Machine Learning Approaches to Active Stylus for Capacitive Touch Screen Panel Applications

Hyoungsik Nam, Ki-Hyuk Seol, Seungjun Park
Dept. of Information Display, Kyung Hee University, Seoul Korea

Abstract

This paper introduces machine learning approaches on adding the stylus-touch to the capacitive touch screen technology. The proposed schemes can discriminate the stylus-touch from finger-touch as well as no-touch by means of classification algorithms using support vector machine and anomaly detection. The high frequency pulses are sent from a stylus to a touch screen and the receiver classifies the received sample sequences into three classes of no-touch, finger-touch, and stylus-touch. In addition, some possible applications of data transmission and user authentication are demonstrated.

Author Keywords

Machine learning; classification; support vector machine; anomaly detection; active stylus; touch screen

1. Introduction

The convenience and intuitiveness have pushed most mobile devices to adopt the mutual capacitive touch screen technology to be adopted. Consequently, it can be easily seen anywhere that users deal with devices by touching a screen with their fingers these days.

A conventional capacitive touch screen panel (TSP) consists of the array of mutual electrostatic capacitors formed at the intersection regions of transmitting (Tx) electrodes and receiving (Rx) electrodes [1]. Tx drivers sequentially transmit pulses to Tx lines and those pulses arrive at the charge amplifiers on Rx lines through mutual capacitors. Because finger-touches change mutual capacitances, charge amplifiers provide different output voltages for two cases of finger-touch and no-touch. This voltage difference allows finger-touches to be recognized. The host processor computes the touch position with the output voltages converted into digital values by an analog to digital converter (ADC). In general, the improved signal-to-noise ratio (SNR) for the higher detection accuracy is achieved by sending multiple Tx pulses over one touch point and by using the voltage difference between adjacent Rx lines [2].

In addition to finger-touch sensing, various types of other input tools for touch screens have supported more elaborate works such as writing texts and drawing pictures. A representative tool is a stylus that has been widely used in laptops, tablet PCs, and smartphones. A simplest passive stylus mimic finger based on conductor fibers [3], therefore, it is impossible to distinguish a stylus-touch from a finger-touch. Active stylus technologies, however, are capable of more sophisticated works by sending pulses to the TSP at the same timing as Tx pulses, transmitting different frequency pulses [4], or adopting the electromagnetic resonance technology [5]. Although these methods allow the TSP to distinguish the stylus-touch from the finger-touch along with sophisticated works, increased power consumption, operation complexity, hardware complexity, and cost is not avoidable.

This paper introduces another active stylus approach that considers the capacitive touch-screen technology as one of machine learning problems, especially, a classification problem. Therefore, two touch-screen algorithms based a support vector machine (SVM) [6,7] and an anomaly detection (AD) [8] are described. These machine learning methods enable finger and stylus discrimination for sophisticated works without adding layers, increasing operation modes, and hardware complexity. Finger and stylus are detected simultaneously and discriminated on a single platform. This finger and stylus discrimination capability is required for the palm-rejection.

2. Proposed Stylus-Touch Algorithms

SVM-based Algorithm: The SVM-based algorithm sees the capacitive TSP as the classification problem with three classes. They are three different sequences of K samples such as no-touch, finger-touch, and stylus-touch from the ADC of a Rx side as shown in Fig. 1. The stylus-touch sequence is generated by higher frequency pulses of a stylus than Tx pulses. Then, a 3-class SVM discriminates these sequences by comparing the distance with support vectors obtained through the training procedure. Consequently, even both stylus and finger placed on the screen are detected separately.

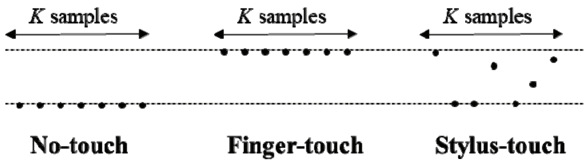


Figure 1. Three sequences of no-touch, finger-touch, and stylus-touch for a proposed SVM-based algorithm

The Rx circuit model of the SVM-based capacitive TSP can be described with a mutual capacitor (C_M), a stylus capacitor (C_{ST}), charge amplifier, ADC, and SVM classifier as presented in Fig. 2. A finger-touch reduces C_M, while C_{ST} is added only by a stylus-touch.

