2015 IEEE/SICE International Symposium on System Integration (SII)
December 11-13, 2015. Meijo University, Nagoya, Japan

Projectable Interactive Surface Using Microsoft Kinect V2: Recovering Information from Coarse Data to Detect Touch

P. Sharma, R. P. Joshi, R. A. Boby, S. K. Saha, *Member, IEEE* and T. Matsumaru, *Member, IEEE*

Abstract—**An Image-projective Desktop Varnamala Trainer (IDVT) called SAKSHAR has been designed to improve the learning by children through an interactive audio visual feedback system. This device uses a projector to render a virtual display, which permits production of large interactive displays with minimal cost. The user's hand is recognized with the help of a Microsoft Kinect Version 2. The entire system is portable, i.e., it can be projected on any planar surface. Since the Kinect does not give precise 3D coordinates of points for detecting a touch, a model of recognition of a touch purely based on the contact of the user's hand with the surface would not yield accurate results. We have instead modeled the touch action by using multiple points along the trajectory of the tracked point of the user's hand while hand makes contact with the surface. Fitting a curve through these points and analyzing the errors is used to make the detection of touch accurate.**

I. Introduction

Primary education is necessary for overall development of a person as well as a society. There are major improvements in social indicators when quality of education improves [1]. The traditional education system based on procedural and rote learning does not help kids to improve basic skills like literacy and arithmetic skills [2]. The use of incomprehensible books makes the children to shy away from learning and may not be suitable for primary education. This might be a major contributing factor for school drop outs in developing countries [3].

In this paper, we are proposing a device which can make learning interactive and full of fun. We are suggesting the use of a virtual touch screen in place of the computer for the primary education of children. This device facilitates a new concept of human-machine interaction for helping younger students to learn.

In recent years, touch screen has become very popular. Samsung's display table known as SUR40 [4], Microsoft's PixelSense [5], and Surface Table [6] are examples of large touchscreens. These devices recognize fingers, hands and objects placed on the screen, enabling vision-based interaction. Though, they deliver high quality of graphics with good interactive experience, they are expensive (about Indian ₹.3,00,000 or US $5000) and not easily transportable. These are important requirements for a system to be used in fields like education in developing and underdeveloped countries. With this purpose in mind "SAKSHAR: Image-projective Desktop Varnamala Trainer (IDVT)" has been designed. The device has been named SAKSHAR which means "Literacy" in Hindi (an Indian language). "Varnamala" is also a Hindi word which stands for complete set of alphabets. We feel that devices like SAKSHAR will give a boost to reading skills and improve literacy.

SAKSHAR is based on SOI (step-on interface) concept, which uses the projected screen as a bidirectional interface. Through this the information is transferred from a robot to the user, and the user instructions are delivered to the robot [7, 8]. Based on SOI concept, a device named IDAT [9-11] was earlier designed for upper limb rehabilitation by training to improve eye-hand coordination. SAKSHAR: IDVT is the next generation of IDAT but designed for use in educational purposes. To lower the total costs, a cheaper RGBD sensor Kinect V2 is proposed in this paper. For using this sensor, a touch detection using statistical information and calibration procedure was developed. The other difference between IDAT device and the current device is that IDAT uses only 2D data along a plane close to the projection surface, whereas the proposed set up it uses 3D information about the hand movement.

Many earlier implementations of interactive surfaces have used a camera for sensing and thus are not portable [12, 13]. In the proposed set up, we have used a RGBD sensor instead and have overcome the shortcomings of inaccuracies in detection due to low resolution using regression analysis. There is one implementation of projectable touch surface using a Kinect called Touchless Touch [14] but the calibration was complex and the results were not satisfactory even after multiple trials of calibration. Unlike other devices such as High Precision Multi-Touch sensing [15] which are custom made for LCD kind of displays we are trying to get rid of these expensive displays. Devices in [16,17] can detect only on plane since the formulation is made on the basis of homography between camera and projected plane. RGBD based sensing can possibly be used to even model gestures in 3D. But this as well as modeling of 2d gestures, may be considered in the future.

