Mathematical Modeling and Algorithm Application
ISSN: 3006-0842 | Vol. 2, No. 3, 2024

Pressure simulation algorithm for non pressure sensitive touch screen

Tiancai Zhou, Zhenshen Zhao

College of Southwest Petroleum University, Chengdu, China

Abstract: The application of non pressure-sensitive touch screens in smart tablet software is becoming increasingly widespread, but they cannot directly sense pressure, which limits the user's writing experience. This article proposes a pressure simulation algorithm based on gestures and speed, which simulates pressure by analyzing touch data. This algorithm identifies the user's touch gestures and speed changes, infers the corresponding pressure values, and maps them to simulated pressure data. The experimental results show that the algorithm can significantly improve the writing performance of non pressure-sensitive touch screens, making the user experience more natural and smooth.

Keywords: Non pressure sensitive touch screen; pressure simulation; gesture recognition; smart board writing; algorithm design.

1. Introduction

With the popularization of smart education, the application of smart chalkboard software in teaching is becoming increasingly widespread. Smart blackboard writing software has improved the interactivity and efficiency of teaching through digital means, becoming an important tool in modern education. As an important input device for smart chalkboard, the performance of touch screen directly affects the user experience. The response speed, accuracy, and pressure sensing ability of touch screens are key factors determining the user's writing and drawing experience.

Non pressure-sensitive touch screens have been widely used due to their low cost and easy maintenance. Compared to pressure-sensitive touch screens, non pressure-sensitive touch screens have a simpler hardware structure and lower manufacturing costs, making them widely promoted in the field of education. However, non pressure-sensitive touch screens cannot directly sense pressure, which limits users' expressive power when writing and drawing. When users use non pressure-sensitive touch screens for writing, they cannot achieve changes in stroke thickness by applying different pressures, which to some extent affects the naturalness and fluency of writing.

In order to enhance the user experience of non pressure-sensitive touch screens, this paper proposes a pressure simulation algorithm based on gestures and speed. This algorithm simulates the corresponding pressure values by analyzing the user's touch data, including touch position, speed, and gesture changes. Specifically, the algorithm identifies the user's touch gestures (such as light touch, heavy press, quick swipe, etc.) and changes in touch speed, infers the pressure applied by the user, and maps it to simulated pressure data. Through this method, users can obtain a writing experience similar to pressure-sensitive touch screens when using non pressure-sensitive touch screens for writing and drawing.

The research objective of this article is to design and implement this pressure simulation algorithm, and verify its effectiveness through experiments. By optimizing algorithms and data processing flow, this article aims to improve the writing performance of non pressure-sensitive touch screens, making the user experience more natural and smooth. The research results of this article will provide new ideas and technical support for the development of smart blackboard writing software, and promote the development of smart education.

2. Organization of the Text

2.1. Related work

2.1.1. Traditional pressure simulation methods

Traditional pressure simulation methods mainly rely on hardware improvements, such as adding pressure sensors. These methods directly measure the pressure applied by the user by installing pressure sensors underneath the touch screen. However, this method has several significant drawbacks. Firstly, adding pressure sensors will significantly increase the manufacturing cost of touch screens. Secondly, the installation and calibration process of pressure sensors is complex, which increases the difficulty of production and maintenance. In addition, the sensitivity and durability of pressure sensors are also affected by the usage environment and time, leading to a decrease in measurement accuracy. Therefore, although traditional hardware improvement methods can directly sense stress, their high cost and complexity limit their widespread application.

2.1.2. Gesture based pressure simulation

In recent years, gesture based pressure simulation methods have gradually gained attention. These methods infer the corresponding pressure values by analyzing the user's touch gestures and speed. Specifically, the gesture based pressure simulation method utilizes touch data collected from a touch screen, including touch position, speed, and time information, and analyzes the trend of these data changes through algorithms to infer the pressure applied by the user. For example, when the user slowly moves their finger on the touch screen, the algorithm can infer that the user has applied significant pressure; And when the user quickly slides their finger, the algorithm speculates that the applied pressure is relatively small. The gesture based pressure simulation method does not require additional hardware support, has low cost, and is easy to implement, therefore it has broad prospects in the application of non pressure-sensitive touch screens.

2.2. System design

2.2.1. System Architecture

The system mainly includes a touch data acquisition module, a data processing module, a pressure simulation module, and a display module. The touch data acquisition module is responsible for obtaining user touch data from the touch screen; The data processing module preprocesses the collected data; The pressure simulation module performs pressure simulation based on preprocessed data; The display module applies simulated pressure values to the display effects of writing and painting.

2.2.2. Touch data collection

The touch data acquisition module collects user touch data through the touch screen, including touch position, speed, time and other information. Touch position data is used to determine the specific position of the user's finger on the screen; Speed data is used to analyze the speed of user finger movement; Time data is used to record the occurrence time of touch events. These data provide a foundation for subsequent pressure simulations.

2.2.3. Data Processing

The data processing module preprocesses the collected touch data, including denoising, smoothing, and normalization. Denoising processing is used to eliminate random noise in touch data and improve data accuracy; Smooth processing is used to eliminate sharp changes in touch trajectories, making them smoother; Normalization processing is used to convert touch data to a unified scale range for subsequent algorithm processing. The preprocessed data is more stable and reliable, providing a good data foundation for pressure simulation.

2.3. Pressure simulation algorithm

2.3.1. Gesture recognition

Gesture recognition is one of the key steps in pressure simulation algorithms. By analyzing the shape and changes of touch trajectories, identify user gesture types such as light touch, heavy press, quick swipe, etc. Gesture recognition algorithms can use machine learning methods to train models to recognize different types of gestures. For example, algorithms such as Support Vector Machine (SVM) or Convolutional Neural Network (CNN) can be used to classify touch data and recognize the user's specific gestures. The accuracy of gesture recognition directly affects the effectiveness of pressure simulation, therefore requiring a large amount of training data and precise model design.

2.3.2. Speed analysis

Speed analysis is another important step in pressure simulation algorithms. By calculating the speed changes between touch points, infer the pressure applied by the user. Generally speaking, the slower the touch speed, the greater the applied pressure; The faster the touch speed, the less pressure is applied. The speed analysis algorithm can calculate the distance and time difference between adjacent touch points to obtain the touch speed, and infer the pressure value based on the speed change. To improve the accuracy of speed analysis, different speed analysis methods can be used for different gesture types based on the results of gesture recognition.

2.3.3. Pressure Mapping

Pressure mapping is the process of mapping the results of gesture recognition and velocity analysis onto pressure values. The specific mapping relationship can be calibrated and optimized through experimental data. For example, actual pressure values under different gestures and speeds can be measured through experiments to establish a mapping relationship between gestures, speeds, and pressure values. The pressure mapping algorithm can use methods such as linear regression and neural networks to calculate the corresponding pressure values based on gesture and velocity data. In order to improve the accuracy of mapping, a multidimensional mapping model can be used to comprehensively consider gestures, speed, and other related parameters.

2.3.4. Real time processing

In order to achieve a smooth user experience, pressure simulation algorithms need to perform real-time processing while collecting touch data. By adopting multi-threaded technology and efficient data structures, the real-time performance and response speed of the algorithm can be improved. Real time processing algorithms need to complete data collection, preprocessing, gesture recognition, velocity analysis, and pressure mapping steps in a short period of time to ensure that users can feel pressure changes in real time while writing and drawing. In specific implementation, the following optimization strategies can be adopted:

1. Data block processing: Divide touch data into small blocks and process them block by block to reduce the amount of data processed each time and improve processing speed[J].

2. Parallel computing: Utilizing multi-threaded or multi-core processors to allocate different processing steps to different threads or processors for parallel execution, improving overall processing efficiency[K].

3. Cache mechanism: Cache commonly used calculation results to reduce duplicate calculations and improve processing speed.

4. Incremental update: Only update the changed portion of data during each processing, reducing unnecessary calculations and improving real-time performance.

Through the above optimization strategies, the stress simulation algorithm can achieve efficient real-time processing while ensuring accuracy, providing a smooth user experience[L].

2.4. Experimental result

2.4.1. Experimental setup

To verify the effectiveness and applicability of the pressure simulation algorithm proposed in this article, we conducted experiments on different types of non pressure-sensitive touch screens. The experimental setup includes multiple touch screen devices, different gesture types (such as light touch, heavy press, quick swipe, etc.), and speed ranges. Through these experimental data, we can comprehensively evaluate the performance of the algorithm in various usage scenarios.

2.4.2. Performance evaluation

The experimental results show that the pressure simulation algorithm proposed in this paper can significantly improve the writing experience of non pressure-sensitive touch screens.

The simulated pressure value has a high correlation with the actual pressure value, which can accurately reflect the pressure changes applied by the user. Compared with traditional hardware improvement methods, this algorithm not only has low cost and simple implementation, but also performs well in performance. Specifically, the algorithm exhibits high accuracy and stability across different devices and gesture types, meeting the needs of practical applications.

2.4.3. User Feedback

To further validate the actual effectiveness of the algorithm, we conducted user testing and collected feedback from users on the stress simulation results. Most users believe that the algorithm in this article can significantly improve the expressiveness of writing and drawing, and provide a more natural and smooth user experience. User feedback indicates that the pressure simulation method based on gestures and speed has good performance and user acceptance in practical applications. Users generally indicate that the simulated pressure changes make the writing process more realistic and intuitive, enhancing the overall user experience.

Literature References

[1] IEEE transactions on cybernetics. Volume PP , Issue . 2023

[2] Theoretical Computer Science. Volume 1016 , Issue . 2024. PP 114777-114777

[3] International Journal of Data Science and Analytics. Volume , Issue prepublish . 2024. PP 1-21