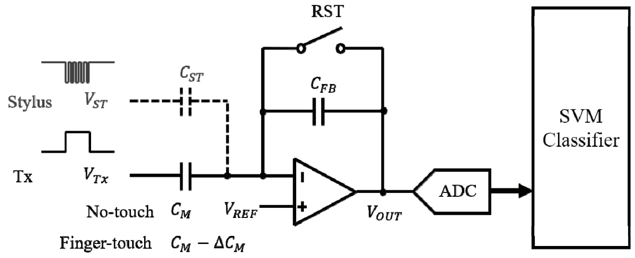


Figure 2. Rx circuit model of a proposed SVM-based algorithm

A timing diagram of ADC sampling operations is described in Fig. 3, where $V_{OUT\text{-}NT}$, $V_{OUT\text{-}FT}$, and $V_{OUT\text{-}ST}$ are outputs of the

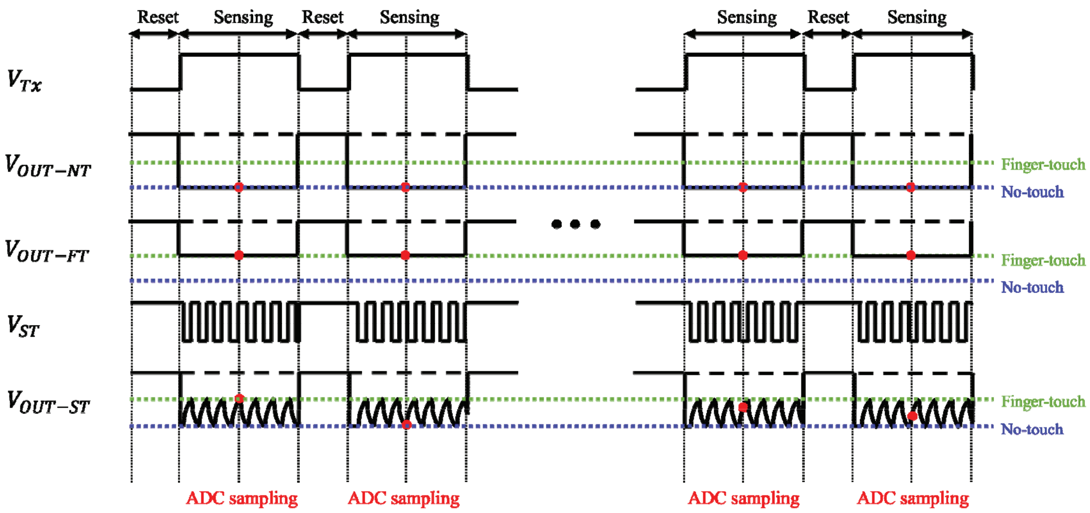


Figure 3. ADC sampling operations for three touch sequences. Red dots represent sampling points.

charge amplifier for no-touch, finger-touch, and stylus-touch, respectively. Especially, because red dots represent the sampling points, the stylus-touch samples can be placed between no-touch and finger-touch levels due to high frequency stylus pulses.

AD-based Algorithm: A proposed AD algorithm is implemented based on a simple autoencoder (AE) architecture as presented in Fig. 4. One latent variable that is generated by an encoder from the input sequence (x_1, …, x_K) represents one case among no-touch and finger-touch. The decoder reconstructs the no-touch or finger-touch sequence (y_1, …, y_K) according to the latent variable. The reconstructed sequence is generated similarly to the input sequence. This AE-based network is trained only with the dataset of finger-touches and no-touches to minimize the difference between the output sequence and the input sequence. Therefore, whereas finger-touches and no-touches cause very small errors between input and reconstructed sequences, the stylus-touches show the large errors, enabling the stylus-touch discrimination from finger-touches and no-touches as plotted in Fig. 5, where 100 sequences for each cases are evaluated.