The structure of the paper is as follows: Section II discusses the system, whereas regression analysis used for touch detection is explained in Section III. Later, sensor calibration is discussed in Section IV. Section V talks about the game that has been developed for educational purpose.

*Research supported by Waseda University (2014K-6191, 2014B-352, 2015B-346), Kayamori Foundation of Informational Science Advancement (K26kenXIX453), SUZUKI Foundation (26-Zyo-I29) and BRNS, India

P. Sharma, R. P. Joshi, R. A. Boby and S. K. Saha are with the Dept. of Mechanical Engineering, IIT Delhi, Hauz Khas, New Delhi -110016, India; (e-mail: pratyusha1995@gmail.com, ravi2008joshi@gmail.com, ribyab@gmail.com, saha@mech.iitd.ac.in,).

T. Matsumaru is with Bio-Robotics & Human-Mechatronics Laboratory, Graduate School of IPS, Waseda University, Kitakyushu, Fukuoka, Japan, (e-mail: matsumaru@waseda.jp).

The performance of the system and its future scope is discussed in Section VI. Finally the conclusions are presented in Section VII.

II. SAKSHAR SYSTEM

SAKSHAR, as shown in Fig. 1, consists of a projector, Microsoft Kinect Version 2 and a PC. The projector projects graphics rendered by the PC on a projection surface. Kinect, which is mounted on a stand, provides the information about the coordinates at which surface has been touched. The Kinect [18, 19] is a RGBD sensor made by Microsoft. It has built in APIs for the detection of various joints of the body. It has detection range of 0.8 to 3 m [20]. Currently the projector is mounted on an aluminum stand and the Kinect on a camera stand, both of these being portable. A Microsoft .NET based [21] application was used to build the interface between the sensor and the audio/visual system. Multiple threads run simultaneously to gather/analyze sensor data from the user and to give suitable feedback to the user. A game aiding in the learning of alphabets by children was programmed.

We have used Kinect Version 2 for this purpose [22]. The advantage of using a Kinect is that it is 1/8th of the price of a scanning laser range finder (A laser sensor costs about Indian ₹.85,000 (US $1284.10) whereas a Kinect would cost us Indian ₹10,000 (US $151.07)) or highly sensitive cameras and thus more affordable. More importantly, we have the advantage of getting 3D information rather than just the 2D information as in the case of IDAT devices. Ability to gather 3D information enables application of algorithms to even correctly detect the gesture of the user or select one user out of many users. On the other hand, we have overcome the disadvantage of having a poor resolution of data as compared to a laser sensor by adequately supporting it through algorithm. The algorithm used will be explained in Section III.

III. DETECTION OF TOUCH USING REGRESSION ANALYSIS

A. Purpose

Building a sensitive projectable interactive surface that can compete with the existing expensive touch tables available in terms of cost without compromising on the user experience was of primary importance. One serious issue with trying to use RGBD sensor was that we compromised on the resolution of the data obtained.

The problem of detecting a touch on a table top can be solved by looking at the distance between the tip of user's hand and the interactive surface. Modelling a touch based on the height of the hand from the surface can turn out to be highly inaccurate mainly because of looking at the position of the tip of the user's hand at just one point alone. Making a decision in this fashion may be wrong, since significant levels of fluctuations are present in the 3D data point obtained from the RGBD sensor. This is inherent because of the resolution of the camera and image processing routine

