BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2019
http://excel.fit.vutbr.cz

Signature Forgery - System for Imitation of Static and Dynamic Handwriting Characteristics

Jan Pawlus

Abstract
This project deals with designing and assembling a device for imitation of static and dynamic handwriting characteristics. First part describes design of a system, which is composed of a special pen targeted for getting static and dynamic characteristics of handwriting, working with these characteristics and an imitation with a 3D printer altered for this purpose. The key of this system is data obtained from user's handwriting with help of specific sensors - this data is used for reconstruction of the pen's trajectory during the writing as well as for analyzing the dynamic biometric attributes of the writer, followed by conversion to *G-Code*, executed by a 3D printer. This topic might be interesting because research about this specific topic, which would include a real demonstration of how a signature can be forged using dynamic handwriting characteristics, barely exists. The problem with preventing forgery is that we need to know the attack well - this is the point of this project and it is also why Brno's criminal police are interested in this topic.

Keywords: Signature forgery, Handwriting characteristics, Dynamic biometric attributes

Supplementary Material: Demonstration Video

*xpawlu00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Thanks to its old history, signature is still a leading way of giving an approval to legal texts, authorship expression etc. However, signature became an easy target for forgery and falsification. This happens because an impostor is not required to have some heavily trained skill or a big budget to complete a signature forgery - it just requires training to imitate static handwriting characteristics (signature's shape). This led to formation of a field that would analyze human handwriting - *graphoanalysis* [1], which focuses (among others) on person's dynamic biometric attributes like writing speed, pressure, tilt etc.

This project focuses on analyzing and imitating static and also dynamic handwriting characteristics of handwriting or signature. The goal is to design a system that obtains these characteristics and imitates them. For this purpose, a special pen with *accelerometer* and *gyroscope* was designed and assembled. From the sensors' output the system can analyze handwriting's characteristics, which are shape (static), speed and tilt (dynamic). Afterwards, from these attributes the signature is reconstructed and accurated with an actual scan or photo of the original signature, followed by imitation by a 3D printer specially altered for this purpose. The output should be evaluated by a person from the *graphoanalysis* field.

There are no known existing researches about this specific topic - the most similar solutions are *signing machines* (or *autopens*) used mainly by politicians

and artists [2]. The output is the same - a machine replicates a signature (although it is unclear which dynamic handwriting characteristics are replicated), the difference is that these machines work with a prepared model of the signature - that is why they have little use for this research.

There is for example a research about signature forgery, taking into account dynamic biometric features. Nine people agreed to provide their signatures, signing onto a touch screen, which captures dynamic characteristics, followed by attemps to train these characteristics and imitating them by selected forgers [3]. However, the methodology differs from this research a lot. The same goes for various applications focusing on handwriting recognition, which focus on signature *verification*.

2. Signature as biometric attribute

Biometry stands for automatic people recognition based on their characteristic anatomical attributes or characteristic behavior. An example of anatomical attribute could be fingerprint, eye's iris or retina. Signature falls into the second type - behavioral or dynamic (among with walking style etc). Anatomical attributes are far more popular these days because it is significantly easier to recognize them and to work with them - for example fingerprint does not change during person's life (when not taking into account skin diseases). On the other hand, dynamic attributes are harder to analyze mostly because of a high intra-class variability - person's signature is slightly different every time [4].

Nevertheless, signatures are used for person recognition nowadays. A signature is composed of *strokes*. A *stroke* is a continuous handwriting starting with putting the pen's tip to the surface and ending by putting the tip out of the surface. There are two types of systems working with signature as a biometric attribute - *off-line* and *on-line*. *Off-line* systems use just static signature attribute - it's shape, so it is enough to provide just photo or scan of the signature, whereas *on-line* systems take into account dynamic attributes (those which are changing during a signature) like speed, tilt and pressure. This means that within these systems, it is required to use some additional hardware - mostly a touch screen and a stylus [5]. The difference between these systems and this project is that for my research, I do not have any additional hardware to write on - dynamic handwriting characteristics are far easier to obtain when writing on a touch screen, where the trajectory, speed and pressure can be recorded.

Both *on-line* and *off-line* systems do post-processing to the input signal and use an algorithm for matching the processed input with their model - this is called *verification*. Signature is not suitable for *identification* due to its high *inter-class* and *intra-class* variability [5].

3. Design

First, it was crucial to decide which handwriting characteristics will be imitated. The most necessary ones is without a doubt shape (trajectory) - this is the first step for imitating a signature. If only the shape was used (from photo or scan), it would be impossible to imitate a handwriting properly because the order of strokes would be unknown - that is why there is a need for some sensors to use so the order of strokes is known. There are three methods of how person can write:

1. moving the whole hand while not rotating the pen (picture 1, left),
2. rotating the pen while moving the wrist at most (picture 1, right),
3. a combination of both.

