Tracking Hand Movements and Detecting Grasp

Ra'na Sadeghi Chegani, Carlo Menon*, *Member, IEEE*

***Abstract*— Post stroke rehabilitation exercises are often repetitive and monotonous. Interactive gaming technology plays an important role of encouraging patients to exercise more and makes rehabilitation exercises less monotonous. In this paper, a real-time method for interaction between human and computer is explored which utilizes Microsoft Kinect to measure hand movements and detect grasp gesture. This approach could be integrated into a computer game which, for instance, simulates the pick and place exercise during rehabilitation of individuals with stroke. The hand's binary image and its corresponding depth data were collected using Kinect. Then a SVM classifier was trained to detect grasp gesture. Trained model was tested online for classifying grasp and non-grasp hand gestures. The trained model was able to detect grasp gesture with 89.1% accuracy.**

Keywords— Microsoft Kinect; SVM; Hand gesture recognition; Hand movement estimation; Hand rehabilitation

I. Introduction

Stroke, also known as cerebrovascular accident, is the fifth cause of death and one of the major reasons of disability in the United States [1]. Somatosensory impairment is a common disability after stroke. While this impairment can affect different parts of the body, 7-53% of patients suffer from hand impairment [2], which affects the patient's ability to effectively and accurately process input received from sensory receptors on the hand's skin. Many rehabilitation exercises are designed to help patients regain their ability to use the hand naturally in daily activities. As suggested by Dawson *et al.* [3], exercises for stroke survival patients should be designed in such a way that do not exhaust patients and encourage them to get involved in the exercises while using the impaired part of the body. Regarding hand disabilities, therapies should be designed in a way that increase the functional movement of the affected hand after each session. One of the commonly-used exercises is grasping small objects, picking them up, moving them for a certain distance in the predefined direction, and placing them on a desired location [3].

Research has shown that recovery from the secondary complications of stroke, such as limited range of motion or strength deficits in the affected limb, is achieved by a high repetition, high intensity, and task specific practice [4]. Usually the number of the rehabilitation sessions is limited to the sessions that patients spend in hospitals and rehabilitation centers, these limited number of sessions may not provide the necessary repetition and intensity needed for recovering natural movement. Accordingly, using rehabilitation exercise in the form of an engaging and motivating game, can help to provide needed repetition and intensity [5]. Also, it has been shown by Cameirao *et al.* that using games can speed up the rehabilitation process [6]. This motivates researches to try to use gaming consoles for designing rehabilitation games and tools [7, 8, 9,10]. This also motivates us to investigate the usefulness of gaming consoles as a user interface for rehabilitation exercises.

One such gaming console is Kinect released by Microsoft. Kinect version 2 has a high resolution color camera and depth sensor, which can provide coordinates in millimeter accuracy with respect to its own coordinate system, as well as red, green and blue channel (RGB) values of each pixel in a color video frame captured by it [11]. In [12] authors have used Microsoft Kinect to obtain input data for a hand rehabilitation game. The rehabilitation game that they designed used depth data and RGB images provided by Kinect as input. The game provides feedback for the patient, in form of score in the game. The patient was engaged with the game, and tried to increase her score in each session.

As mentioned earlier one of the post stroke rehabilitation exercises is grasping small objects, picking them up, moving them for a certain distance and placing them on a specified location. In this paper, a custom program was developed to detect the grasp gesture using the hand's binary image and its corresponding depth data provided by Kinect. The program we developed is able to detect if the hand has reached a desired object (a cup, in this paper), and measure the hand's movement with respect to a certain origin.

The system can detect if participants reach the cup, whether their hand is in grasp gesture, estimate the hand movement in 3D space, and provide feedback for participants. Acquired information can be used as the input data in a game for after-stroke hand rehabilitation. The game can be designed for the exercise of picking up and moving objects.

The remainder of this paper is organized as the following. Section II describes the system, and its testing protocol, section III provides results, and section VI contains discussion and conclusion.