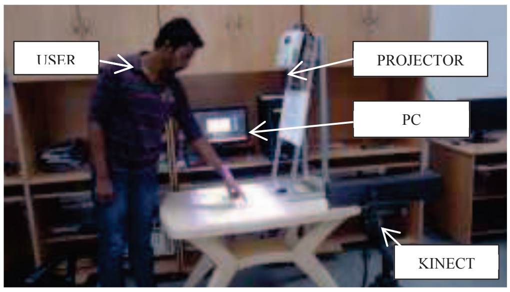


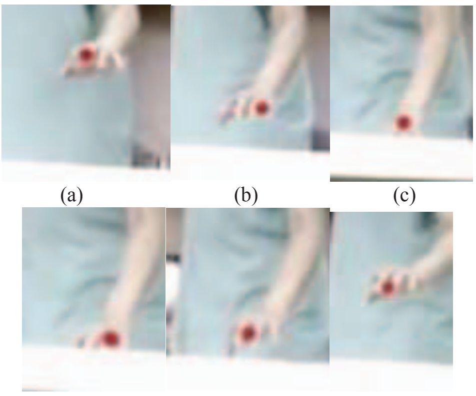
Fig. 1. The current setup

used in the sensor. To make the device sensitive to touch without compromising on the accuracy of different gestures, we built an algorithm which was based on regression analysis performed on the data points obtained from the sensor. A touch was looked at as a hand approaching a surface and touching it and then moving away from the surface. In this case we can use the data points along the direction of movement of the hand towards the point of contact and then moving away from it, as shown in Fig. 2. A best fitting curve through height of the points of approach (Figs. 2(a) and 2(b)), touch (Figs. 2(c) and 2(d)), and separation (Figs. 2(e) and 2(f)) would give us a larger number of relevant data points to analyse statistically. This will hence help us to make a decision based on the information we gathered by looking at the motion of the hand and not the location of the hand at any particular instant of time alone.

B. Overview of the Method Used

The first stage was calibration. Calibration is necessary for the projected screen to behave in accordance with the decisions taken based on data obtained from the Kinect. The Kinect has its own coordinate system defined with respect to its camera, while for the projected surface we had defined another coordinate system. The transformation of a point's coordinates from one system to another wass taken care by the calibration (Section IV). The calibration of the device after setting it up, fixes the plane of projection, coordinate axes and the origin.

Now, as the user's hand was moved over the plane, the joints of the hand were tracked by the Kinect using the inbuilt Kinect Version 2 API. The points of concern were the centre of the hand, the thumb and the tip of the hand. The centroid of the triangle formed by joining these three points was the final point of interest which was used for all further calculations (We shall call this point the virtual centre of the palm). This is shown by the dots in Fig. 2. This minimises the error that might be caused because of inaccurate tracking of any one of these joints. The height of the tip of the finger above the projection surface was constantly calculated (once in every 33.33 milliseconds, which depended on the ability of the Kinect to generate data points [20]). As soon as the tip of the user's hand entered the critical region, which is the region between the projection plane and a parallel plane above it as shown in Fig. 3, the values of the height of the tip



(a) (b) (c)

(d) (e) (f)

Fig. 2. The motion of the user's hand and the points detected.

of the hand from the surface were stored. Depending on how swift or slow the motion of the hand was, the number of data points varied between 20 and 50. We have assumed that the time taken for the user's hand to descend, touch the surface and then ascend takes from anywhere between 0.6 seconds to 1.5 seconds. The Kinect takes one image every 33.33 milliseconds. This would mean that in 0.6 of a second 20 images were taken, and in 1.5 seconds 50 images. The data was accepted for further analysis only if the total number of data points were in this range. Otherwise, the motion was discarded as an unintentional one.

Further analysis of the stored data points included the fitting of a second degree polynomial through the values of height of the tip of the hand from the surface as shown in Fig. 3. The standard deviation of the points from the curve was checked to tell if it was a touch or some other gesture. In Fig. 4, the dotted line is a representation of the value corresponding to three times the standard deviation of the difference in the actual points (shown by crosses) and the curve that was fit (shown in solid line). When we fitted a second degree curve through the data points, we obtained the value of the standard deviation of the points from the curve take many different values based on what the gesture was. This is explained in Section III-C. The location of the touch on the surface was decided based on the point at which the tip of the hand was at the least height from the surface. The coordinates of the touch were the coordinates of the virtual centre of the hand at this least value of height.

