Journal of Korea Multimedia Society Vol. 15, No. 4, April 2012(pp. 501-507)

Human-Computer Natural User Interface Based on Hand Motion Detection and Tracking

Wenkai Xu[†], Eung-Joo Lee[††]

ABSTRACT

Human body motion is a non-verbal part for interaction or movement that can be used to involves real world and virtual world. In this paper, we explain a study on natural user interface (NUI) in human hand motion recognition using RGB color information and depth information by Kinect camera from Microsoft Corporation. To achieve the goal, hand tracking and gesture recognition have no major dependencies of the work environment, lighting or users' skin color, libraries of particular use for natural interaction and Kinect device, which serves to provide RGB images of the environment and the depth map of the scene were used. An improved Camshift tracking algorithm is used to tracking hand motion, the experimental results show out it has better performance than Camshift algorithm, and it has higher stability and accuracy as well.

Key words: Human-Computer Interaction; Natural User Interface; Hand Tracking; Kinect

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. Figure 1 shows the evolution of NUI.

NUI [2-4] 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. This paper focuses on NUI for human hand motion recognition using Kinect.

For information, many interfaces already developed to improve this area such as Kinect, EyeToy, and Microsoft Surface, 3D Immersive Touch, Dragon Naturally Speaking and Perceptive Pixel.

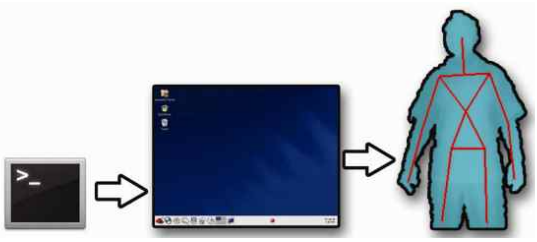


Fig. 1. Evolution of NUI.

※ Corresponding Author : Eung-Joo Lee, Address : Dept. of I.C. Eng., Tongmyong University, 535 YongDang-Dong, Nam-Gu, Busan 608-711, Korea, TEL : +82-51-629-3700, FAX : +82-51-629-3729, E-mail : ejlee@tu.ac.kr
Receipt date : Mar. 23, 2011, Revision date : May 15, 2011
Approval date : June 1, 2011
[†] Dept. of Information Communication Engineering, Tongmyong University
(E-mail: xwk6298@hotmail.com)
[††] Dept. of Information Communication Engineering, Tongmyong University
※ This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the IT/SW NHN Program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2011-C1820-1102-0010).

Today, Kinect is one of the most popular devices used in games in NUI. Kinect is used with Xbox 360 console and play with body movement. According to Microsoft's page, Kinect is designed for a revolutionary new way to play with no controller required. Kinect already proved concepts of "you are the controller"[5].

In this paper, we use open natural interaction (OpenNI) for Kinect, it is possible to use Kinect under Windows OS with all libraries needed for data processing. Both of the libraries are open source and can be used in our testing and implementations. Depth

Image data based on the depth information from Kinect can be measure to get distance from camera to object and ban be manipulate to various things like hand motion detection, motion tracking and so on.

2. RECEIVING DEPTH INFORMATION USING KINECT

Kinect is a new game controller technology introduced by Microsoft in November 2010. Since its launch date it was evident that Microsoft's device is transforming not only computer gaming but also many other applications like robotics and virtual reality, thanks to its ability to track movements and voices, and even identify faces, all without the need for any additional devices [6].

Kinect (Fig. 2) interprets 3D scenes from a continuously projected infrared structure. It has a webcam - like structure and allows users to control and interact with a virtual world through a natural user inter-face, using gestures, spoken commands or presented objects and images. The device includes a RGB color camera, a depth sensor and a multi-array microphone. It provides full-body 3D motion capture, facial and gesture recognition. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, and allows the Kinect sensor to process 3D scenes in any ambient light condition. The depth sensor

