real-time performance when used. We choose a learning rate of 0.01 to maintain the system's stability.

3.3.2. Hand Gesture Segmentation and Recognition: We use the skin color to locate the hand's position. If a person wears a long sleeve shirt, the captured image area is the hand gesture candidate. But if one wears short sleeve clothing, we need to exclude the whole arm and preserve the hand section only.

From observations, we find the almost fixed width from the elbow to the wrist. Using this characteristic, we can eliminate the arm and capture only parts of the hand needed for hand gesture recognition. We first convert the hand image into a binary image, and then project the binary image onto the y-axis (x-axis indicates black pixel count of the same y-axis value). There is a large difference in terms of the change of the black pixel count from the wrist to the hand. We segment the whole hand based on the difference.

After obtaining the segmented hand, we first compute the hand's long axis that passes the centroid of the hand. Then we rotate the hand to the upright position based on the long-axis. Finally, we normalize the size of the hand image to 40x40. Figure 4 shows (a) the long-axis of a hand, (b) the hand in the upright position, and (c) the normalized hand.

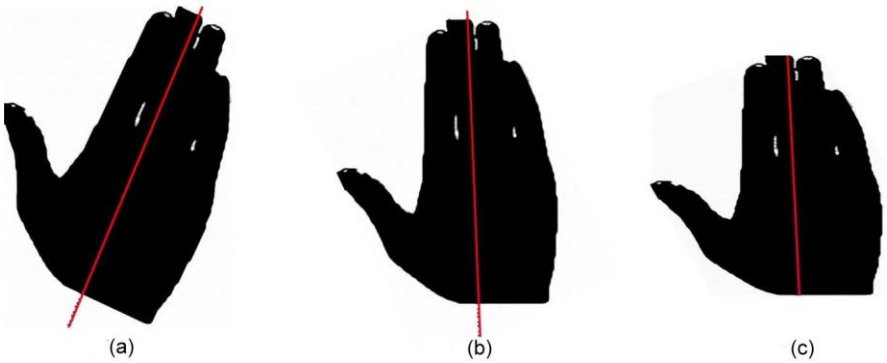

Figure 4. (a) The Long-axis of a Hand, (b) the Hand in the Upright Position, and (c) the Normalized Hand

In the hand gesture normalization process, some gestures will show different results. For example, a "Roll Right" gesture could be normalized to a thumb down or a thumb up position. Figure 5 shows two gestures with different results. This can be solved in the neural network training process. We can allow the output of both these images to have the same output results.

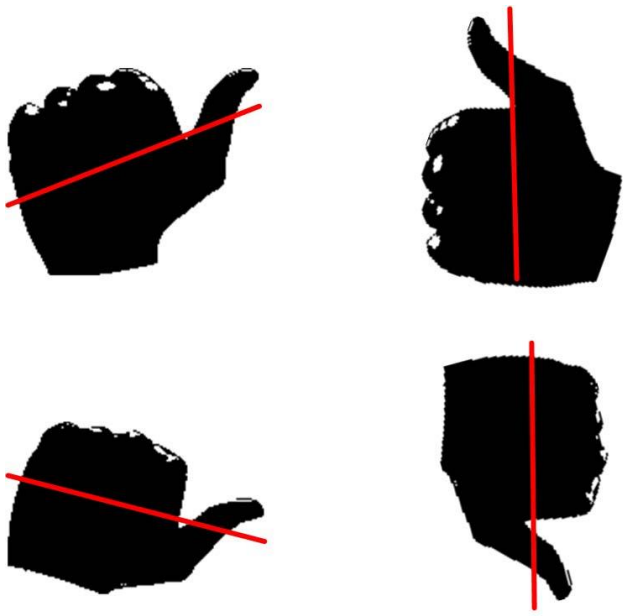

Figure 5. Two Gestures with Different Results

After the above steps, we are ready for hand gesture recognition. Based on the gesture recognition approach of Wagne et al. [16], we segment the hand image into 16 pieces and each piece has 100 (10x10) pixels. We record how many black pixels are in each piece, and normalize the pixel count from 0~100 to 0~1 for the inputs of the neural network. We define 11 hand gestures, and give the input of each hand gesture with 10 images for the training process of the neural network. For consistency, all images used for training are first segmented, and normalized into the upright position beforehand.

4. Results

We divide our experiments into two parts. One part shows the results of face detection and hand gesture recognition for the proposed system. The other part is the experiment controlling the hardware with our system.

4.1. Face Detection and Hand Gesture Recognition

Using Microsoft Visual C++ 6 and OpenCV library for reading and outputting image data, we can process a 320 x 240 resolution image in 0.06 seconds. Hence, we can achieve the real-time calculation with 15 frames per second.