C. Judging the Intention

The gesture is considered intentional only when it is not either too fast or too slow, i.e., gestures that are unintentional may be because a person randomly just moved his hand around the area by mistake or was holding onto the surface for too long, and unable to decide. The program does not evaluate gestures that take lesser than 0.6 seconds and more than 1.5 seconds. This was done so as to ensure that we get enough data points for curve fitting and also at the same time get rid of unwanted gestures. This method considers that some people might take a reasonably long time to touch (at most 1.5 seconds) and some might touch really fast (at least 0.6 seconds), but it does not change the range of acceptance from one person to another. This range was found by asking 8 people to touch at their natural speeds and also as fast as they could. The experiment was repeated multiple (about 10) times for every person and the average time taken by the person to touch really fast and at his/her natural speed was considered. Hence, it has been taken into consideration that there is sufficient leverage for a person to change his speed of touching within this limit.

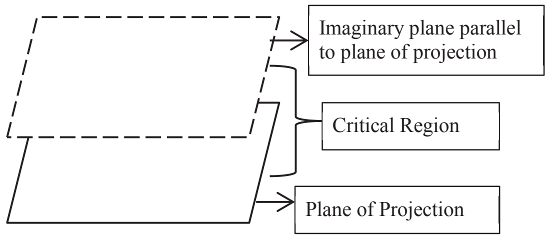


Fig. 3. Depiction of the critical region.

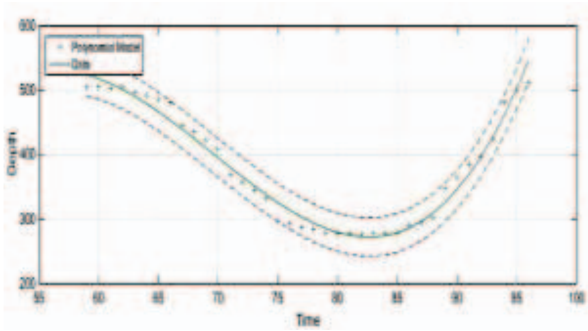


Fig. 4. Best fitting curve drawn through the different heights of the hand from the surface of projection for a touch.

D. Statistical Analysis

If the user's gesture is considered intentional based on the acceptable range of time taken, then the program proceeds with the statistical analysis of the gesture. The data points consist of the height of the virtual center of the palm of the user from the plane of projection. For a touch action the persons hand should first approach the plane of projection and then depart from it after contact. This will mean that the height should first decease, come down to zero and then increase. The point where the height of the user's hand from the surface is the least should be the point of contact.

We fit a second degree polynomial curve through the data points recorded. This was done such that the sum of the squares of the difference between the value given by the polynomial and the data points of actual height of the user's palm from the surface is minimum. Consider the value of height y_d given by

$$y_d(t) = A_1 t^2 + A_2 t + A_3 \qquad (1)$$

Which is a second degree polynomial equation with coefficients A_1, A_2 and A_3, t is time. This is called the regressed value. The difference between this value and the

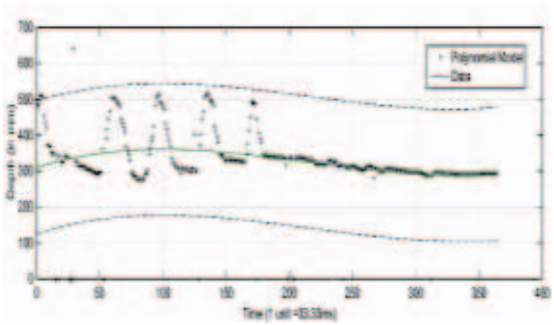


Fig. 5. Data obtained from multiple touches.