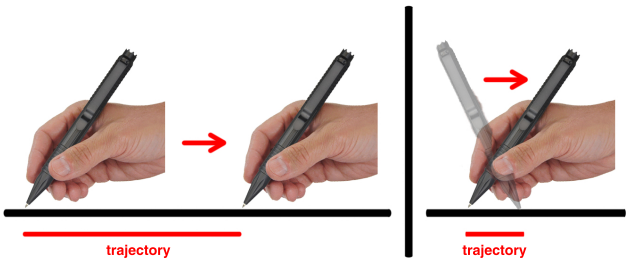

Figure 1. Writing methods.

The first method can theoretically be recorded by *accelerometer* - integration over time outputs *velocity*, whereas double integration outputs *displacement*. The second method could be covered by *gyroscope*, which measures *angular velocity* [6]. That is why *MPU-6050* was chosen - it is a module combining both of these sensors. It also includes a *digital motion processor*, which can post-process the raw signals from sensors - it is able to remove gravity effect from *accelerometer* and can compute *yaw*, *pitch* and *roll* angles, which describe a rotation of the module around its axis. This is how writing speed and tilt can be covered [7][8].

To get the trajectory, it is necessary to combine both of there sensors. If somebody wrote using only first method, *accelerometer* would be sufficitent as the double integration, *displacement*, would be the trajectory. If somebody wrote using only the second method, orientation of the module in space would represent the trajectory (knowing the distance between the module and the pen's tip, a point in space can be computed from the orientation). The orientation can be obtained

by constructing a rotation matrix created from the *yaw*, *pitch* and *roll* angles. The third method - that is how everybody writes - can be computed simply by combining both of these trajectories.

To sum it up, *MPU-6050* can provide data for trajectory, writing speed and tilt of a signature. Therefore, the requirement is to create a pen containing this module. This results into another problem - writing detection. That is why a micro-switch, which switches on when actually writing, is placed into this pen. Thanks to this fact it is also possible to divide the handwriting into *strokes*.

Knowing how the pen should work, a mechanism described in the picture 2 was designed. The *MPU-6050* module sits in the bottom of the pen and the micro-switch on the top. When writing, the ink container makes the micro-switch go on. Afterwards, the additional springs push the ink container back down again as the micro-switch goes off. This design allows to obtain data from the *MPU-6050* module alongside with the writing detection from the micro-switch when wiring all the components to *Arduino*.

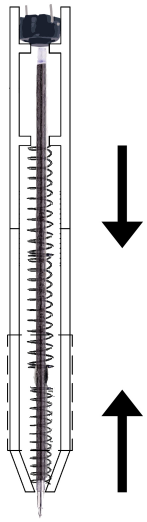

Figure 2. The pen mechanism.

Reconstructed trajectory is presumed to be inaccurate because of sensors' measurement errors, therefore the next step is to improve the signature's shape with photo or scan of the original signature. This demands image processing including cropping, binarization and thinning, and finally mapping the reconstucted trajectory to the image. After that, a complete accurate signature is available with its dynamic characteristics. The last step is the imitation - there were a few device possibilities to choose from (robot arm, plotter), but I decided to use a 3D printer containing a pen holder. The choice resulted in the fact that there will be no possibility to imitate the tilt but the 3D printer has plenty of other utilization and can imitate the writing speed. Moreover, when using Z axis of the 3D printer, it is possible to imitate all the handwriting's *strokes*.

4. Assembly and implementation

First, it was necessary to assembly the pen. A 3D model of the pen's torso was created according to the designed mechanism, followed by 3D printing on faculty printer. Afterwards, the micro-switch and the *MPU-6050* module were placed inside and soldered to connectors for wiring the pen to *Arduino*. The pen can be seen in the picture 3 (although with a lot of tape, which can be removed and replaced with glue). The process of creating the pen was iterative - two prototypes were created before assembling this final version as I was searching for the ideal pen's design and mechanism.

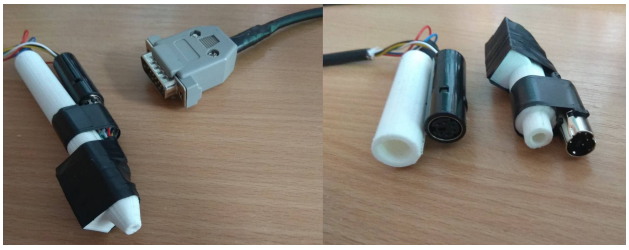

Figure 3. The pen for collecting static and dynamic handwriting characteristics.

Afterwards, signature characteristics were reconstructed according to the mathematical model described in section 2. Apart from *Arduino* code for collecting the sensor data, *Python 3* (with help of `numPy`, `sciPy` and `openCV` libraries) was used for the rest of the implementation. Unfortunately, during the implementation, the biggest problem of this research occured: the sensors are not accurate enough.

