

# Touch Screen Calibration Based on Laser Sensors

Seung Byum Seo<sup>1</sup> and Sang-Young Cho<sup>1,a</sup>

<sup>1</sup> *Division of Computer & Electronic Systems Engineering, Hankuk University of Foreign Studies, Republic of Korea*

**Abstract.** Interactive learning is becoming important in teaching and learning. In particular, the electronic board has educational significance in terms of being able to share and discuss learner's contents with instructors. However, the existing electronic board is not easily used in an educational environment due to the high cost and complicated installation process. This paper propose a mathematical model to calibrate the position of user coordinates by installing two laser sensors and a reflector on a wall. We virtually locate 9 points with software. To validate the algorithm, Raspberry Pi2 is used as a virtual sensor and simulation was performed by PC using RS-232 communication. This algorithm is powerful because up to four pre-saved coordinates are used to provide the higher accuracy.

## 1 Introduction

Electronic board, as known as SMART Board, is one of the devices for inputting information to a computer. The electronic board recognizes the coordinates or the path of the point pointed by the user and transmits the coordinates to the computer. The importance of electronic blackboard is increasing as a tool to help communication between teachers and learners [1].

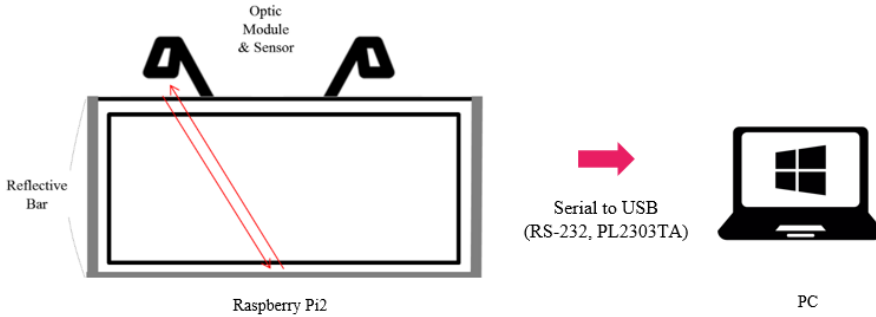
However, the existing electronic boards have disadvantages due to high cost and complicated installation environment [2]. These electronic boards include electromagnetic induction system, pressure sensitive system, and ultrasonic system. In this paper, we propose a calibration method that uses laser scanned method. Unlike previous systems, our system is free from unintended obstacles including acoustic noise and electromagnetic signals. Since our system can be installed anywhere on a wall, it is also affordable and precise [3].

This paper is organized as follows. Section 2 discusses the overview of our proposed system. Section 3 introduces the proposed algorithm. Finally, section 4 discusses the results of this paper and the future works needed.

## 2 System Overview

---

<sup>a</sup> Corresponding author: [sycho@hufs.ac.kr](mailto:sycho@hufs.ac.kr)



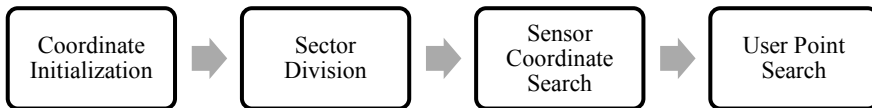
**Figure 1.** System Overview

Various touchscreen methods are used to implement electronic blackboards [4]. The proposed system consists of optic module, laser sensor, and reflective bar. For simulation, Raspberry Pi2 takes the role of laser sensor and electronic board. PC receives data from Raspberry Pi2 and uses it to process the location entered by the user (Figure 1).

Python was used as the development language for Raspberry Pi2 to simulate the sensor. On the PC, GUI is configured using C # and WPF [5]. RS-232 serial communication was used for communication between PC and Raspberry Pi2.

In this paper, we assumed that the laser sensor always rotates counterclockwise. In addition, the sensor on the left side from user perspective is called right sensor because it is installed on the left side from the sensor standard. Finally, the screen output projected by the projector is called *display*, while the wall where the reflection plate is installed is called *screen*.