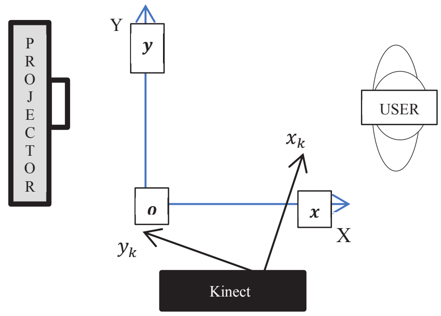


Fig. 6. The coordinate system of the projector and Kinect

actual height at time t corresponding to our i^{th} data point h_{ti}, is denoted as d_i, and expressed as

$$d_i = y_d(t) - h_{ti} \quad (2)$$

The Sum of Squares Regression (SS_R) is the sum of the squared differences between the prediction for each observation and the population mean. There are N such data points at which the difference was evaluated. The value of N varied from 20-50 as discussed earlier (Section III-B).
Hence, we take a summation of the d_i squared across all these data points. This is given below:

$$SS_R = \sum_{i=0}^{N} d_i^2 \quad (3)$$

To be able to fit the best second degree polynomial, we wish to find its coefficients such that the value of SS_R is minimized.

The curve fitting and the spread in the error term obtained i.e. three times the standard deviation is first calculated. The maximum allowable error through experimentation for a reasonable touch was found out. The experiment was performed several times by several different people and the results obtained were similar. Accordingly, we chose a reasonably wide range for acceptance.

If N is the number of data points through which the curve is fit, the statistical distribution of the $SS_R/(\sigma)^2$ is a Chi-Square distribution with n-3 degrees of freedom, where σ is the standard deviation of the data points from their mean values, i.e.,

$$\frac{SS_R}{\sigma^2} = \chi_{n-3}^2 \quad (4)$$

The permissible value of $SS_R/(\sigma)^2$ was found by experimentation and it corresponded to the 90 % confidence interval. Based on the calculations and data obtained for test samples (nearly 100 touches) maximum possible and reasonable error value was fixed (Fig. 5). For different N this would be fixed. Hence, for the reasonable number of data points, these fixed values were stored in an array and the error term was compared with those values in the final stage to check if the touch was intentional or not.

In this way, the statistical analysis takes care of the inaccuracies that might arise if the decision of a touch was made based on just a single 3D point of touch obtained from RGBD sensor with high levels of fluctuations in it.

IV. CALIBRATION

The calibration of the setup involves placing the Kinect at a distance such that the plane of projection is visible and at least the user's torso is visible. The program made for the calibration of the device requires the user to touch various points on the plane as directed by instructions on the screen. The 3D coordinates of the various points touched are hence recorded. The user is asked to place his palm on the points indicated for about 8 seconds and the average of the coordinates obtained is used for calibration to get rid of the possible error in tracked hand position.

The coordinates that the game requires are 2D coordinates such that the origin is fixed at the user's top left, and the horizontal and vertical axes are X and Y axis, respectively, as shown in Fig. 6. A transformation is required to calculate the coordinates of a touch on this plane.

A. Calibration of Kinect with respect to Projection plane

During the initial calibration of the device, the user is asked to place his palm on the four corners of the projected surface. Doing this will give the coordinates of these points in 3D with respect to the Kinect's frame of reference. The top corner point is fixed as the origin and the vectors in the direction of the horizontal line and vertical line can be found out by merely subtracting their corresponding 3D coordinates. The points $\boldsymbol{o}$, $\boldsymbol{x}$ and $\boldsymbol{y}$ correspond to the points shown in Fig. 6.

$$\boldsymbol{o} = [o_1, o_2, o_3] \quad (5)$$

$$\boldsymbol{x} = [x_1, x_2, x_3] \quad (6)$$

$$\boldsymbol{y} = [y_1, y_2, y_3] \quad (7)$$

$$\overrightarrow{OX} = [o_1 - x_1, o_2 - x_2, o_3 - x_3] \quad (8)$$

$$\overrightarrow{OY} = [o_1 - y_1, o_2 - y_2, o_3 - y_3] \quad (9)$$

Similarly, if the coordinates of the point of contact are known we can find the vector joining the origin to the point of contact $\boldsymbol{a}$.

