MATEC Web of Conferences **56**, 02011 (2016) DOI: 10.1051/matecconf/20165602011
*ICCAE 2016*

# NUI framework based on real-time head pose estimation and hand gesture recognition

Hyunduk Kim, Sang-Heon Lee and Myoung-Kyu Sohn

*Department of IoT and Robotics Convergence Research, DGIST, Daegu, Korea*

**Abstract.** The natural user interface (NUI) is used for the natural motion interface without using device or tool such as mice, keyboards, pens and markers. In this paper, we develop natural user interface framework based on two recognition module. First module is real-time head pose estimation module using random forests and second module is hand gesture recognition module, named Hand gesture Key Emulation Toolkit (HandGKET). Using the head pose estimation module, we can know where the user is looking and what the user's focus of attention is. Moreover, using the hand gesture recognition module, we can also control the computer using the user's hand gesture without mouse and keyboard. In proposed framework, the user's head direction and hand gesture are mapped into mouse and keyboard event, respectively.

## 1 Introduction

The history of interaction and interface design is a flow and step from complex interaction to simple interaction between human and computer [1]. The word natural interaction came from Natural User Interface (NUI) that use human body interaction and voice interaction, verbal and non-verbal communication, becoming a one of Human Computer Interaction (HCI) area. It is an evolution from Graphical User Interface (GUI). GUI is the translation from command to graphic for easier purpose for users. Before GUI era, Command Line Interface (CLI) was the starting computer interaction generation which just used codified and very strict command. NUI is a human computer interaction which targeting on some of human abilities such as body movement, touch, motion, voice, vision and using cognitive functions to interact with computer or machine [2-4].

For many human computer interaction applications it would be helpful to know where a user is looking at and what user focus of attention is. Such information can be obtained from tracking the orientation of a human head, or gaze [5]. A person's gaze direction is determined by two factors: the orientation of the head, and the orientation of the eyes. In this paper, we limit our discussion to the head pose orientation. Then the gaze estimation can be formulated as a pose estimation problem. Determining head pose is one of the most important topics in the field of computer vision. There are many applications with accurate and robust head pose estimation algorithms, such as human-computing interfaces, driver surveillance systems, entertainment systems, and so on. For this reason, many applications would benefit from automatic and robust head pose estimation systems. Accurately localizing the head and its orientation is either the explicit goal of systems like human computer interfaces or a necessary pre-processing step for further analysis, such as identification or facial expression recognition. Due to its relevance and to the challenges posed by the problem, there has been considerable effort in the computer vision community to develop fast and reliable algorithms for head pose estimation [6-8].

Recently, hand gesture recognition systems based on vision have undergone an increasingly popularity due to their wide range of potential applications in the field of human computer interaction, such as multimedia application control, video-games, and medical systems. These interfaces are considered more natural, intuitive, friendly, and less intrusive for the user than traditional HCI devices (mouse, keyboard, remote control, etc.). Although the use of keyboard and mouse can be still very useful for some applications, there are situations and applications, where hand-based interfaces can be a key advantage [9].

In general, approaches relying solely on 2D images are sensitive to illumination changes and lack of distinctive features. Since 3D sensing devices have become available, computer vision researchers have started to leverage the additional depth information for solving some of the inherent limitations of image-based method. In this paper, we develop interaction modules based on 3D depth, head pose estimation and hand gesture recognition, and we propose NUI framework based these interaction modules.

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

DOI: 10.1051/matecconf/20165602011

## 2 Head pose estimation module

In this paper, proposed system consist of two process, input system and main system. In the input system, color image and depth map are initially acquired from RGB-D camera, such as ASUS Xtion Pro and Kinect. And then, depth information is converted to real world coordinate system due to estimate the position of head in the real world coordinate system. The main system consists of three processes. In the head pose estimation process, position and orientation of head are estimated using random forests classifier. Due to this classifier, system can be operated in real time and deal with the large set of training data. In the user's gaze tracking process, calculate the position of users gaze on the display device using proposed geometric model. In the Correction process, estimated user's gaze position is corrected using Kalman filter. Due to this process, it is possible to perform stable gaze tracking. Figure 1 shows the structure of head pose estimation module [8].