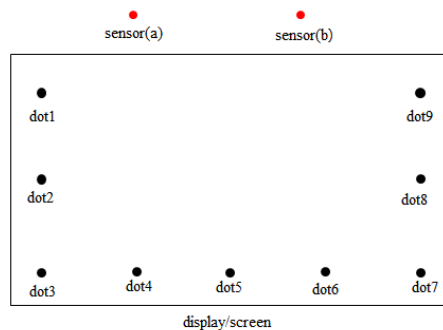
### 3 Mathematical Model



**Figure 2.** Four stages of algorithm

Figure 2 shows the operation steps. The ultimate goal of the algorithm is to find the coordinates of the input values from the user. In stage1, the algorithm initializes coordinates by arranging nine points in software. In stage2, sector division step, a total of 7 sectors are created based on nine points. In stage3, coordinate search, trigonometric function is used to find the position of the sensor coordinates in each sector. In stage4, two angles (left and right) outputted from the sensor and the sensor coordinates obtained in stage3 are used to find the position of the value inputted by the user

#### 3.1 Coordinate Initialization



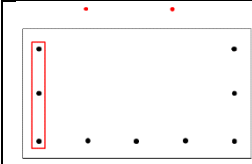
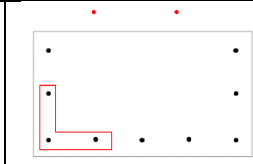
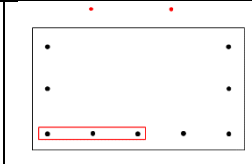
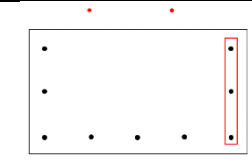
**Figure 3.** Setting Default Coordinates

Nine points were displayed on the display to determine the position of the sensor at various locations in Figure 3. The position of the coordinates can be set arbitrarily, but it is suggested to be close to the border of the screen. This is because, at coordinates close to the rim, the need for correction is higher because the sensor is most likely to detect errors.

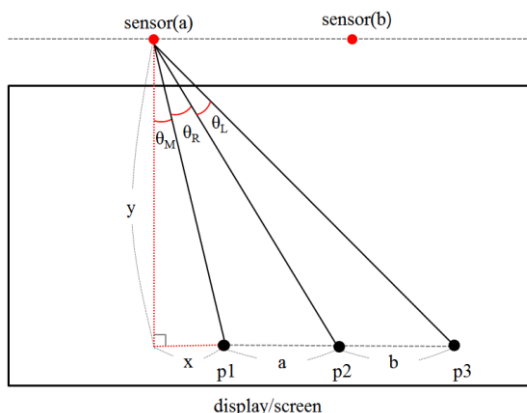
### 3.2 Sector Division

Based on the above nine coordinates, we divide the area into seven sectors. Then, the coordinates of the *sensor(a)* and the *sensor(b)* are stored in a separate space by using three consecutive coordinates for each sector. For example, in the area corresponding to *dot1*, *dot2*, and *dot3*, the coordinates of the *sensor(a)* and the coordinates of the *sensor(b)* can be known by using coordinate search algorithm given in stage 3. The coordinates of sensors are saved separately in a list. Table 1 shows the examples of how sectors are divided from sector one to sector seven.

**Table 1.** Examples of seven sectors and their type

Sector One (Type C)	Sector Two (Type B)	Sector Three (Type A)	Sector Seven (Type C)
			

### 3.3 Sensor Coordinate Search



**Figure 4.** Sensor coordinate search algorithm (Type A)

In each sector, the coordinates of *sensor(a)* and *sensor(b)* in Figure 4 can be obtained through the following algorithms. The seven sectors can be divided into three types. While *sector three, four, five* are arranged horizontally (Type A), *sector one and seven* is arranged vertically (Type C). *Sector two and sector six* are arranged in a perpendicular direction (Type B).

The method of obtaining the sensor coordinates in type A is as follows. In type A, there are three arbitrary coordinates on a straight line. In this case, the values of  $\theta_R$  and  $\theta_L$  can be measured by the angle formed by *sensor(a)* and *p1*, *p2*, *p3*. Also, the values of *a* and *b* can be measured as the number of pixels between *p1* and *p2*, and the number of pixels between *p2* and *p3*, respectively. Therefore,

Type A is an algorithm that obtains  $x$  and  $y$  when  $\theta_R, \theta_L, a$ , and  $b$  are given. To solve this problem, the following formula can be obtained.