$$\boldsymbol{a} = [a_1, a_2, a_3] \quad (10)$$

$$\overrightarrow{OA} = [o_1 - a_1, o_2 - a_2, o_3 - a_3] \quad (11)$$

The projection of this vector on OX and OY, respectively, will give us the X and Y coordinates of the point $\boldsymbol{a}$ in the plane of projection coordinate system. The projection is as follows.

$$OA_X = \left|\frac{\overrightarrow{OA} \cdot \overrightarrow{OX}}{|OX|}\right| \quad (12)$$

$$OA_Y = \left|\frac{\overrightarrow{OA}\cdot\overrightarrow{OY}}{|OY|}\right| \quad (13)$$

B. Calibration of the Plane

The equation of the plane of projection in the coordinate system of the Kinect can be found out for given coordinates of at least three points on the plane. During calibration, we find out the coordinates of three points of the plane $\boldsymbol{o}$, $\boldsymbol{x}$ and $\boldsymbol{y}$. We can hence find out the equation of the projection plane as

$$Ax + By + Cz + D = 0 \quad (14)$$

where the coefficients are given by

$$A = (x_2y_3 - x_3y_2) \quad (15)$$

$$B = -(x_1y_3 - x_3y_1) \quad (16)$$

$$C = (x_1y_2 - x_2y_1) \quad (17)$$

$$D = -[o_1.(x_2y_3 - x_3y_2) - o_2.(x_1y_3 - x_3y_1) + o_3.(x_1y_2 - x_2y_1)] \quad (18)$$

The plane equation was used to find the height of the palm above the surface in the procedure used after calibration.

C. Projection of Virtual Centre of the Palm

We checked the results obtained and found that the maximum possible error in the location of the touch came to around 2.5 cm. For the game environment, the objects that are visible on the screen and are to be touched are quite bigger than this size. Additionally, a cursor was hence introduced so the user could be sure of what point on the surface does the position of his hand hovering it correspond to. The same principle used in the calibration was used to make this happen.

If $\boldsymbol{a}$ denotes the location of the virtual center of the palm at any instant (Fig. 7), the projection of $\boldsymbol{a}$ on $\overrightarrow{OX}$ and $\overrightarrow{OY}$ will give the coordinates of the projection on the plane.

V. The Game

The game displays a question to find out the correct image of an object. As an example, it asks to 'Find: 'K' for Kettle' (Fig. 8(a)) and displays four available options and the player/learner has to select the correct answer. The game was designed in Hindi language also (Fig. 8(b)). About 4.7 % of total world population in India speaks Hindi (fourth largest spoken language in the world) [23]. In-spite of this fact, there is no suitable library to render Hindi fonts in OpenTK. Hence, we developed a library for rendering Hindi fonts in OpenTK. This library uses System.Drawing component from Microsoft .Net framework to generate a graphics object, in which we write the text. Finally this graphics object is converted into bitmap image and rendered by OpenTK as a texture. This library can be extended for various languages. The system can be used for learning elementary mathematics as well. The game has two operating modes, teaching and testing, which are explained next.

A. Teaching Mode

This mode was designed for targeting kids, who are new to schools. Corresponding to each alphabet, this mode contains basic and well-known objects. In this mode, the user is given unlimited number of attempts for each alphabet.

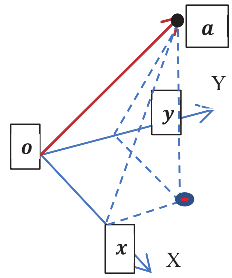


Fig. 7. The cursor shown by the dot.

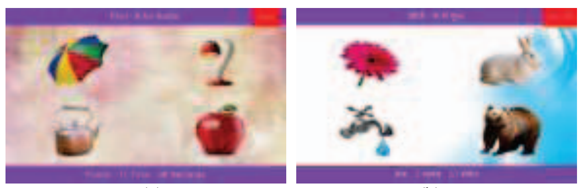
B. Testing Mode

Testing mode contains multiple images for each alphabet, which exposes the user to more number of alphabets. This mode does not provide unlimited attempts. It displays a countdown timer on the basis of which, time available for making choice for a given slide is kept constant. The total score shown in the end can be used to keep track of learning outcome for each user.