II. Experiment

A. Setup and Protocol

The experiment setup was prepared in a way so that Kinect was placed in the left hand side of participants, on a desk with 2m length. For distinguishing the hand from the rest of the image, participants wore a white glove on the hand and a black

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Canadian Institutes of Health Research (CIHR) and the Michael Smith Foundation for Health Research (MSFHR). Ra'na Sadeghi Chegani and Carlo Menon are with the MENRVA Research Group, School of Engineering Science, at Simon Fraser University, Burnaby, Canada. *Correspondence: Carlo Menon, School of Engineering Science, Simon Fraser University, 8888 University Dr., Burnaby, BC, V5A 1S6, Canada. Email: cmenon@sfu.ca. Tel.: (001)-778-782-6860.

sleeve on the arm to make the best contrast, which helped us to detect the hand area more precisely. It was better that the participant does not wear white clothes as color threshold was used for detecting hand region. Existence of any other white part in the user's clothing could disturb the system's detection procedure. A black fabric covered the table, black paper covered the background, and the cup was covered with a red foam. This way, there was the maximum contrast between the hand, the cup, and the background (Fig.1).

The participant entered the Kinect's workspace, caught the cup, held and moved it in the space. The back of the hand was facing to Kinect all the time.

The location of the cup was calculated in millimeters with respect to the Kinect's coordinate system. Calculating the location of the hand depended on how accurate the color threshold could distinguish the hand region in image. If the RGB threshold range was not wide enough, by moving and changing the lightening condition, the whole hand region was not detectable anymore, and if the threshold's range was set too high, other objects could cause noise and disturb the process of finding the hand's location. The correct range for color threshold's RGB values include the RGB value of hand area in the present lightening conditions.

B. *Data Acquisition*

A custom program was written in NI LabVIEW 2014 to collect data directly from the Microsoft Kinect. Haro3D library, which is a 3D visual library for LabVIEW, was used for programming.

For our objective, both color image and depth data of each frame were needed simultaneously. X, Y, and Z coordinates and RGB value of each pixel in a video frame were provided by a component in the Haro 3D library. Implementation of the image processing sections of the program was done in MATLAB. RGB image was used to find the location of the hand in 2D image, and X, Y, and Z coordinates are helpful for finding the hand's location in 3D space.

To acquire an image region of interest, the background image was subtracted from every frame. The location of the cup was found by getting an average over the coordinate values of the pixels that show the cup. The cup's coordinates was used as the origin for measuring hand movements in 3D space.

A color threshold was applied to each image, and pixels with RGB value in the selected range of color threshold were set to white and the rest of the image is set to black. The result is a binary image. In the binary image, the hand area could be cropped by finding white pixels in the image. Having the hand area, the corresponding depth data could be obtained from the Z values. An average of all the white pixels' coordinate values provides the hand's coordinates. The distance between the hand and the cup was calculated using formula 1. Whenever this distance was less than 5cm, we supposed that the hand has reached the cup.

$$d = \sqrt{(x_{hand} - x_{cup})^2 + (y_{hand} - y_{cup})^2 + (z_{hand} - z_{cup})^2} \quad (1)$$

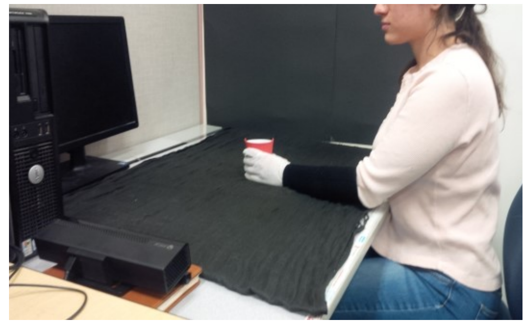

Fig. 1. Participant doing experiment

By subtracting the cup coordinates from the hand coordinates the movement of the hand in each axis was obtained. Fig. 2 shows the over view of the system. In order to detect grasp gesture a support vector machine (SVM) classifier was trained. Participants were asked to put their hand in different locations, and doing any grasp and non-grasp gestures that they want. The binary image and depth map of their hand were recorded, in form of JPG images and Excel files. This provided 443 samples. The size of non-grasp dataset is 187 samples, and The size of grasp dataset is 256 samples.

C. *Train SVM to Detect Grasp*

Pre-processing Samples: As explained in the previous section, for every sample the area around the hand was selected and cropped from the whole image. As a result the sizes of samples were not equal and might vary from 99x84 to 1078x458. To use the data as an input for the training algorithm samples need to be in the same size. To have samples with same size, the size of the smallest sample was chosen as the standard size for all samples, which was 98x71 in our case. Samples were resized to the selected size. Binary images of the hand and their corresponding depth map were used as inputs for training.