Accelerometers suffer from a lot of noise. After first integration, the error, which originates from noise, reflects linearly, so double integration makes error's impact exponential. There are possibilities of using filters for smoothing the signal like *low-pass* filter (and many more), the truth is that, as I found out, they simply cannot reduce the error enough. The only solution that turned out to be working was to implement an algorithm that would remove the linear trend when the module is lying still [9]. It is necessary to obtain a few acceleration values before the writing detection switches on, integrate these values over time to get the linear trend and substract these values from actual *accelerometer's* output while writing.

On the other hand, *gyroscopes* suffer from *drifting* - this means that their output changes during the measurement even when lying still. The second problem

I had to deal with using the *gyroscope* was *gimbal lock* [10]. The *MPU-6050* module is expected to be used in a horizontal position, but in my design it is placed vertically. Thus, the *gimbal lock* problem occurs, resulting in impossibility to measure all three angles *yaw*, *pitch* and *roll*. This problem was solved by a restriction of using the pen - when writing, the pen needs to be rotated around the *yaw* axis with the module facing the writer. In that case, I can assume that the *yaw* angle is always a zero value. With this assumption, the tilt can be calculated (and also the trajectory).

After trajectory reconstruction, various algorithms were implemented for the output improvement. For instance, the trajectory part from gyroscope actually projects on a globe, also it could be useful to rotate the computed points, shear them and so on. All of those post-processing steps are parametrized, so after the reconstruction, the user can see the result and alter the parameters. The result can be seen in the picture 4 - on the left side the real writing, on the right side the reconstructed one.

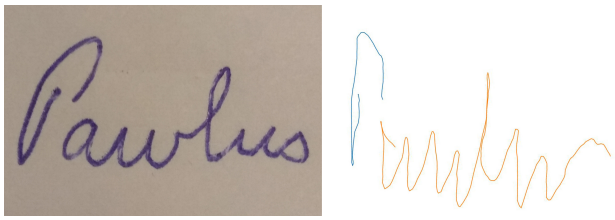

Figure 4. The real and the reconstructed writing

Even though the result is quite good given the circumstances (sensors are not precise enough and other problems), it is still obviously far away from the real writing. That is why I implemented an image processor that finds the writing on provided picture, applies thresholding, cropping and thinning, followed by mapping the reconstructed trajectory onto the processed image.

This turned out to be another challenging problem. At first I tried *feature matching*, but the results were not satisfying at all. Therefore, I decided to implement my own algorithm that interpolates the reconstructed trajectory points with a *B-spline* curve [11] and evaluates approximately the same amount of points as the processed image has (pixels). Then, when knowing the beginning point in the processed image from reconstructed trajectory, it is possible to go through the pixels until a crossing of pixels occurs. In this case, slope of the potentially next pixels (picture 5, left) is compared with slope of the corresponding points in reconstructed trajectory (picture 5, right). According to slope differce, the correct pixel is chosen and the process repeats until reaching end pixel of the image.

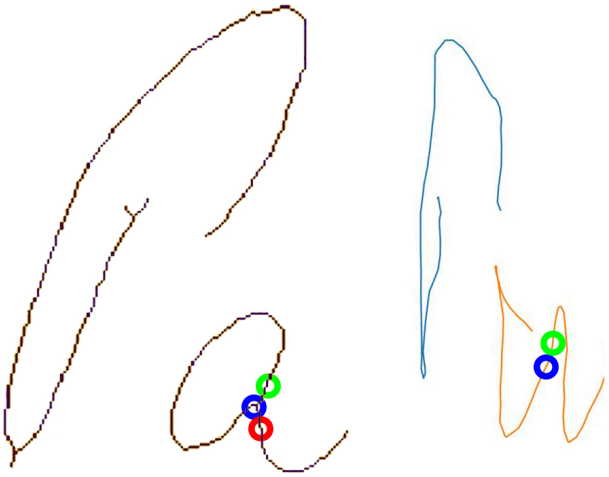

Figure 5. Mapping processed image onto reconstructed trajectory.