Figure 6 shows the results of face detection. We capture the positions of face features (the eyes and mouth) in multiple head tilted situations. We circle and connect each of them, and show an inverted triangle on the screen. The bounding box marking the face area is achieved by using data retrieved from our face detection. The system can locate face features even if the person is wearing glasses. We test our algorithm using a 300 frame standard video clip "akiyo" [19]. We detect face features correctly in 278 frames, and achieve an accuracy rate of 92.7%.

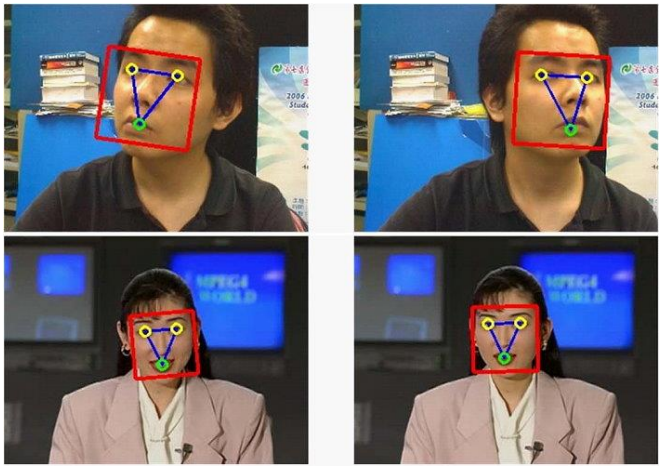

Figure 6. The results of face detection

Moreover, we show some hand recognition snapshots in Figure 7. The name of the recognized hand gestures is labeled under the snapshots. Images can be processed in real-time, and it also shows the reliability estimation outputted by the recalling algorithm of the neural network. It is only displayed to evaluate results efficiently. Reliability estimation indicates how similar it is with the data fed to the neuron network at the training step. We place a colorful poster in the background and our system can extract the hand region correctly even in busy background environments. In some special cases where the hand region were not extracted correctly, the reliability rate decreases greatly. However, due to the high tolerance ability of neural networks, our system can recognize the hand gesture correctly. The proposed approach is accurate with gesture recognition rate of 93.6%.

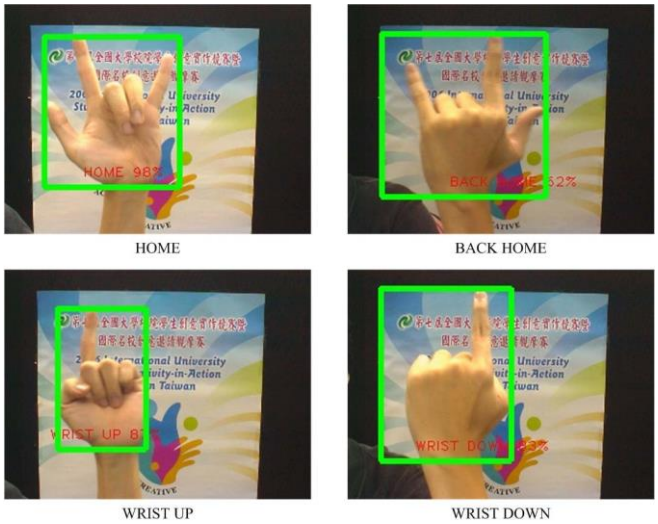

Figure 7. Hand Gesture Recognition Results

4.2. Hardware Control

We design a system where the head of a robot is controlled by hand gestures wirelessly (Figure 8). Using the system setup in the previous experiments, we then add codes to specifically trigger the head movement of the robot. The predefined hand gestures, UP, DOWN, WRIST UP, WRIST DOWN, control the robots head in four directions. The PC with the video camera connected will detect and recognize the hand gestures. Only when correct hand gestures are detected, the PC sends signals to the robot wirelessly to make it turn its head.

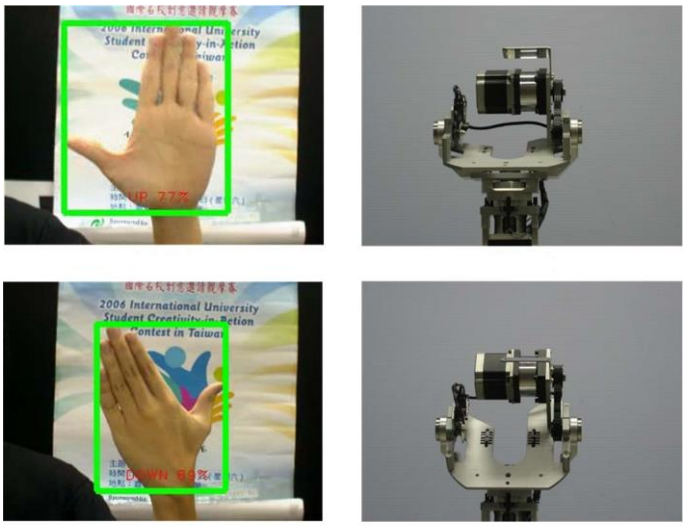

Figure 8. Controlling Movements of Robot's Head Wirelessly Using Hand Gestures

5. Conclusions

We have proposed a human computer interaction system using a PC and a video camera. We control the system using head and hand gestures. The contributions of the proposed system are as follows: a) The face detection can be done in real-time. The centroids of the face and head are used to indicate head tilt directions; b) The user does not need to wear long clothing and keep gestures in upright position; c) The proposed system allows users to control home appliances and hardwares using simple hand gestures and face positions.