$$\tan \theta_M = x/y \tag{1}$$

$$\tan(\theta_R + \theta_M) = (a + x)/y \tag{2}$$

$$\tan(\theta_L - \theta_M) = (b - x)/y \tag{3}$$

Based on tangent addition theorem, equation (2) and (3) can be simplified as below

$$\frac{\tan \theta_R + \tan \theta_M}{1 - \tan \theta_R \tan \theta_M} = \frac{a}{y} + \frac{x}{y} \tag{4}$$

$$\frac{\tan \theta_L - \tan \theta_M}{1 + \tan \theta_L \tan \theta_M} = \frac{b}{y} - \frac{x}{y} \tag{5}$$

The value of  $\tan \theta_M$  can be obtained by replacing  $x/y$  with  $\tan \theta_M$  and simplify the equation in terms of  $y$ .

$$\tan \theta_M = \frac{a \tan \theta_L - b \tan \theta_R}{(a+b) \tan \theta_R \tan \theta_L} \tag{6}$$

By plugging in the value of  $\tan \theta_M$  into equation (2) and (3) respectively, the value of  $\theta_M, x, y$  are obtained.

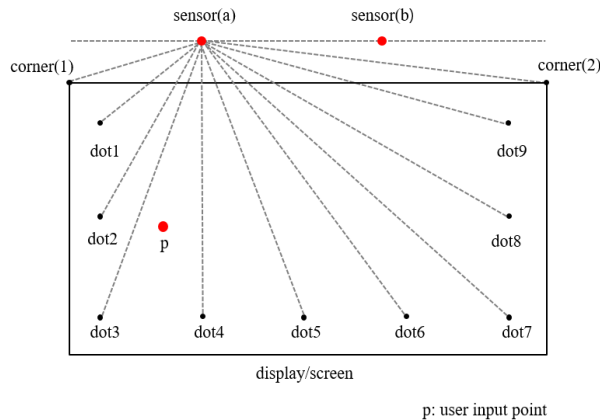
$$\theta_M = \tan^{-1} \left( \frac{a \tan \theta_L - b \tan \theta_R}{(a+b) \tan \theta_R \tan \theta_L} \right) \tag{7}$$

$$y = \frac{a}{\tan(\theta_R + \theta_M) - \tan \theta_M} \tag{8}$$

$$x = y \tan \theta_M \tag{9}$$

All of the coordinates of  $sensor(a)$  and  $sensor(b)$  in type A, B, and C can be obtained in a similar way.

### 3.4 User Point Search



**Figure 5.** Default angle and region setting

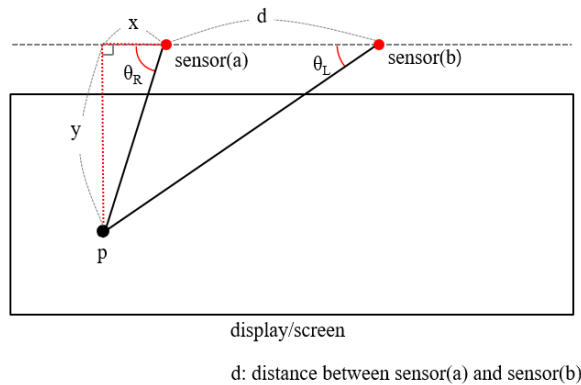
After storing the sensor position coordinates by sectors, the coordinates input by the user can be obtained. To obtain the coordinates input by the user, the following angle should be obtained as shown in Figure 5.  $D(n)$  where  $n$  is an integer from 0 to 9 denotes the default sensor angle.

$$\begin{aligned}
 D(0) &= \angle \text{corner}(1), \text{sensor}, \text{dot1} \\
 D(1) &= \angle \text{dot1}, \text{sensor}, \text{dot2} \\
 D(2) &= \angle \text{dot2}, \text{sensor}, \text{dot3} \\
 &\dots \\
 D(9) &= \angle \text{dot9}, \text{sensor}, \text{corner}(2)
 \end{aligned}$$

The values from  $D(n)$  are calculated as above are stored and managed separately. Then sector angle determines which sector does the user point belongs to. Region scope,  $R(n)$ , can be obtained by subtracting the angle formed by each angle from the angle at which the sensor outputs.

$$R(n) = \theta_{input} - \sum_{k=0}^{n-1} D(k) \quad (1 \leq n \leq 10, n \text{ is integer}) \quad (10)$$

Since the region scope is achieved from  $R(1)$  to  $R(10)$ , it is possible for system to locate which regions the user point  $p$  belongs to. For example, suppose that the user point  $p$  is located on the touch screen in Figure 5. Since  $p$  belongs to the region consists of  $\text{dot2}$ ,  $\text{dot3}$ ,  $\text{dot4}$  and  $\text{dot5}$ , it can be inferred that  $p$  belongs to  $R(4)$  and  $R(5)$ .