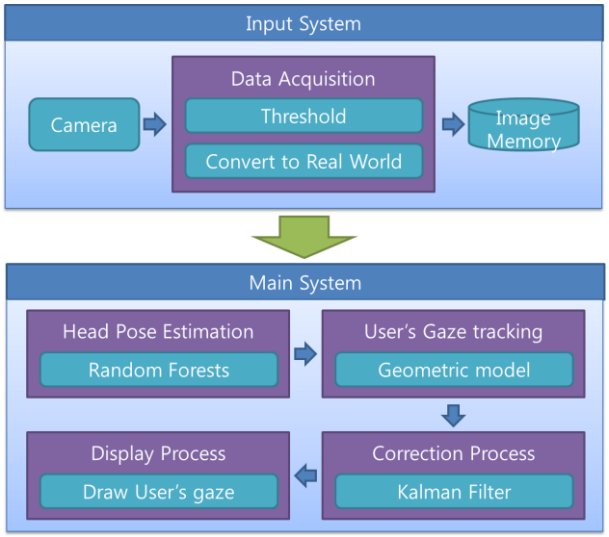


**Figure 1.** The structure of head pose estimation module

### 2.1. Head pose estimation

In this process, we estimate the position and orientation of head using random forests classifier with both hue and depth information. Decision trees can map complex input spaces into simpler, discrete or continuous output spaces, depending on whether they are used for classification of regression purposes. A tree splits the original problem into smaller ones, solvable with simple predictors, thus achieving complex, highly non-linear mappings in a very simple manner. A non-leaf node in the tree contains a binary test, guiding a data sample towards the left or right child node. The tests are chosen in a supervised-learning framework, and training a tree boils down to selecting the tests which cluster the training such as to allow good predictions using simpler models.

Random forests are collections of decision trees, each trained on a randomly sampled subset of the available data; this reduces over-fitting in comparison to trees trained on the whole dataset, as shown by Breiman [10]. Randomness is introduced by the subset of training examples provided to each tree, but also by a random subset of tests available for optimization at each node. Figure 2 shows the example of head pose estimation using random forests [8].

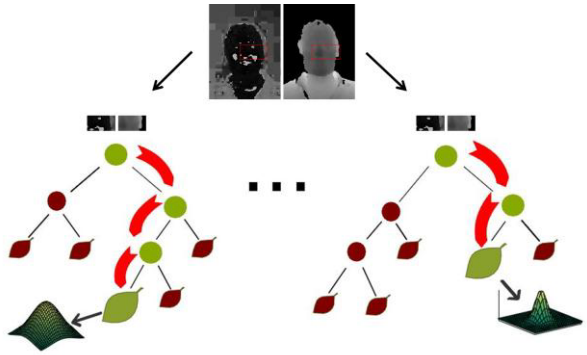

**Figure 2.** Head pose estimation using random forests

### 2.2. Gaze tracking

In this process, we calculate the position of user's gaze on the display device from estimated head position and orientation using geometric model, which consists three steps. Let $(x, y, z)$ and $(\theta_x, \theta_y)$ be the estimated head position and orientation respectively. In real world coordinate system, we assume that camera is located at (0, 0, 0). And then user's gaze position in the real world coordinate system is estimated in the first step. Let $w_1 \times h_1$ be the resolution of RGB-D camera and let $f$ and $p$ be the focal length and pixel size of RGB-D camera, respectively. Then, user's gaze position in pixel coordinate system is converted in the second step. Let $w_2 \times h_2$ be the resolution of diplay device. Then, user's gaze on the display device can be determined in the third stpe. Figure 3 shows the overview of the geometric model for gaze tracking.

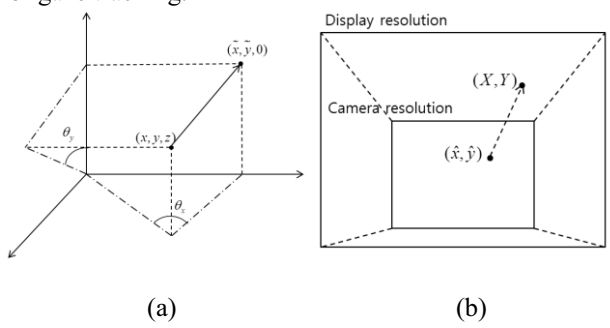


**Figure 3.** Overview of geometric models: (a) Real-world coordinate system, (b) Pixel coordinate system

In general, resolution of RGB-D camera is 640×480 and that of display device is Full-HD or 4K. The estimated user's gaze contains the noise due to the difference between camera resolution and display resolution. In order to solve this problem, we perform correction process using Kalman filter. The Kalman filter has numerous applications in technology because of its convenient form for real time processing and easy to implement given a basic understanding [8].

## 3 Hand gesture recognition module

The Hand Gesture Key Emulation Toolkit (HandGKET) facilitates integration of hand gesture control with PC applications such as games and media center [11]. This middleware recognizes user's hand gestures and generates keyboard or mouse events to control applications in your computer. By editing the gesture-key mapping script, user can control existing off-the-shelf applications which have different control keys. HandGKET consists of three parts. First part is recognition module. This module analysis hand movements and recognize hand gesture. Second part is gesture-key mapping script. When HandGKET runs, first load the gesture key mapping script, and stores it in inter-memory. Third part is event generator, which create keyboard or mouse event according to gesture-key mapping information stored in inter-memory. Figure 4 shows the structure of hand gesture recognition module named HandGKET.