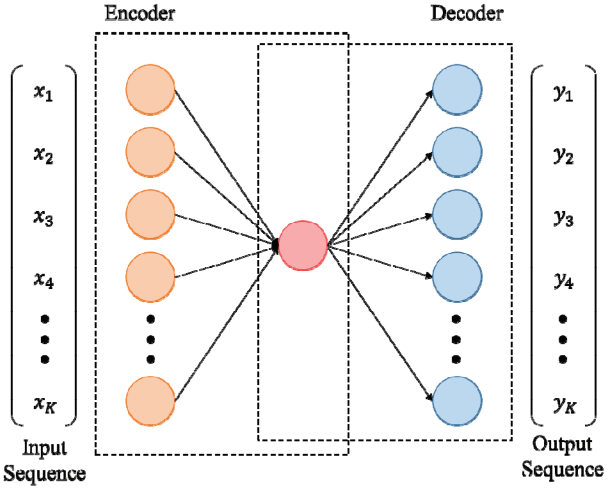


Figure 4. Autoencoder (AE) structure for a proposed AD scheme

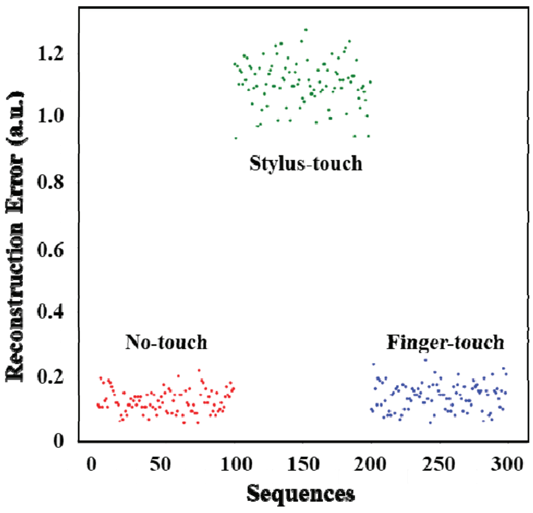
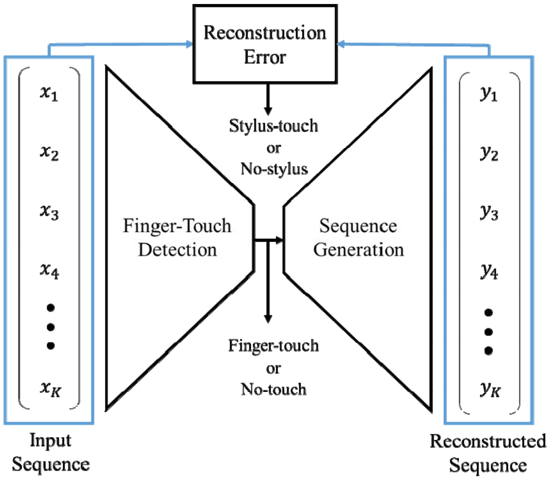


Figure 5. Reconstruction error distributions for no-touch, finger-touch, and stylus-touch sequences

The resultant AD scheme consists of the finger-touch detection that is the encoder of the trained AE, the sequence generation that corresponds to the decoder, and the stylus-touch detection based on the error between input and reconstructed sequences as depicted in Fig. 6. When the error is higher than the threshold value, the input sequence is classified into the stylus-touch, and otherwise, the finger-touch or the no-touch is determined according to the output of the finger-touch detection. Because the finger-touch detection provides the hard-decision output of two digital values of 0 and 1, the hard-decision case must train them separately, but the sequence generation is replaced with the simple multiplexer of two input sequences showing the low hardware complexity.



Finger-Touch Detection	Reconstruction Error	Decision
Finger-touch	Stylus-touch	Stylus-touch
No-touch	Stylus-touch	Stylus-touch
Finger-touch	No-stylus	Finger-touch
No-touch	No-stylus	No-touch

Figure 6. Overall block diagram of a proposed AD-based stylus-touch detection algorithm

3. Evaluation Results

The evaluation of proposed algorithms has been conducted with 8-inch capacitive TSP, programmable system-on-chip board (PSoC) [9], and host processor board (Raspberry Pi 3) [10] as presented in Fig. 7. Tx lines are driven by pulses of 5 V and 32 kHz. The 8-bit ADC in a PSoC board samples the output of a charge amplifier with a feedback capacitor (C_F) of 3 pF for a mutual capacitor (C_M) of 1.8 pF. The number of samples in a sequence (K) is set to 16 leading to the reporting rate of 100 Hz for 20 Tx and 26 Rx lines. The stylus tip has a diameter of 5 mm and the tip voltage is given at 3 V and 315 kHz.