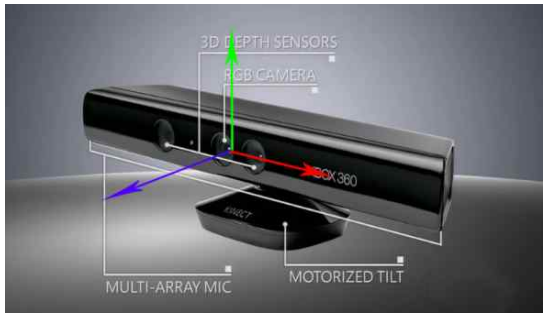


Fig. 2. The Kinect device by Microsoft with the Kinect or Camera Reference Frame. The z-axis is pointing out of the camera(courtesy of Microsoft).

technology was developed by Israeli Prime Sense. It interprets 3D scene information from a continuously-projected infrared structured light. A variant of image-based 3D reconstruction was used to recover the depth of the observed points in the 3D scene.

The approach proposed in this paper is different. We want to use the Kinect technology not to build a digital map of the environment but as an inherent component of the haptic display itself. In particular, we want to use the Kinect sensor to identify the position of the avatar in the virtual environment. In other words, we are not focused on the object but on the organ involved in the touch experience: the human hand.

Kinect has been developed for Xbox 360; however, independent developers also offer solutions for using Kinect separate from the game console and for the most common operating systems. Using the CLNUI Platform or the OpenNI Platform (by Prime Sense), it is possible to use Kinect under Windows OS with all libraries needed for data processing. In this paper, we used the CLNUI Platform which has a simple set of functions to retrieve data from the device (depth raw data, color depth data, color RGB data) and to control the motor of Kinect.

The most relevant data to this study are taken from Kinect's depth sensor. These are 11 bit values and represent the raw value d_{raw} of the depth of

a point p of the 3D scene. According to the calibration procedure developed in [7,8], one gets

$$d = K * \tan(Hd_{raw} + L) - O \quad (1)$$

Where $H = 3.5 \times 10^{-4} rad$, $K = 0.1236m$, $L = 1.18rad$, and d is the Kinect camera depth of p expressed in m. This tangential approximation has a sum of squared difference of $0.33cm^2$ for the calibration data. Once the depth has been obtained using the calibration function above, we can recover the complete coordinate vector for point p in the Kinect camera frame. Let (i, j) be the coordinates (pixels) of the projection of point p onto the Kinect camera frame (Fig. 1). Let (x, y, z) be the coordinates of the 3D point p in the camera frame expressed in m.

In reference [9], the authors proposed the following equations to compute vector (x, y, z) from projection (i, j) and depth d for point p,

$$x = (i - c_x) f_x d \quad (2)$$

$$y = (j - c_y) f_y d \quad (3)$$

$$z = d \quad (4)$$

In reference [9], the authors found that the accuracy for point p reconstruction from the Kinect depth camera is lower than 1cm. For more details on the accuracy of measurements that one can get with this sensor we refer the reader to the Kinect node of the ROS project at MIT.

$f_x = 0.5942143$, $f_y = 0.5910405$, $c_x = 3339.3078$ and $c_y = 242.739$

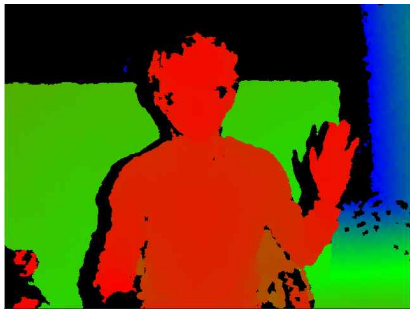
Read the Kinect depth map and build a grayscale image of the 3Dscene. For each point with image projection (i, j) the 11bits depth variable read from the Kinect sensor is converted in an 8bits depth variable. In Fig. 3 an example of the depth image is reported.

3. HAND MOTION TRACKING

3.1 Preprocessing

To prepare the data for our processing, some basic pre-processing is needed. In the depth image

(a)



(b)

Fig. 3. Test Depth Image: (a) Color Image, (b) Depth Image.

taken by the Kinect, all the points that the sensor is not able to measure depth are offset to 0 in the output array. We regard it as a kind of noise. To avoid its interference, we want to recover its true depth value. We suppose that the space is continuous, and the missing point is more likely to have a similar depth value to its neighbors. With this assumption, we regard all the 0 pixels as vacant and need to be filled. We use the nearest neighbor interpolation algorithm to fill these pixels and get a depth array that has meaningful values in all the pixels. Then we use median filter with a 4×4 window on the depth array to make the data smooth.

3.2 Human Motion Detection and Tracking by Kinect

Camshift tracking algorithm based on color performs well in solving the bottom problems of computer vision. Due to its robust and real-time quality, Camshift has become a basic tracking method

which can adapt to the continuous variation of the shape and size of the target, compute fast and has strong anti-jamming capability, guaranteeing the stability and real-time of the system. Camshift algorithm is a dynamic change in the distribution of the density function of the gradient estimate of non-parametric methods. The course of algorithm is as follows [10]:

1. Choose an initial search window W1;
2. Run the Mean-shift algorithm;
3. Resize the search window according to the result of Step (2), and get a new window W2;
4. Use W2 as the initial search window for the next video frame and repeat the algorithm. The flowchart is displayed as below.

Though Camshift algorithm is simple and efficient, it still has a lot of shortcomings, e.g. semi-automatic initialization, low tracking accuracy and notable color markers. Thus, further improvement is needed. Present Camshift algorithm only uses the 2D information, and cannot segment the human movement from background accurately, making the accuracy and stability of the tracking result decline. If depth information can be used, many problems will be solved for most moving objects and background are separated in real scene.

Because the Camshift algorithm is based on color images, tracking error will easily occur when there is similar color in background. Considering the object is usually separated from the surrounding environment in depth, and has fixed moving range, so threshold segmentation in depth map can accurately distinguish the player from the background.

Depth segmentation is shown in Fig. 5: Fig. 5 (a) and Fig. 5 (b) are corresponding color image and input depth map. We can find that depth concentrate on three ranges: [60,95] for human hand area; [95,120] for the human body, it is a little further from the Kinect than hand; most depth map distributes in [120,255], standing for the background which contains a number of irrelevant information, as it is the furthest things from the Kinect. Fig. 5 (c) is the human image after depth threshold segmentation, and in the picture it can easily find background is separated. It is noticeable that making use of depth map can quickly and easily remove the interference from background, and is convenient for the following color tracking.

Existing Camshift methods are generally initialized by artificially calibration, or searching

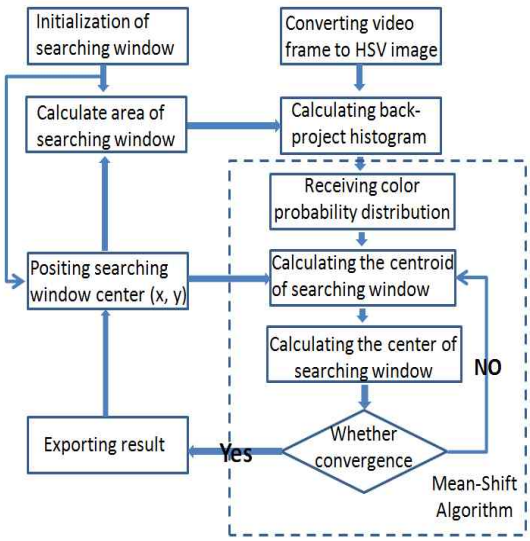
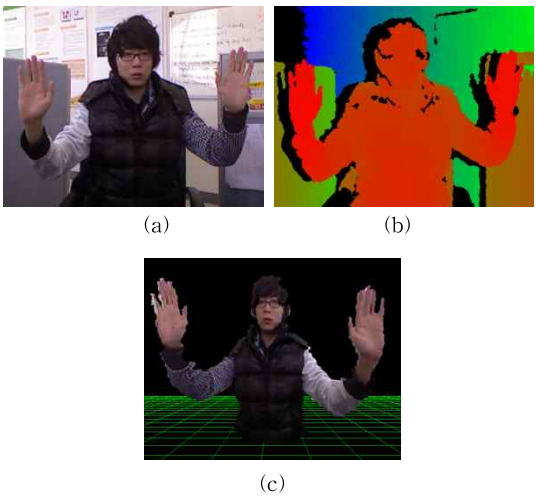


Fig. 4. Camshift Algorithm Flowchart.