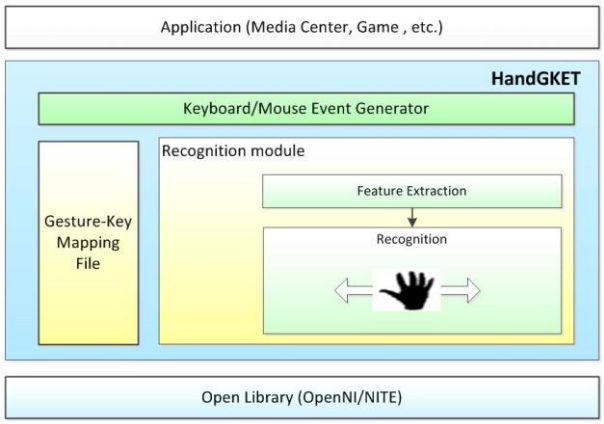


**Figure 4.** The structure of hand gesture recognition module

### 3.1. Recognition module

In the recognition module, hand gesture is recognized using hand trajectory and threshold. The type of hand gesture is defined by round trip gesture. For example, in the case of hand-moving operation to the right direction, hand must is moved to right at the starting point and is returned to the starting point in order to complete the operation.

In the recognition module, hand gesture is recognized using hand trajectory and threshold. The type of hand gesture is defined by round trip gesture. For example, in the case of hand-moving operation to the right direction, hand must is moved to right at the starting point and is returned to the starting point in order to complete the operation.

By setting the starting point, recognition accuracy can be increased in the continuous gesture sequences. The starting point is updated by motion energy in real time and is determined by hand position when the movement of hand is not detected for a certain amount of time after hand is detected. Another advantage of HandGKET is that it can set the threshold value based on real world distance. Despite the same gesture, it can be recognized as different gesture by setting a different threshold value. This makes it possible to increase the number of gestures and to extend the functionality of the toolkit. Figure 5 shows the example of the round trip gesture.

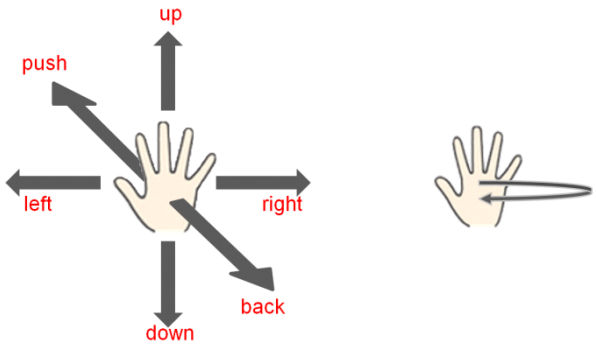


**Figure 5.** The example of round trip gesture.

### 3.2. Gesture-Key mapping script

Figure 6 shows the example of gesture-key mapping script. In the Figure 6, first column means the hand gesture and the threshold value is set in the second column. The hand gesture is recognition if the size of gesture is bigger than threshold value. Third column is type of event, such as key click, double key click, key hold, key toggle, and so on. Fourth and fifth columns mean the generated and released event. Sixth column means additional information, for example, the interval of event time

```
# for one-hand gestures
HAND_LEFT   5  KEY_CLICK  VK_LEFT    0  0
HAND_RIGHT  5  KEY_CLICK  VK_RIGHT   0  0
HAND_RIGHT 30  KEY_CLICK  M          0  0
HAND_UP     5  KEY_CLICK  VK_UP      0  0
HAND_DOWN   5  KEY_CLICK  VK_DOWN    0  0
HAND_PUSH  10  KEY_CLICK  VK_RETURN  0  0
HAND_BACK   7  KEY_CLICK  VK_ESCAPE  0  0
```

**Figure 6.** The example of a gesture-key mapping script.

## 4 Proposed NUI framework

In this paper, we design a NUI framework based on real time head pose estimation and hand gesture recognition. The user's head direction and hand gesture are mapped into mouse and keyboard event, respectively. Using proposed NUI framework, we can control our computer without mouse and keyboard. Figure 7 shows the structure of proposed NUI framework.

The proposed NUI framework consists of two modules, head pose estimation module and hand gesture recognition module. The head pose estimation module first detects face region, and then estimation head position and orientation. The user's gaze is estimated using head position and orientation. By mapping user's gaze into mouse event, we can control contents without mouse device.