The problem is that the reconstructed image does not have to (and often does not) correspond with the reconstructed image proportionally, which leads to wrong choice of pixel in the image. Fortunately, the wrong decision can be detected because in this case, mapping never finds the ending pixel. That is why a little bit of a brute-force and randomness was brought into mapping. When the ending pixel is not reached, the mapping process starts all over again, although with slightly different points from reconstructed trajectory - thanks to the *B-spline* interpolation, it is possible to evaluate points from the curve with different density in different parts. This results in altering the proportions of reconstructed trajectory - in each mapping iteration, different set of points is chosen (parameters determining density in each part are randomly generated), meaning that sooner or later the mapping comes to an end when reaching the ending pixel. In the end, *Savitzky–Golay* filter is used for smoothing the points (picture 6, left)

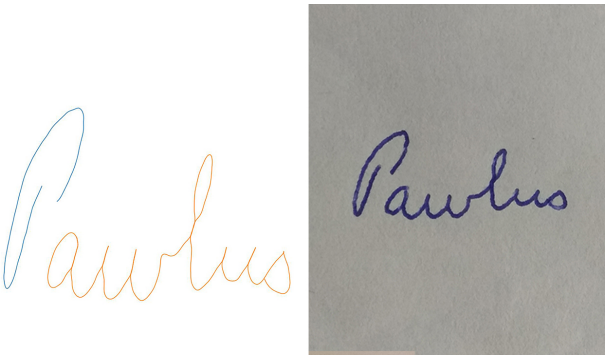

Figure 6. Mapped handwriting and imitated handwriting by the 3D printer.

The last step is to actually imitate the handwriting. The 3D printer I am using operates with *Marlin*

firmware, so it was necessary to study *G-Code*[1]. The point is to control the X and Y axes to move through points from reconstructed handwriting (also the Z axis, when signature is divided into multiple *strokes*). With each linear move the printer does, it is possible to set the speed by milimeters per second. This is how the handwriting is imitated with static and dynamic characteristics (picture 6, right). The imitation result is still subject for improvement.

5. Conclusion

This article briefly described my work on system for imitating static and dynamic handwriting characteristics. The first part focused on design of this system. It was crucial to think about the design really well as the assignment was quite general - all the specifics had to be designed well, otherwise the project could become impossible to finish - from the pen with correct sensors through the trajectory reconstruction theory to imitating device choice. This is why choosing this topic was definitely risky.

The result is promising so far as I was able to reconstruct a handwriting from my pen and also imitate it with a 3D printer. There is still some work to be done, mainly when speaking about reconstructing an actual signature with multiple strokes. Moreover, an opinion from somebody from the *graphoanalysis* field is still missing. But the whole point of this project is the research whether it is even possible to do such thing as mechanical signature forgery using dynamic handwriting characteristics - my work proved that it is and moreover, an impostor is not even required to have a big budget. This is actually why more attention should be paid to this topic because signatures' popularity does not seem to be decreasing.

There are a lot of possible improvements to the design, starting from the pen - instead of the microswitch, a *barometer* could be used so another handwriting characteristic can be imitated - pressure during writing. Also having a bigger budget, some better quality sensors could be obtained and even the whole pen could be manufactured in a way that a writer would not distinguish it from a classic pen - currently the pen's purpose is for testing but it could be assembled for serving a real forgery. Finally, a robotic arm could be used instead of a 3D printer so the computed tilt could be utilized. The topic is definitely not explored enough.

[1] *Marlin G-Code* - http://marlinfw.org/meta/gcode/

Acknowledgements

I would like to thank my supervisor prof. Ing. Martin Drahanský, Ph.D for his time to consult the progress. Another thanks belongs to Ing. Michal Dvořák for creating the 3D model of the pen and Ing. Tomáš Goldmann for help during the model printing.

References

[1] John Jensen. *What is graphoanalysis*. https://www.tokenrock.com/graphoanalysis/.

[2] Jimmy Stamp. President obama's autopen: When is an autograph not an autograph?, Jan 2013. https://bit.ly/2l6vcyA.

[3] Luiz Felipe Belem de Oliviera and Richard Guest. An assessment of dynamic signature forgery creation methodology and accuracy, 2015. https://www.researchgate.net/publication/278683948_An_assessment_of_dynamic_signature_forgery_creation_methodology_and_accuracy.

[4] Martin Drahanský. *Úvod do biometrických systémů*, 2018.

[5] Martin Drahanský. *Rozpoznávání podle písma a podpisu*, 2005.

[6] Ryan Goodrich. *Accelerometer vs. Gyroscope: What's the Difference?*, 2018. https://www.livescience.com/40103-accelerometer-vs-gyroscope.html.

[7] Steven M. LaValle. *Yaw, pitch and roll rotations*. http://planning.cs.uiuc.edu/node102.html.

[8] Arduino. *MPU-6050 Accelerometer + Gyro*. https://playground.arduino.cc/Main/MPU-6050.

[9] Arthur Charpentier. *What is a linear trend, by the way?*, 2017. https://freakonometrics.hypotheses.org/50003.

[10] Eric M. Jones and Paul Fjeld. *Gimbal Angles, Gimbal Lock, and a Fourth Gimbal for Christmas*, 2000. https://www.hq.nasa.gov/alsj/gimbals.html.

[11] Nicholas M. Patrikalakis, Takeshi Maekawa, and Wonjoon Cho. *B-spline curve*. http://web.mit.edu/hyperbook/Patrikalakis-Maekawa-Cho/node17.html.