VI. Discussions and Further Scope

The proposed method of regression-based touch detection can be looked at as an improvisation over the shortcomings of the hardware, such as inability of the Kinect to give us the exact coordinates of a point, inability to reduce noise, etc. In cases where the quality of the camera used is not very high or when precise decisions have to be made out of data which is not as precise, this method can be used.

The added cursor makes it possible for the user to estimate at every instant what point on the screen a hand is about to touch. On checking the accuracy of this method we found that all the touches by the user were detected correctly. Eight people tried to use this device multiple times and the detection of the touch failed only once in 50 times. Speaking of the precision on the plane of the surface, we looked at the distribution of the points on the plane of projection and looked for spread in the position of the points detected by the Kinect. We took four points in the projection plane and took about 30 readings of the coordinates for each of these points. The average value of the respective coordinates differed from the real value by at most 2 cm. The maximum deviation of the value was about 4 cm from the mean value. The distribution of the points on the plane of the surface can be seen in Fig. 9. In the game the size of the buttons is a square of 4 cm side. This is to ensure that all the



(a) (b)

Fig. 8. Screen shots of the game in (a) English and (b) Hindi.

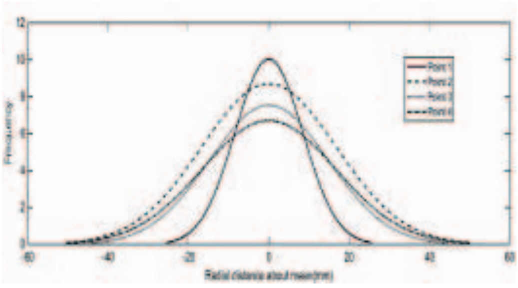


Fig. 9. Result of sample trials: Normal distribution of the distance of points from their actual value

touches are rightly detected.

Some people might touch really fast while others might do it slowly. In future, an algorithm can also be brought in place to adjust the lower limit or upper limit based on the user's average speed of touching from the user's first few countable touches, i.e., we can use the specific user data to fine tune the detection. This would improve the accuracy further and will hence improve the user's experience. It could also be used to make gestures highly user-friendly by incorporating machine learning algorithms and curve fitting techniques to suit every user uniquely by tracking the user's initial few trials.

The method of analysis used can also be extended to other types of gestures like swiping where the linear motion of the hand can be analyzed using regression. As of now analysis for only the click type of gesture has been tested but for gestures like swiping we can look at the equation of lines traced out by the user and look at their deviations from a straight line and analyze the error term. For gesture like pinch zoom the analysis with just one hand will be difficult, but if the user were to use both his hands, this can be implemented by analyzing the separation between the user's hands. The biggest advantage of this approach is that using this method will not affect the user's natural movement. Also, this could be extended up to 7 users (maximum number of people a single Kinect can detect at the same time) and multiple touch points at the same time may be detected.

VII. Conclusions

A system named Image-Projective Desktop Varnamala Trainer (IDVT) was developed for use in educational purposes. It consists of a projector and RGBD sensor: Kinect V2. Regression-based touch detection was also implemented such that the device can be used as an interactive device which will accept user directions and provide suitable audio/video feedback to the user. A game-based application was developed for enabling children to learn alphabets and basic mathematics.

The regression-based method used is not only robust and accurate, but also very simple and practical. The method is designed to get over the constraint of low resolution data and not using highly powerful cameras. This can bring down the costs of such devices. Replacing the 2D laser sensor in IDAT with a Kinect was thus made possible without compromising on the accuracy.

