Penn Project Report
Machine Learning in Real Time Character Recognition

Abstract

This paper presents a MEMS based digital pen for handwriting recognition. While there are several devices which are capable of digitizing handwritten notes, all commercially available devices require secondary hardware to function. We propose a single device solution for digitizing text as it is written on a page. Our device tracks it's motion and uses machine learning methods to identify written characters in real time. By solving for the dynamics of the system, and using them in our Kalman Filter, we are able to estimate the pen tip accelerations and rotations as a function of time. We will use this data to compare several machine learning algorithms based on their ability to recgnize handwritten characters.

1. Prototype

In this section we give a brief summary of the hardware incorporated into the proposed device. We first describe the mechanical device before then discussing the electronics.

1.1. Hardware

The pen layout and body was designed in CAD (pictured below) to ensure appropriate packing of components. The clear body was 3D printed in ABS in the University's prototyping labs.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

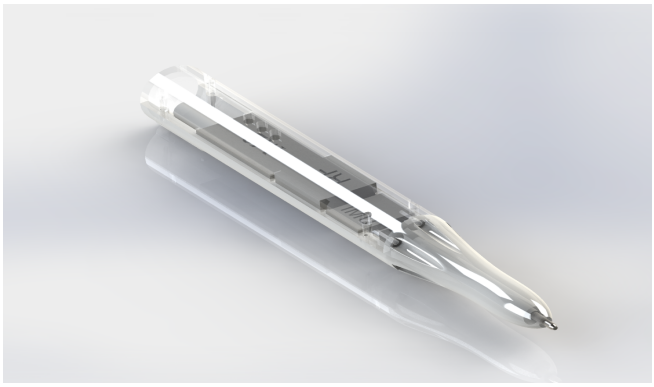

Figure 1. Rendering of the 3D printed model of the penn

1.2. Electronics

An ATMega32u4 microcontroller provides on-board processing and interfaces to the wireless module, IMU, switch, and indicator LEDs. Two Nordic nRF24LE1 transceivers handle wireless communication between the pen and base station computer. Inertial measurements are provided by an MPU6050 digital IMU with configurable gain triaxial accelerometers and gyroscopes. A Honeywell HMC5883L triaxial magnetometer provides a reference to earth's magnetic field for orientation. A switch behind the pen tip detects contact with the page for character segmentation. Three LEDs on the case display pen state. The pen also includes on-board batteries, power management, and an on-off switch.

1.3. Filtering the data

The raw data from the IMU pose issues for learning algorithms. Sources of variance can include posture, handedness, roll angle of the pen, and sensor drift. To minimize sources of error, we compute the trajectory of the pen tip on the page. Using various filtering and sensor fusion techniques we are able to compensate for gravity and perform attitude and position estimation. We use a Kalman Filter to estimate the orientation of the pen relative to gravity so that we can subtract the gravity components from the acceleration measurements.

1.4. Coordinate Transformation

Accelerations and rotations of the pen body measured at the IMU are translated to the tip by the following transform:

P: location of the IMU
O: origin of frame G on the page
frame G: SRT fixed to the page with origin O (ground frame)
frame M: SRT fixed to the pen at the IMU with origin at point P (pen frame)
Q: location of the pen tip
PQ: vector from the IMU origin to the pen tip

The acceleration of the pen tip in the ground frame is given by:

$$ {}^{G}\underset{\sim}{a}^{Q} = {}^{G}\underset{\sim}{a}^{P} + {}^{M}\underset{\sim}{a}^{Q} + {}^{G}\underset{\sim}{\alpha}^{P} \times \underset{\sim}{\mathrm{PQ}} + 2{}^{G}\underset{\sim}{w}^{M} \times {}^{M}\underset{\sim}{v}^{Q} + \ldots $$

$$ + {}^{G}\underset{\sim}{w}^{M} \times \left({}^{G}\underset{\sim}{w}^{M} \times \underset{\sim}{\mathrm{PQ}} \right) $$