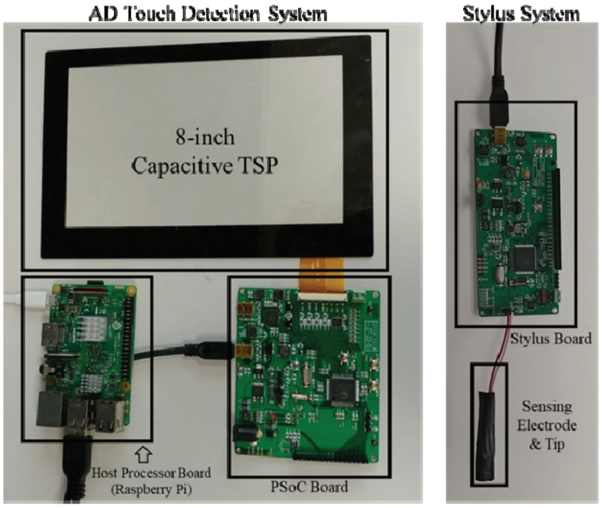


Figure 7. Evaluation setup

The output waveforms of the charge amplifier are measured for three touch cases as shown in Fig. 8. While no-touch and finger-touch cause 3 V and 2.6 V voltage swings respectively, the stylus-touch generates 0.4 V fluctuation between no-touch and finger-touch voltage levels as mentioned in the previous section.

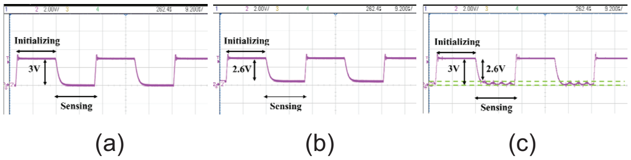


Figure 8. Measured output waveforms of the charge amplifier (a) no-touch (b) finger-touch (c) stylus-touch

For the SVM-based algorithm, 10^6 sequences of each case, that is, totally 3×10^6 sequences are used for extraction of 25 support vectors. However, the AD-based algorithm uses only 2×10^6 sequences of no-touch and finger-touch without stylus-touch sequences to train the AE network. Both SVM and AD approaches achieve lower bit error rate (BER) than 10^{-6}.

To compare the performance of SVM and AD schemes further, signal-to-noise ratios (SNRs) are also estimated as described in Table 1. The SNR of the SVM method is measured with 1-dimensional components extracted by a principle component analysis (PCA) [11] and the SNR of the AD algorithm is calculated from reconstruction errors. Both SVM and AD schemes show the similar SNR performance of around 20 dB. On the other hand, the AD method shows much lower hardware complexity than the SVM in the viewpoint of the number of multiplexers as summarized in Table 2. While the SVM requires 400 multipliers of 25 support vectors and 16 touch samples, the AD needs only 16 multipliers for a finger-touch detection that is an encoder part of the AE network.

Table 1. Measured SNR performance of SVM and AD algorithms

	SVM	AD
SNR (dB)	20.22	20.49

Table 2. Estimated number of multipliers for SVM and AD schemes

	SVM	AD
Number of multipliers	400 (25×16)	16

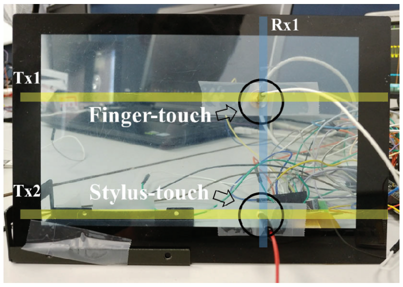
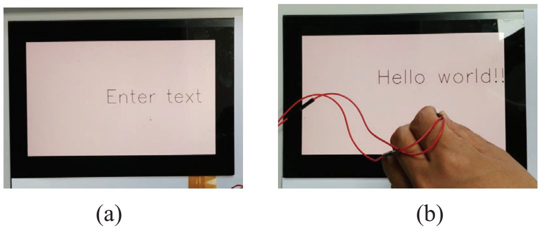


Figure 9. Evaluation of finger and stylus discrimination

The simultaneous finger-touch and stylus-touch discrimination is verified as shown in Fig. 9 where both finger and stylus are put on the screen. In this setup, a finger-touch is replaced with the grounded metal tack. The same lower BERs than 10-6 are established for both finger-touch and stylus-touch. Therefore, it is ensured that the proposed scheme can support palm-rejection by detecting stylus-touches separately from palm-touch regions.