Although using NCC r-g color space for skin color detection enables tolerance for different lighting situations, it is still vulnerable when surrounding lighting situations change too constantly. In future studies, we will develop algorithms to overcome this issue. Moreover, the proposed system cannot detect the face correctly when some face features are covered. In future works we will conquer this problem.

Acknowledgements

This paper is a revised and expanded version of a paper entitled, "Human Computer Interaction Using Face and Gesture Recognition," presented at *APSIPA Annual Summit and Conference 2013*, Kaohsiung, Taiwan, October 2013.

References

[1] R. Azad, B. Azad, N.B. Khalifa and S. Jamali, "Real-time human-computer interaction based on face and hand gesture recognition," International Journal in Foundations of Computer Science & Technology (IJFCST), vol.4, no.4, **(2014)**, pp. 37-48.

[2] Q. Chen, M.D. Cordea, E.M. Petriu, T.E. Whalen, I.J. Rudas and A. Varkonyi-Koczy, "Hand-gesture and facial-expression human-computer interfaces for intelligent space applications," Proceedings of IEEE International Workshop on Medical Measurements and Applications, **(2008)**, pp. 1-6.

[3] Q. Chen, H. We, T. Fukumoto and M. Yachida, "3D head pose estimation without feature tracking," Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, **(1998)**, pp. 88-93.

[4] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," IEEE Transactions on Multimedia, **(1999)**, pp. 264-277.

[5] R.L. Hsu, M. Abdel-Mottaleb and A.K. Jain, "Face detection in color images," IEEE Transactions on Pattern Analysis and Machine Intelligence, **(2002)**, pp. 696-706.

[6] K.K. Kim, K.C. Kwak and S.Y. Chi, "Gesture analysis for human-robot interaction," ICACT Advanced Communication Technology, **(2006)**, pp. 1824-1827.
[7] T. Lee, S.K. Park and M. Park, "Novel pose-variant face detection method for human-robot interaction application," Proceedings of IAPR Conference on Machine Vision Applications, **(2005)**, pp. 281-284.
[8] P. Mcollum, "An introduction to back-propagation neural networks," *http://www.seattlerobotics.org/encoder/nov98/neural.html*
[9] S. Ohayon and E. Rivlin, "Robust 3d head tracking using camera pose estimation," Pattern Recognition, **(2006)**, pp. 1063-1066.
[10] H.A. Rowley, S. Baluja and T. Kanade, "Rotation invariant neural network-based face detection," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, **(1998)**, pp. 38-44.
[11] S. Siddharth and A. Rautaray, "Vision based hand gesture recognition for human computer interaction: a survey," Springer Journal Artificial Intelligence Review, **(2012)**, pp. 1-54.
[12] S.Kr. Singh, D.S. Chauhan, M. Vatsa and R. Singh, "A robust skin color based face detection algorithm", Tamkang Journal of Science and Engineering, **(2003)**, pp. 227-234.
[13] M. Soriano, B. Martinkauppi, S. Huovinen and M. Laaksonen, "Using the skin locus to cope with changing illumination conditions in color-based face tracking," Proceedings of IEEE Nordic Signal Processing Symposium, **(2000)**, pp. 383-386.
[14] Y. Tu, C. Kao and H. Lin, "Human computer interaction using face and gesture recognition," Proceedings of IEEE conference on Signal and Information Processing Association Annual Summit (APSIPA), **(2013)**, pp. 1-5.
[15] J.P. Wachs, H. Stern and Y. Edan, "Cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system", IEEE Transactions on Systems, Man and Cybernetics, **(2005)**, pp. 932-944.
[16] S. Wagner, B. Alefs and C. Picus, "Framework for a portable gesture interface", Proceedings of 7thInternational Conference on Automatic Face and Gesture Recognition, **(2006)**, pp. 275-280.
[17] M. Yamada, O. Yamaguchi, A. Nakashima, T. Mita and K. Fukui, "Head pose estimation using adaptively scaled template matching," Proceedings of IAPR Conference on Machine Vision Applications, **(2005)**, pp. 285-289.
[18] A. Zelinsky and J. Heinzmann, "Real-time visual recognition of facial gestures for human-computer interaction," Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, **(1996)**, pp. 351-356.
[19] URL:http://meru.cecs.missouri.edu/free download/videos/