Since point Q (the pen tip) is fixed in frame M (the pen frame), ${}^{M}\underset{\sim}{a}^{Q}$ and ${}^{M}\underset{\sim}{v}^{Q}$ are both 0. The above equation simplifies to:

$$ {}^{G}\underset{\sim}{a}^{Q} = {}^{G}\underset{\sim}{a}^{P} + {}^{G}\underset{\sim}{\alpha}^{P} \times \underset{\sim}{\mathrm{PQ}} + {}^{G}\underset{\sim}{w}^{M} \times \left({}^{G}\underset{\sim}{w}^{M} \times \underset{\sim}{\mathrm{PQ}} \right) $$

The IMU measures accelerations and angular velocities of frame M relative to frame G expressed in frame M. Using the attitude estimate, these measurements can be rotated into frame G to produce ${}^{G}\underset{\sim}{a}^{P}$ and ${}^{G}\underset{\sim}{w}^{M}$. The angular acceleration ${}^{G}\underset{\sim}{\alpha}^{P}$ is obtained by numerically differentiating ${}^{G}\underset{\sim}{w}^{M}$.

The trajectory of the pen tip in the page frame is obtained by twice-integrating ${}^{G}\underset{\sim}{a}^{Q}$. Since the velocity and position are integrations of noisy signals they are prone to very significant drift over time. To counteract this drift, the position and velocity estimates are re-zeroed at the start of every character and both the velocity and position are high pass filtered. The resulting trajectory can be plotted in the plane to approximate the appearance of the character on the page. This trajectory image is sampled to produce a black and white image which is then scaled and blurred to produce the final learning features.

1.5. Preliminary Results

We are currently able to produce relatively accurate trajectory estimates. The integrated character image is prone to distortion due to velocity and position drift over the course of writing the character. Our current filtering schemes are somewhat successful at reducing integration drift and rejecting noise from vibration. Figure 2 shows a trajectory recovered 'g' that we were able to produce using our methodology.

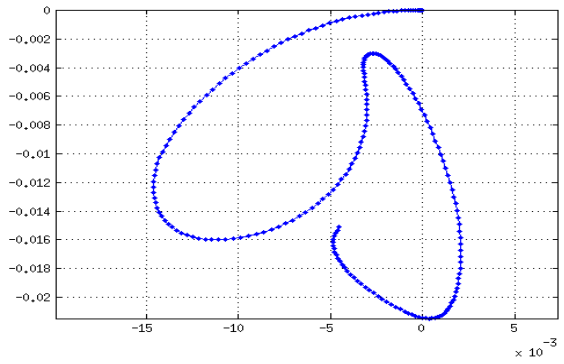

Figure 2. Recovered letter "g"

1.6. Next Steps

The next step in the project is to create a labeled test set. With a training set in hand, we will begin testing different learning algorithms so that we can find the most successful model. To this end, we will approach the learning component of this project in two ways. One method involves projecting the trajectory into an image and using learning algorithms such as neural networks. The other method involves using feature selection methods on the filtered signals before integration and differentiation. This method will require that we transform the time series data into a constant width feature vector. By interpolating, we are able to produce this result. Reducing dimensionality might prove to be an important step if we experience performance issues. In that case we intend to use either PCA or LDA to generate features. We intend to test the performance of various learning algorithms including Hidden Markov Models, Neural Networks, and Naive Bayes.

Acknowledgments

Attitude estimation Kalman Filter and initial approximation provided by **open-source** code from Sebastian Madgwick (July 2012), based on his paper **Estimation of IMU and MARG orientation using a gradient descent algorithm**.

For trajectory reconstruction, consulted Jeen-Shing Wang, Yu-Liang Hsu, and Jiun-Nan Liu's paper entitled **An Inertial-Measurement-Unit-Based Pen With a Trajectory Reconstruction Algorithm and Its Applications**.