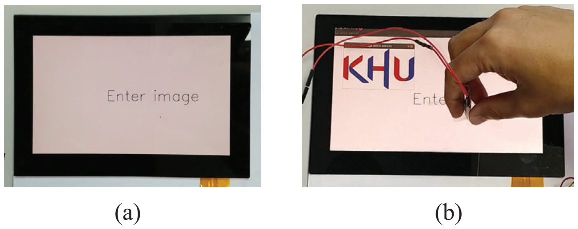
4. Applications

The proposed stylus scheme can be employed in various applications of data transmission and user identification. When the display presents the command of 'Enter text' on a screen like Fig. 10(a), the data of "Hello world!!" are sent from the stylus and then, as shown in Fig. 10(b), that data appear on the display after the whole data transmission is completed. Similarly, it can be also applied to image transmission. First, the command of 'Enter image' is displayed in Fig. 11(a). Then, the stylus sends the image data that appears on the top-left region of the screen in Fig. 11(b). In addition to text and image data transmission, it is possible to identify multiple stylus pens with the proposed technology. For example, two stylus modules transmit different identification codes from each other that allow the display to find out which stylus is placed on a screen. In Fig. 12(a), a red circle is displayed when the red-line stylus is used. Whereas, the green circle is shown in Fig. 12(b) when the green-line stylus is touched.



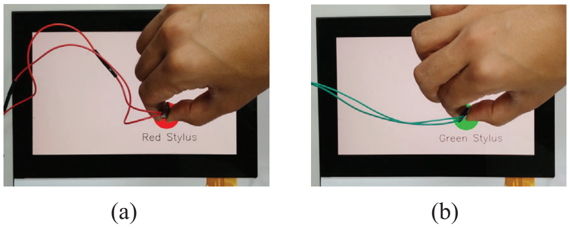
(a) (b)

Figure 10. Text data transmission (a) Preparation stage (b) Received text



(a) (b)

Figure 11. Image data transmission (a) Preparation stage (b) Received image



(a) (b)

Figure 12. User identification (a) Red stylus contact (b) Green stylus contact

5. Conclusion

We demonstrate machine learning approaches that see the touch screen technology as one of classification problems. Consequently, one more class of a stylus-touch is added to two existing classes of no-touch and finger-touch by means of high frequency pulses on the tip voltage at a stylus. These three classes are successfully discriminated by SVM and AD algorithms. We are very sure that these researches will pave the way to many machine learning applications in a display area down the road.

6. Acknowledgements

This research was supported by IDEC (EDA Tool) and the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2016R1A2B4009787).

7. References

1. T. Hwang, W. Cui, I. Yang and O. Kwon, “A highly area-efficient controller for capacitive touch screen panel systems,” IEEE Trans. Consum. Electron. **56**(2), pp. 1115-1122 (2010).
2. I. Yang and O. Kwon, “A touch controller using differential sensing method for on-cell capacitive touch screen panel systems,” IEEE Trans. Consum. Electron. **57**(3), pp. 1027-1032 (2011).
3. M. Badaye and R. R. Schediwy, “Passive stylus for capacitive sensors,” U.S. Patent 8,125,469 (2012).
4. J.-S. An, S.-H. Han, J. E. Kim, D.-H. Yoon, Y.-H. Kim, H.-H. Hong, J.-H. Ye, S.-J. Jung, S.-H. Lee, and J.-Y. Jeong, “9.6 a 3.9 khz-frame-rate capacitive touch system with pressure/tilt angle expressions of active stylus using multiple-frequency driving method for 65' 104×64 touch screen panel,” in IEEE ISSCC Dig. Tech. Papers, Feb. 2017, pp. 168-169.
5. C. Chang-Hsien and C.-H. Huang, “Method for sensing fast motion, controller and electromagnetic sensing apparatus,” U.S. Patent Appl. 14 472 345, Nov. 19, 2015.
6. K.-H. Seol, S. Park, S.-J. Song, and H. Nam, “Finger and stylus discrimination scheme based on capacitive touch screen panel and support vector machine classifier,” Jpn. J. Appl. Phys. **58**(7), Art. no. 074501 (2019).
7. K.-H. Seol, S. Park, and H. Nam, “A Contact-based Data Communication Technique using Capacitive Touch Screen Panel and Support Vector Machine Classifier,” IEEE Access **7**, pp. 178596-178604 (2019).
8. K.-H. Seol, S. Park, and H. Nam, “Active stylus-touch discrimination scheme based on anomaly detection algorithm,” J. Soc. Inf. Disp., Submitted (2020).
9. Cypress Semiconductor Corporation PSoC 5LP. Accessed: Sep. 9, 2019. [Online]. Available: http://www.cypress.com/products/32-bit-armcortex-m3-psoc-5lp/
10. Raspberry Pi Foundation Raspberry Pi 3. Accessed: Sep. 9, 2019. [Online]. Available: https://www.raspberrypi.org/documentation/
11. C. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2007.