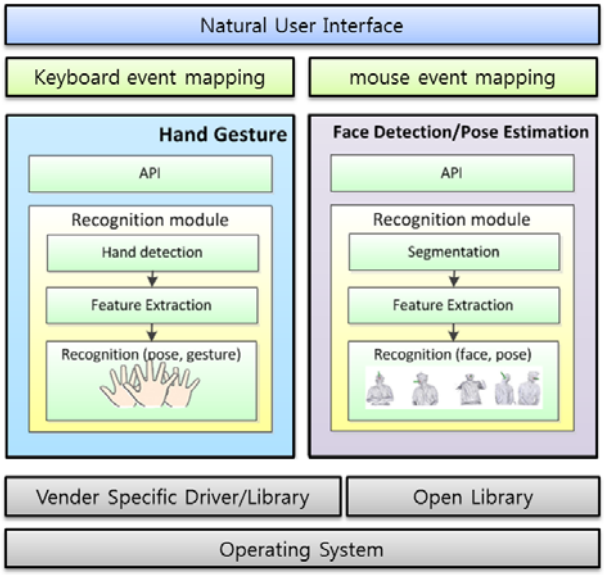


**Figure 7.** The structure of propose NUI framework.

The hand recognition module first detects hand region, and then recognize hand gesture. The hand gesture is mapped into keyboard event when the user takes predefined hand gesture. Using this, we can control contents without keyboard device. Figure 8 shows the application of proposed NUI framework.

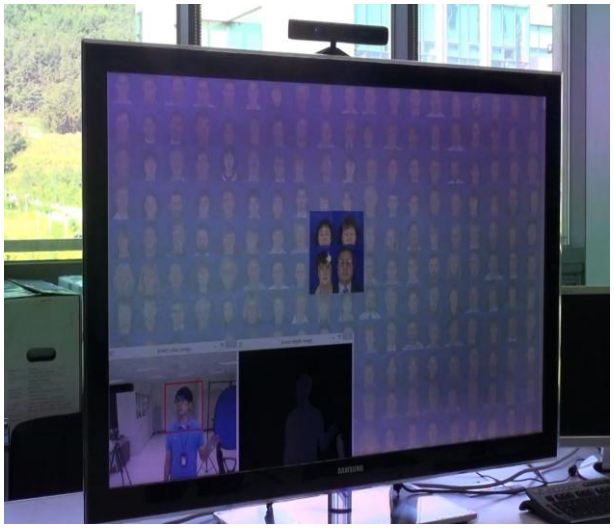

**Figure 8.** The application of proposed NUI framework.

## 5 Conclusion

In this paper, we proposed natural user interface (NUI) framework without traditional devices, such as mouse and keyboard. Proposed NUI frame work consists of two modules, head pose estimation module and hand gesture recognition module. Each module is mapped into mouse and keyboard event. For this reason, user can naturally interact with content and feel the immersive. The future work is development of more natural interface based on head pose and hand gesture as well as voice and emotion.

## Acknowledge

This work was supported by the DGIST R&D Program of the Ministry of Education, Science and Technology of Korea(14-IT-03). It was also supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program (Immersive Game Contents CT Co-Research Center).

## References

1. A. Valli, The Design of Natural Interaction," *Multimedia Tool Applications*, **38(**3), pp. 295-305 (2008)
2. J. Calle, P. Martínez, D. Del Valle, and D. Cuadra, Towards the achievement of natural interaction, *Engineering the User Interface*, Springer, London, pp. 1-9 (2009)
3. A. Del Bimbo, Special issue on natural interaction, *Multimedia Tools and Applications*, **38(**3), pp. 293-294 (2008)
4. W. Xu and E. J. Lee, Human-Computer Natural User Interface Based on Hand Motion Detection and Tracking, *J. of Korea Multimedia Society*, **15(**4), pp. 501-507 (2012)
5. R. Stiefelhagen, J. Yang, and A. Waibel, A model-based gaze tracking system, *Int. J. on Artifical Intelligence Tools*, **6(**2), pp. 193–209 (1997)
6. E. Murphy-Chutorian and M. M. Trivedi, Head pose estimation in computer vision: A survey, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **31**(4), pp. 607–626 (2009)
7. H. Kim, S. H. Lee, M. K. Sohn, and D. J. Kim, Illumination invariant head pose estimation using random forests classifier and binary pattern run length matrix, *Human-centric Computing and Information Science*, **4**(9), pp. 1-12 (2014)
8. H. Kim, M. K. Sohn, D. J. Kim, and N. Ryu, User's Gaze Tracking System and Its Application Using Head Pose Estimation, *IEEE Int. Conf. on Artificial Intelligence, Modelling and Simulation*, pp. 166-171 (2014)
9. A. I. Maqueda, C. R. del-Blanco, F. Jaureguizar, and N. García, Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns. *Computer Vision and Image Understanding*, **141**, pp. 126-137 (2015)
10. L. Breiman, Random forests, *Machine learning*, **45**(1), pp.5–32 (2001.)
11. http://openni.ru/files/handgket/index.html