**Figure 6.** User coordinate search

User point  $p$  of Figure 6 can be obtained by solving the equations below.

$$\tan \theta_R = y/x \quad (11)$$

$$\tan \theta_L = y/(d + x) \quad (12)$$

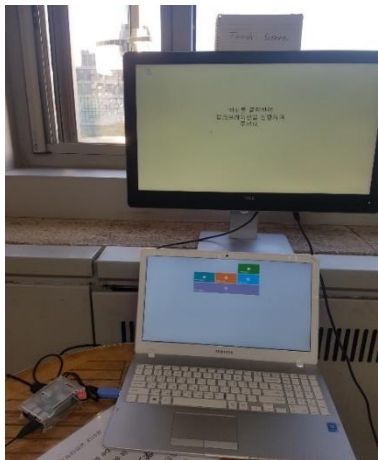
Therefore, the user point  $p$  can be obtained as equation (13) and (14).

$$x = (d \tan \theta_L) / (\tan \theta_R - \tan \theta_L) \quad (13)$$

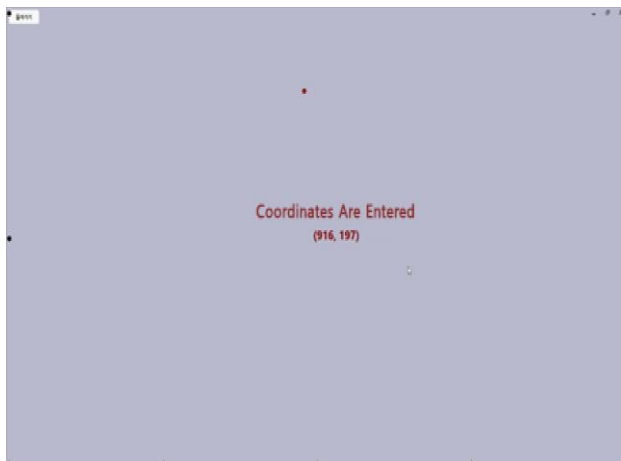
$$y = x \tan \theta_R \quad (14)$$

This algorithm is powerful because the data up to four coordinates could be used. In the worst situation, the value of  $p$  is calibrated by averaging  $sensor(a)$  and  $sensor(b)$  from one region. In an ideal circumstance,  $p$  is calibrated by averaging  $sensor(a)$  and  $sensor(b)$  from two regions which leads to a higher accuracy.

## 4 Conclusion



**Figure 7.** Simulation Environment



**Figure 8.** Simulation result

In this paper, two laser sensors, reflection bar, and nine virtual points were used to calibrate a virtual touch screen. As a result, we developed a total of four stages algorithm and built a test-bed to validate it. For simulation, Raspberry Pi2 was assumed as a laser sensor and was connected to a PC through RS-232 communication method (Figure7 and Figure 8). As a result, we have confirmed that the proposed algorithm is precise and efficient for touch screen calibration based on using a beam projector. For future work, we plan to test the system in a real world setting. At present, the simulation assumed that angles recognized from sensors are precise. However, the external factors from the real world is expected to be more complex. Therefore, future work will include examining unknown dynamics of calibration so that the higher accuracy is guaranteed.

This research was supported by a grant(16CTAP-C117026-01) from Technology Advancement Research Program funded by Ministry of Land, Infrastructure and Transport of Korean government

## References

1. M. Long, Input in second language acquisition, ERIC, p 473, (1985)
2. Sungchul Chun, In-soo Koo, "Beam projector calibration System based on Zigbee," *The Journal of the Institute of Internet, Broadcasting and Communication*, vol. 11, pp. 13-19, (2011)
3. F. Richards-Gustafson, "Smartboard vs. Projector," [Online]. Available: <http://smallbusiness.chron.com/smartboard-vs-projector-14141.html> (2017)
4. Touchscreen, [Online], Available: <https://en.wikipedia.org/wiki/Touchscreen> (2017)
5. T. Robinson, [Online]. Available: <http://libusbdotnet.sourceforge.net/V2/Index.html> (2010)