Model Selection: SVM classification is a reliable method for hand gesture recognition, where training samples are collected using a camera [6, 13, 14, 15, 16, 17, 18, 19]. In our case SVM model was selected for classifying samples because in our approach collected samples comprised of binary images and depth data, and the objective is to detect grasp, which is a hand gesture. The training stage is an offline process. The testing stage can be done both offline [16, 18, 19] and online [6, 13, 14, 15, 17]. One of the reasons that SVM has been commonly used for hand gesture recognition is its ability to find non-linear decision boundaries which can be learned using kernel trick[18]. In our case a Gaussian function was chosen as the kernel function to find non-linear decision boundaries. Using MATLAB's built-in function to train the SVM classifier, the calculated value for sigma is equal to 104.62009.

D. *Online Testing*

The hand movement along each axis was measured. The sign of the measured number represents the hand movement direction.

Participants were asked to grasp the cup and put the hand in the different locations with different wrist angles for 3 minutes;

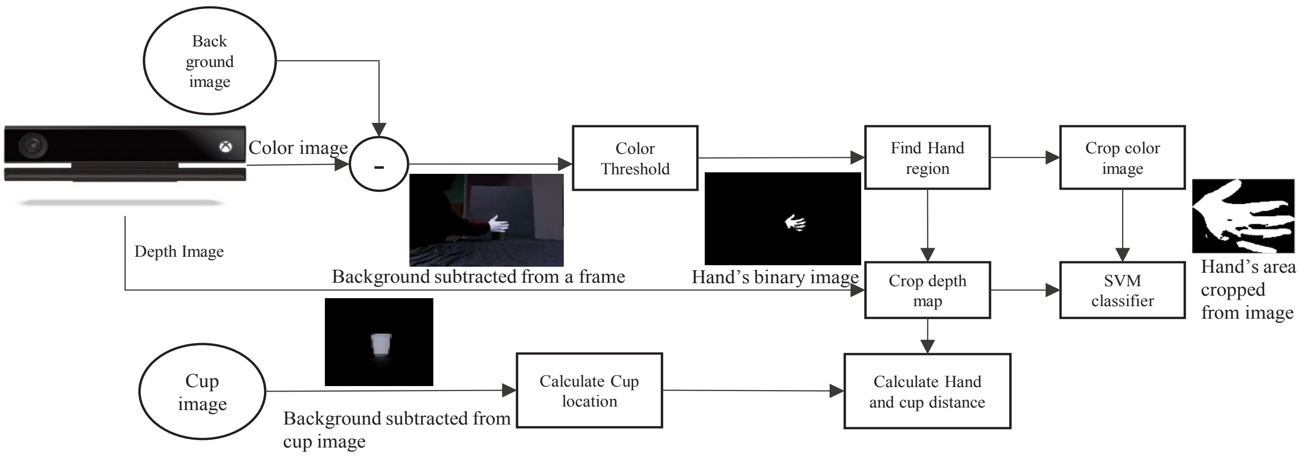

Fig.2. System overview

175 frames were captured this way. While the hand was in the grasp gesture, the number of gestures detected as grasp and non-grasp were counted. The same thing was done for non-grasp gesture.

III. Results and Discussion

In each frame the movement of the hand in 3D space with respect to the cup's initial location was measured by subtracting the hand coordinates and the initial coordinates of the cup, Fig. 3 shows the hand movements.

The reach detection program measures the distance between the hand and the cup in each frame (Fig. 4). Whenever the distance is less than 5cm a Boolean indicator on the monitor turns green. This shows that the program is working correctly. The problem with reach detection is that it does not update the cup's location. For example if user holds the cup and moves it 15cm in Z axis the Boolean indicator switches off, which wrongly means that the hand has not reached the cup, while the hand is grasping the cup.

During online testing for non-grasp gesture the SVM classifier was able to classify 152 samples correctly. Similarly, 175 grasp samples were collected and the system classified 160 of them correctly. Fig. 5 shows the confusion matrix of online testing. As can be seen from the matrix, the sensitivity of the model is 91.43% and its specificity is 86.86%.

The reason that grasp detection works better than non-grasp detection is that in grasp detection the back of the hand always faces the camera, so there are a limited number of scenarios; hand with closed or semi-closed fingers, which can be in different angles and distance from the camera. In a non-grasp gesture, however, there are many different scenarios and it is not possible to collect samples from all of them. Hand can be in fist gesture, semi-closed, open, or tilted in different angles.