References

[1] Education Overview [Online]. Available: http://www.worldbank.org/en/topic/education/overview#1

[2] Banerji, R., Bhattacharjea, S., and Wadhwa, W., 2013. The annual status of education report (ASER), *Research in Comparative and International Education, vol.8, 3.* doi: 10.2304/rcie.2013.8.3.387

[3] Survey for Assessment of Dropout Rates at Elementary Level in 21 States [Online]. Available: http://ssa.nic.in/research-studies-document_old/surveyreport-on-out-of-school-children/list-ofstudies/Dropout%20Study%2021%20States.pdf

[4] SUR40 Specialized Display [Online]. Available: http://www.samsung.com/uk/business/businessproducts/large-format-display/specialiseddisplay/LH40SFWTGC/EN

[5] Microsoft PixelSense [Online]. Available: http://www.microsoft.com/en-us/pixelsense/default.aspx

[6] Microsoft Surface Table [Online]. Available: http://money.cnn.com/2012/06/19/technology/microsoftsurface-table-pixelsense

[7] Matsumaru, T., and Akai, K., 2009. Step-On Interface on Mobile Robot to Operate by Stepping on Projected Button, *The Open Automation and Control Systems Journal, vol.2, pp.85-95*. doi: 10.217411874444300902010085

[8] Matsumaru, T., 2009. A Characteristics Measurement of Two-dimensional Range Scanner and its Application, *The Open Automation and Control Systems Journal, vol.2, pp.21- 30*. doi: 10.217411874444300902010021

[9] Liu, Y., Jiang, Y., and Matsumaru, T., 2012. Development of Image-projective Desktop Arm Trainer, IDAT, *IEEE/SICE International Symposium on System Integration (SII), pp.355-360*. doi: 10.1109/SII.2012.6426964

[10] Matsumaru, T., Jian, Y., and Liu, Y., 2013. Image-projective Desktop Arm Trainer IDAT for Therapy, *22nd IEEE International Symposium on Robot and Human Interactive Communication (IEEE RO-MAN 2013), pp.501-506.* doi: 10.1109/ROMAN.2013.6628411

[11] Matsumaru, T., Liu, Y., Jiang, Y., and Dai, C., 2014. ImageProjecting Desktop Arm Trainer for Hand-Eye Coordination Training, *Journal of Robotics and Mechatronic, vol.26, no.6*, pp.704-717.

[12] Wellner, P., Interacting with paper on the digitaldesk, 1993. *Commun. ACM, vol.36 (7),* pp.87-96. doi:10.1145/159544.159630

[13] SMART Technologies Inc. invents new technology for touch – sensitive display [Online]. Available: http://www.bobsguide.com/guide/news/2003/May/23/smarttechnologies-inc-invents-new-technology-for-touchsensitive-displays.html

[14] Touchless Touch http://www.touchlesstouch.com/

[15] Ankur Agarwal, Shahram Izadi, Manmohan Chandraker, Andrew Blake, High precision multi-touch sensing on surfaces using overhead cameras, *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer System.*

[16] Jingwen Dai; Chung, R., Making any planar surface into a touch-sensitive display by a mere projector and camera, *in Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on , vol., no., pp.35-42, 16-21 June 2012*

[17] Jun Hu, Guolin Li, Xiang Xie, Zhong Lv, and Zhihua Wang, Bare-fingers touch detection by the button's distortion in a projector–camera system, *IEEE Transactions on Circuits and Systems for Video Technology, Vol. 24, No. 4, April 2014*

[18] Microsof Kinect, Product features https://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx

[19] Microsoft Kinect for Windows standard libraries [Online]. Available: https://msdn.microsoft.com/en-us/library/hh973074.aspx

[20] Microsoft Kinect Depth sensing https://msdn.microsoft.com/en-us/library/hh855353.aspx

[21] Microsoft.NET [Online] Available: http://www.microsoft.com/net

[22] Microsoft Kinect for Windows [Online]. Available: http://www.microsoft.com/enus/kinectforwindows/meetkinect/default.aspx

[23] List of languages by total number of speakers [Online]. Available: http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers