EE389 Electronic Design Lab II Project Report, EE Dept, IIT Bombay, November 2008

Digital Note-Taker

Group No: D13

Kshitij Auluck (05D07016) kshitij.a@iitb.ac.in

Ketan Panwar (05D07035) ketan_panwar@iitb.ac.in

K.Jyothi (05007031) jyothik@iitb.ac.in

Supervisors: Deepankar, D.K. Sharma

Abstract

This project proposes a novel real-time method of character recognition from handwriting. It employs a 3-D motion sensor (accelerometer) to characterize the motion of the nib of a pen on which it is attached, as different strokes in a character are written. To extend this capability to the entire Latin alphabet, including small-case letters and jointed writing, we must extract characteristics which are independent of the user. To do so, we require faster processing and larger memory buffer.

1. Introduction

In the previous stage of this work we demonstrated a simple ATMEGA-32 based system which could reliably identify upper-case Latin characters consisting of straight line strokes (such as A, E, F, H…, Z); holding the pen in a certain orientation. To detect more than one letter with either spacing or continuous writing we required faster processing and larger buffer to store huge data. So we implemented the algorithm now in MSP.

Here we report the progress made in extending this concept to ordinary writing.

2. Problem Statement

The ideal aim of this project is to build a Smart Pen which performs almost real-time character recognition in a novel and yet computationally simple way, using a single 3-axis accelerometer. From the end-user's point of view this must involve satisfactory solution to the following problems.

- Reliable, unambiguous identification of all Latin characters.
- Suitable for upper-case, lower-case, jointed handwriting of various sizes; spaces and punctuation scratching and underline; figures and tables.
- Independent of style of holding the pen (such as tilt)
- Writing speed 2 characters/sec; battery life of 4 hours of continuous use.
- Save and continue option; opening multiple documents at a time; starting new paragraph/points; display using Bluetooth.
- Robust, compact and unobtrusive mechanical design.

Within a limited time-frame only a subset of these requirements, could be implemented

3. Design Approach:

A top-down design approach (Fig. 1) was followed in the design of this system. At the highest abstraction level, we have a stroke-input character output block.; which takes in a stream of tokens corresponding to the constituent elementary strokes. For example an 'S' can be derived from a "C" stroke and an inverted 'C' stroke in succession. Thus a succession of strokes can synthesize a character. There are 2 problems in the design of this block: each character can be synthesized differently and a particular stroke-pattern may be a subset of 2 or more characters. Thus, this block is essentially a stroke-array parser which implements a maximal-match algorithm; which matches the stroke permutation with the largest stored pattern.

The next block in the design hierarchy is the stroke-token generator. This takes the processed acceleration array, and outputs the stroke-token which maps to the curvature characteristics of the acceleration signal. The tokens are put in a stream. There are different algorithms which we explored, to extract curvature information from the acceleration. These are detailed in section 4.

The lowest block in the design hierarchy is the acceleration processing block. This takes in a stream of sampled analog acceleration values in 3-axes and outputs a filtered, average corrected, and tilt/orientation corrected. These operations are detailed in Section 6.

4. Experiments: To determine stroke-characteristics from acceleration

These experiments were performed on MATLAB

A) **Angle evolution of secants of a curve.**

In this approach, the curve was split into segments and the angles made by each segment with the horizontal reference was evaluated and displayed. This showed a consistent behaviour in describing curved strokes such as “C” ‘S’ O etc. The angle was determined by taking a tan inverse of X and Y displacements, which were in turn calculated by double integration of corrected acceleration.

Example :

Character 0 : (expected to vary from 0 to 360 degrees)

Reading1 :38 57 114 183 234 291 302 313

Reading2 : 34 54 163 209 243 274 300 324

Character U : (expected to vary from 90 to 270)

Reading1: 92 108 169 227 264 288 300 308

Reading2: 94 103 147 181 215 255 291 309

Character S : (expected to first increase from 0 to 180 and then decrease to 0)

Reading1: 17 49 151 153 133 40 359 340

Reading2: 359 34 161 168 189 86 05 00

B) **Turning-point detection for horizontal and vertical motions**

The idea is that the sense of a curve is characterized by knowing the points where horizontal or vertical velocities are zero: i.e. the respective turning points. This involves much fewer computations, but requires the buffering of acceleration data for accurate velocity-zero computation. The graphs for acceleration and velocity-reversal are displayed (fig 2 and 3.) The disturbance Is the Y-acceleration signal, immediately following which the Y- velocity is displayed. A level of 150 means that velocity is in the UP direction while a level of 100 means that the direction Is DOWN. Thus the graph shows the turning in Y direction for an inverted U and a small case d, which is just as expected.

5. Hardware design:

The Hardware comprised of a 3-axis MEMS accelerometer mounted on a pen, analog filters and Microcontroller MSP430F2417. The character written with the pen will be recognized and displayed on a LCD display. The MEMS accelerometer (MMA7260) generates three voltage output signals, each in proportion to the net force in x, y and z directions

Specifications of Hardware-

1. Microcontroller: MSP430F2417 (TI)

Features:

- 8kB RAM
- 92kB Flash
- 16MHz operation
- 8-channel SAR ADC
- 64-pin LQFP package
- Supply Voltage
- Full load current: 2.8mA at 8MHz

2. Accelerometer: MMA7260QT (Freescale)

Features:

- Selectable Range (1.5g/2g/4g/6g)

  Sensitivity for 1.5g (which we are using)

  min: 749 mV/g

  max: 880 mv/g

- Low Current Consumption

  Full Load Current : 500 μA

  Sleep Mode Current: 3 μA

- Low Voltage Operation: 2.2 V – 3.6 V
- Integral Signal Conditioning with Low Pass Filter

Software Tasks

A) Filtering:

The accelerometer output contains noise and vibrations and must be filtered for ease of interpretation on MATLAB. A simple window averaging concept was used for filtering. The new byte of acceleration was put into a small circular buffer of size 10. The average of these 10 values was then placed into the main acceleration stream.

B) Rest-Thresholding:

The minimum and maximum in a window of the past 30 data was noted. If their difference was less than 1, then the state was determined to be of REST. Otherwise the state is of MOTION. It is important to isolate this threshold of rest and motion to do data correction .

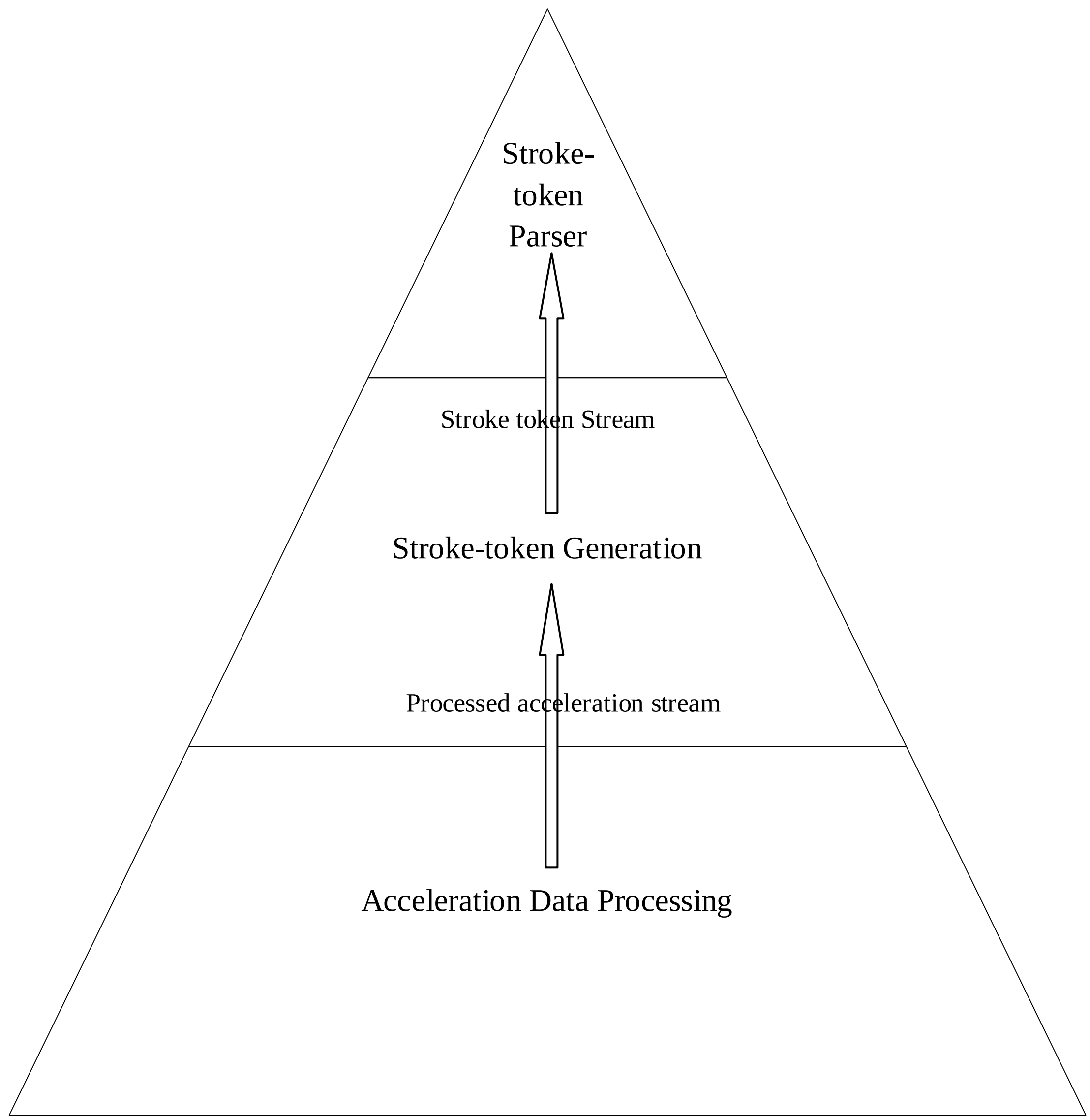
C) Data Correction

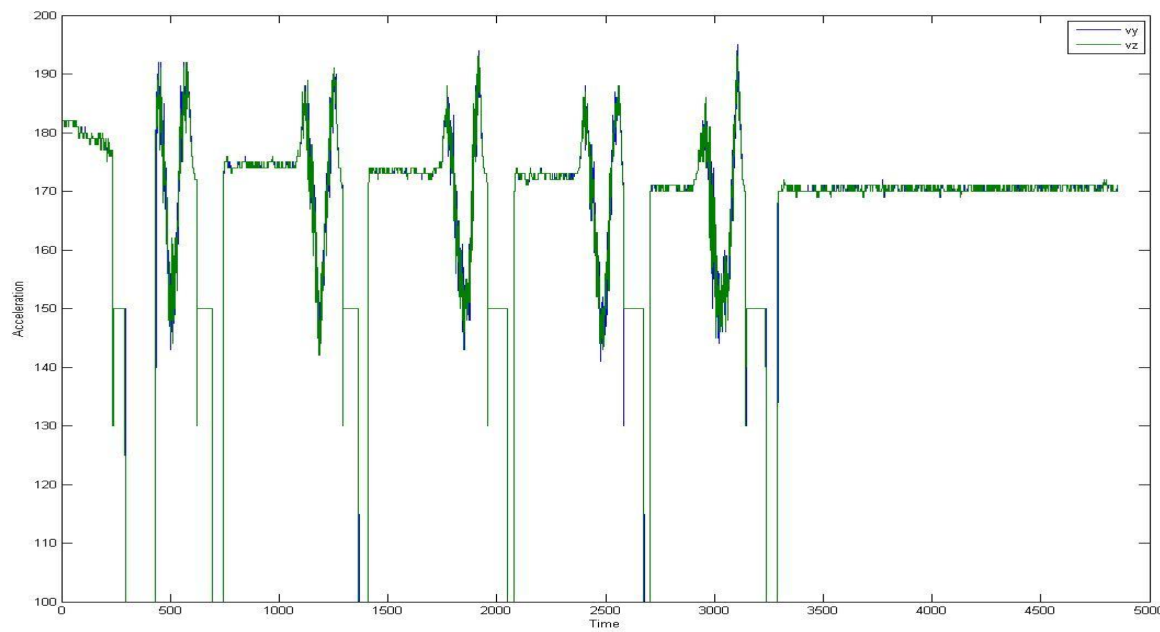
The average value of a stream of continuous writing must be subtracted from each acceleration value in that stream. This will ensure that the final and initial position always have a velocity of 0. This is important for gathering precise information about velocity zeroes.

Conclusions and Future Work:

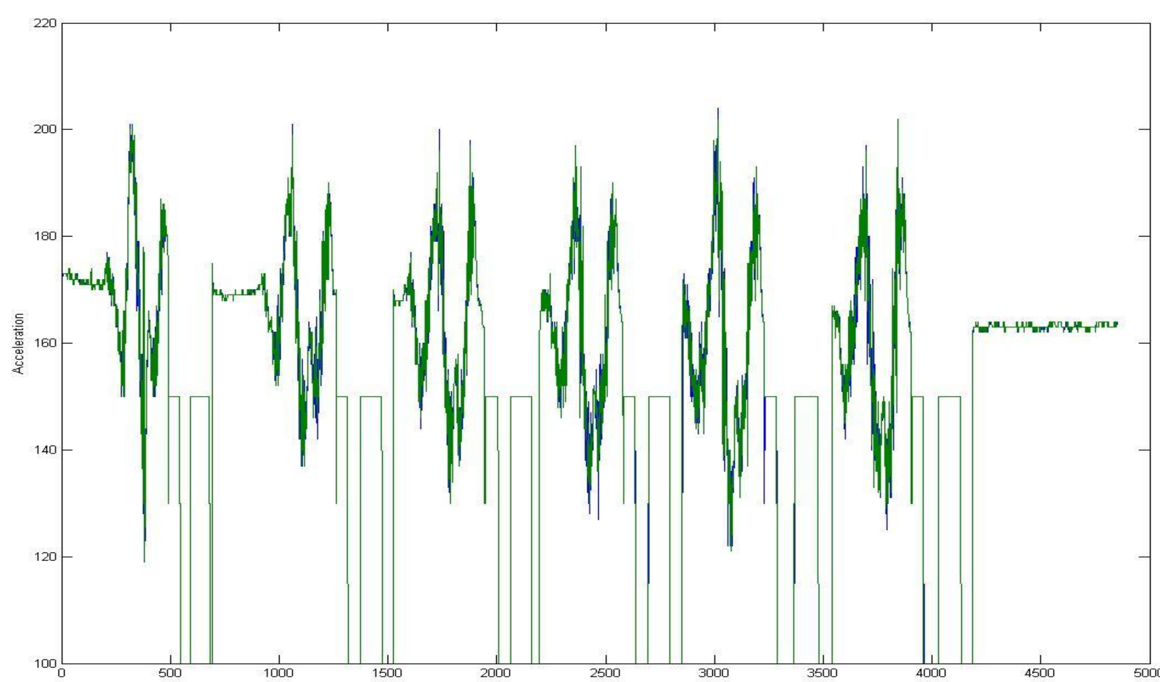
This is a new method which can be used for character recognition. The work which is presented here is only done for Latin alphabets. The algorithms developed can be develop further to incorporate diagrams also better algorithms can be used for tilt effects and other physical constraints such as holding pen etc.

Data flow chart and figures:





Inverted U
Y-axis: Acceleration and
velocities



Small alphabet d
Y-axis: Acceleration and velocities