Even a using combination of depth data and color image, there are some misclassified grasp gestures. As can be seen in Fig. 2, by cropping the hand's area, some black areas are visible as well, that are parts of background, which are far away from the hand. SVM classifier treats all input data equally, so there will be no difference between background data and hand data, although the background data are less valuable. In this way two same gestures in two different distance from the camera have different depth data, even after

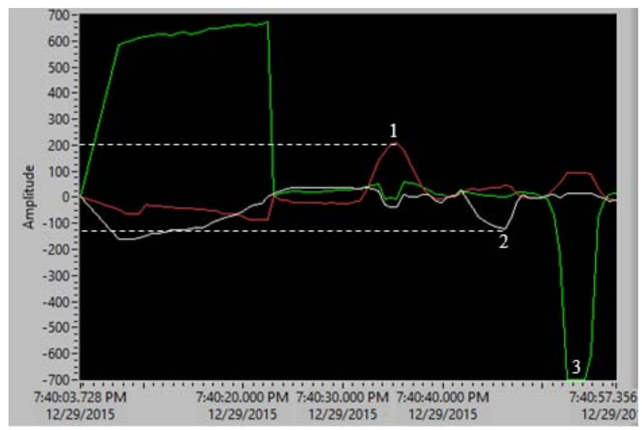

Fig. 3. green, red and white lines show moving in X, Y and Z axis. When the hand reaches the cup all of them have approximately equal values. Point 1: the hand moves 20 cm in Y axis. Point2: the hand moves -12 cm in Z axis. Point 3: the hand moves -70 cm in X axis.

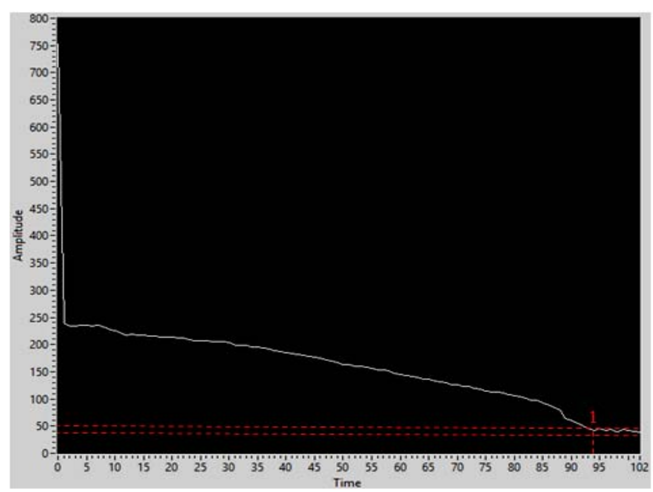

Fig. 4. Distance between the hand and the cup in millimeters vs. time. Point 1: the hand reaches the cup, the distance is approximately 5 cm.

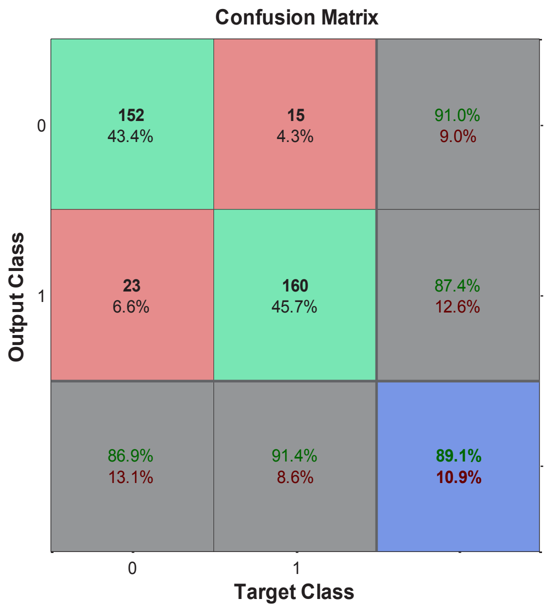

Fig. 5. Confusion matrix of online testing. Zero label is non grasp class and one label is grasp class

normalizing data, which can cause misclassification in grasp gesture. This problem can happen for non-grasp gesture classification as well.

IV. Conclusion

In this paper, a method for detecting grasp gesture was explored and tested online. Microsoft Kinect was used to measure the hand movement, and detect if the hand reaches the cup. In addition, depth data and hand's binary image provided by Kinect was used to train a SVM classifier in order to detect grasp gesture. Using Kinect the hand movement and its direction can be detected in 3D space. Our method for estimating hand movement and detecting grasp gesture could be integrated into a hand rehabilitation game, where patients could perform the pick and place exercise. Although the system can provide some promising result, it still has some limitations. The main Challenge is detecting non-grasp gestures, as it has more various shapes than grasp gesture, which can result in false result.

References

[1] "American Heart Assosiation, American Atroke Assosiation.," [Online].Available: http://www.strokeassociation.org/STROKEORG/AboutStroke/About-Stroke_UCM_308529_SubHomePage.jsp. [Accessed 9 December 2015].

[2] L. Connell, N. Lincoln and A. Radford, "Somatosensory impairment after stroke: frequency of different deficits and their recovery," *Clinical Rehabilitation,* vol. 22, no. 8, pp. 758-767, 2008.

[3] A. Dawson, J. Knox, A. McClure, N. Foley and R. Teasell, "Stroke Rehabilitation," in *Canadian Best Practice Recommendations for Stroke Care*, 2013.

[4] R. Van Peppen, G. Kwakke, S. Wood-Dauphinee, H. Hendriks, P. Van der Wees and J. Dekker, "The impact of physical therapy on functional outcome after stroke: what's the evidence," *Clinical Rehabilitation,* vol. 18, no. 8, pp. 833-862, 2004.

[5] K. Lohse, N. Shirzad, A. Verster, N. Hodges and H. F. M. Van der Loos, "Video Games and Rehabilitation: Using Design Principles to Enhance Engagement in Physical Therapy," *Journal of Neurologic Physical Therapy,* vol. 37, no. 4, p. 166–175, 2013.

[6] M. d. S. Cameirao, S. Bermudez i Badia, E. Duarte and P. F. Verschure, "Virtual reality based rehabilitation speeds up functional recovery of the upper extremities after stroke: A randomized controlled pilot study in the acute phase of stroke using the Rehabilitation Gaming System," *Restorative Neurology and Neuroscience,* vol. 29, pp. 287-298, 2011.

[7] C. Keskin, F. Kirac, Y. Kara and L. Akarun, "Real Time Hand Pose Estimation using Depth Sensors," in *IEEE*, Barcelona, 2011.

[8] J. Antonio, G. Gómez, L. Roberto, A. Mariano and C. Carolina, "Effectiveness of a Wii balance board-based system (eBaViR) for balance rehabilitation: a pilot randomized clinical trial in patients with acquired brain injury," *J Neuroeng Rehabil,* vol. 8, pp. 8-30, 2011.

[9] B. Lange, C.-Y. Chang, E. Suma, B. Newman, A. S. Rizzo and M. Bolas, "Development and Evaluation of Low Cost Game-Based Balance Rehabilitation Tool Using the Microsoft Kinect Sensor," in *33rd Annual International Conference of the IEEE EMBS*, Boston, 2011.

[10] S. N. Tsoupikova D, M. Corrigan, K. Thielbar, R. Vick, Y. Li, K. Triandafilou, F. Preuss and D. Kamper, "Virtual immersion for post-stroke hand rehabilitation therapy," *Annals of Biomedical Engineering,* vol. 43, no. 2, pp. 467-477, 2015.

[11] K. Khoshelham and E. Sander Oude, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors,* vol. 12, no. 2, pp. 1437-1454, 2012.

[12] I. Pastor, H. A. Hayes and S. J. M. Bamberg, "A Feasibility Study of an Upper Limb Rehabilitation System Using Kinect and Computer Games," in *34th Annual International Conference of the IEEE EMBS*, San Diego, 2012.

[13] D.-Y. Huang, W.-C. Hu and S.-H. Chang, "Vision-based Hand Gesture Recognition Using PCA+Gabor Filters and SVM," in *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Kyoto, 2009.

[14] Y.-T. Chen and K.-T. Tseng, "Multiple-angle Hand Gesture Recognition by Fusing SVM Classifiers," in *3rd Annual IEEE Conference on Automation Science and Engineering*, Scottsdale, 2007.

[15] L. Yun and Z. Peng, "An Automatic Hand Gesture Recognition System Based on Viola-Jones Method and SVMs," *2009 Second International Workshop on Computer Science and Engineering,* vol. 2, pp. 72-76, 2009.

[16] Y. Liu, Z. Gan and Y. Sun, "Static Hand Gesture Recognition and Its Application based on Support Vector Machines," in *SNPD '08. Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008.* , Phuket, 2008.

[17] N. H. Dardas and N. D. Georganas, "Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques," *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT,* vol. 60, pp. 3592-3607, 2011.

[18] S. Maji, A. Berg and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *EEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, Anchorage, 2008.

[19] M. Tang, "Recognizing Hand Gestures with Microsoft's Kinect," Palo Alto, 2011.