(a) (b)

(c)

Fig. 5. Depth Segmentation: (a) Original Image, (b) Depth Image, (c) Human Segmentation.

through the full image with the color model which is stored in the system in advance. But this will bring the problem that the system cannot control where the search window will converge to as there are multiple similar targets to track.

As the Fig. 6, we get another group of pictures. Comparing Fig. 6 (a) and Fig. 5 (a), we can easily find that when the hands are raised forward, the depth of the hands is significantly smaller, because of the distance from hands to Kinect being nearer.

So this paper takes this as the initial sign. It is also accessible to get XOY position of the hands from the depth map, as Fig. 6(c), and take this as the initial searching window in color image, as Fig. 6 (d).

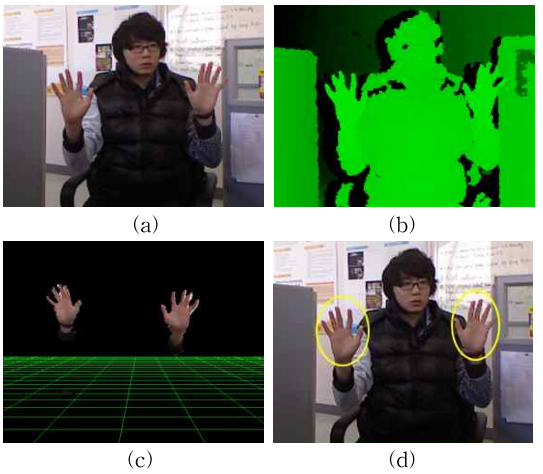


Fig. 6. Initialization Based on Depth Map: (a) Original Image, (b) Depth Image, (c) Hand Area Extraction, (d) Results of Tracking.

4. EXPERIMENTAL RESULTS

The hand tracking system is running on the hardware environment of Intel(R) Core(TM) 2 (2.93GHz), a Kinect camera, and the software environment of Windows 7 and Visual Studio 2008 using OpenNI.

In our experiments, we tested the improved Camshift algorithm and original Camshift tracking method respectively on the computer. Fig. 7 and table.1 show out the results. The system process speed is 30 fps. In the Fig. 7, (a) and (e) are original images, (b) and (f) are depth images, (c) and (g) are the results of improved tracking method we proposed, (d) and (h) are results of original Camshift algorithm. The interval is 25 frames in our test.

From this examination, it is proved that the method we proposed in this paper can exactly track movement of human hands in real time. We can easily find out, the tracking circle using our method

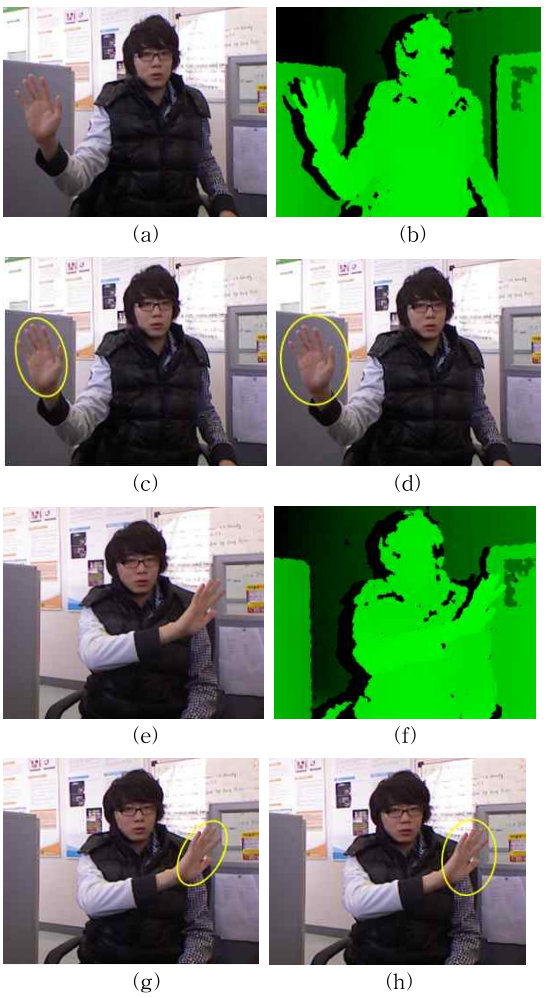


Fig. 7. Comparing experiments: (a), (e) Original Image, (b), (f) Depth Image, (c), (g) Results of improved Proposed Tracking Method, (d), (h) Results of Original Camshift Algorithm.

is smaller than using Camshift and the location of hand movement is more accuracy. The experimental results show out it has better performance than Camshift algorithm, and it has higher stability and accuracy as well.

Table 1. Experimental Results of Proposed Algorithm

	Original Camshift	Improved Camshift
Corr. Rate	96.7%	99.1%
Miss Rate	5.5%	2.3%

5. CONCLUSION

In this paper, we proposed a series of methods to implement human hand detection and hand motion tracking by using Kinect, which is one of the successful device and algorithm in natural interaction that can detect motion of human body. Utilization of depth image data will reduce computational cost because we actually eliminate one step of removing background from object. Thus, the object we concerned can be used directly from the depth information, this also save a lot of processing time. The experimental results show that our methods can track the hand point and recognize dynamic gesture effectively, Nowadays, natural interactions becoming more interesting area and have a bright future to bring our world to the biggest steps in an interaction with virtual environment. In our future work, we plan to develop hand tracking algorithm and continue the research on dynamic gesture recognition.

REFERENCE

[1] Valli. A., "The Design of Natural Interaction," *Multimedia Tools Appl.* Vol.38, No.3, pp. 295-305, 2008

[2] Valli. A., Notes on Natural Interaction, http://www.idemployee.id.tue.nl/g.w.m.rauterberg/Movies/NotesOnNaturalInteraction.pdf, 2005.

[3] Javier Calle, Paloma Martínez, David del Valle, and Dolores Cuadra. "Towards the Achievement of Natural Interaction," *Engineering and the User Interface*, pp. 1-19, 2009.

[4] Del Bimbo, A., "Special Issue on Natural Interaction," *Multimedia Tools and Applications*, Vol.38, No.3, pp. 293-294, 2008.

[5] Kinect Ads: You Are the Controller, http://www.microsoft.com/presspass/features/2010/oct10/10-21kinectads.mspx, 2011.

[6] J. Giles, "Inside the race to hack the Kinect," *The New Scientist,* Vol.208, No.2789, pp. 22-23, 2010.

[7] Open Kinect imaging information, http://openkinect.org/wiki/Imaging_Information, 2010.

[8] Ros Kinect calibration, http://www.ros.org/wiki/kinect, 2011.

[9] Matthew Fisher, Kinect study, http://graphics.stanford.edu/~mdfisher/Kinect.html, 2010.

[10] Wenkai Xu and Eung-Joo Lee, "Hand Gesture Recognition using Improved Hidden Models," *Journal of Korea Multimedia Society*, Vol.14, No.7, pp. 866-871, 2011.

[10] Wenkai Xu and Eung-Joo Lee, "Gesture Recognition Based on 2D and 3D Feature by using Kinect Device," International Conference on Infromation and Security Assurance, Vol.6, No.1, April.28-30, 2012

Wenkai Xu

received his B. S. at Dalian Polytechnic University in China (2006-2010). Currently, he is studying in Department of Information and Communications Engineering Tongmyong University, Korea for master degree. His main research areas are image processing, computer vision, biometrics and hand recognition.

Eung-Joo Lee

received his B. S. , M. S. and Ph. D. in Electronic Engineering from Kyungpook National University, Korea, in 1990, 1992, and Aug. 1996, respectively. Since 1997 he has been with the Department of Information & Communications Engineering, Tongmyong University, Korea, where he is currently a professor. From 2000 to July 2002, he was a president of DigitalNetBank Inc.. From 2005 to July 2006, he was a visiting professor in the Department of Computer and Information Engineering, Dalian Polytechnic University, China. His main research interests includes biometrics, image processing, and computer vision.