

# SENIOR CAPSTONE TECHNOLOGY REVIEW

NOVEMBER 3, 2019

## GESTURE RECOGNITION KEYBOARD

OREGON STATE UNIVERSITY  
CS 461, FALL 2019

PREPARED BY

**GROUP 11**

ZACHARY HORINE

### **Abstract**

This paper explains several topics relative to our project objective, which is to create a gesture recognition keyboard for mobile devices. Specifically, the topics of data storage, ground truthing, as well as letter and grammar frequencies were examined. The first section of this paper outlines the options for data storage in a mobile app, SQLite databases, using application key pairs, and storing individual files in the device's native file system. The second section explores ground truthing, and how we can simplify our data, and make it easier to match against. The final section explores grammar and letter pairs and frequencies. This data is examined and compared to the evolution of the modern computer keyboard. This section also explores how we can use this information to pair together gestures that flow well, and are more conducive to faster input speeds.

## CONTENTS

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>II</b>	<b>Data Storage Method</b>	<b>2</b>
II-A	SQLite Databases . . . . .	2
II-B	External Files . . . . .	2
II-C	Key Pairs . . . . .	3
II-D	Best Option . . . . .	3
<b>III</b>	<b>Categorization engine - how to find a ground truth</b>	<b>3</b>
III-A	What is a Ground Truth? . . . . .	3
III-B	What is included in a ground truth? . . . . .	3
III-C	How can we apply this? . . . . .	4
<b>IV</b>	<b>Grammar statistics (frequencies and common patterns)</b>	<b>4</b>
IV-A	Keyboards . . . . .	4
IV-B	Frequency in Words . . . . .	4
IV-C	Conclusions . . . . .	4
	<b>References</b>	<b>5</b>

## I. INTRODUCTION

The purpose of this paper is to detail research topics to be focused on for our project. These topics each provide insight into the technologies and systems that will need to be incorporated into our project. This paper will cover three of the 12 total topics to be researched for our project. The topics to be covered are (ii) Data storage Methods (iii) Categorization - Finding the Ground Truth and (iv) Grammar Statistics. All of these topics provide unique insight into a particular area, and expand upon previous knowledge to create a base of understanding that our group can work from.

## II. DATA STORAGE METHOD

In order to persist gesture data inside our app, we will need to store a set of gesture point arrays that encode the base data for each gesture, and allow the system to match the gesture that the user has input, and reference it to a character. There are many ways to do this, but the most commonly used are (a) an SQLite database, (b) external files or (c) as a set of key pairs stored inside the app configuration files. Both Android, iOS support all three of these methods, and Windows devices can be configured to support them, although they don't support SQLite databases by default.

### A. *SQLite Databases*

SQLite databases are an efficient way to store highly structured data. In a database, there is a set of tables, that are linked together by a set of relations. these relations allow for the data to be read, interpreted and added to efficiently. [1] SQLite in particular has the benefit of being server-less, and lightweight. because of this, it can be implemented in mobile apps and services without much concern for process, power or storage cost. [1] Because of this, SQLite runs very well in both the Android and iOS architecture, and has been made accessible and easy to use for developers wanting to store relational data in their apps. Windows based systems don't have explicit support for SQLite, although there are libraries that implement similar functions. [2]

Overall, an SQLite database would be a good option to store our data in, as they are efficient and easy to use, but we would have to come up with a database schema that fit our data, which could be difficult.

### B. *External Files*

External files are extremely useful when it comes to application design. sometimes, when the data that you have doesn't fit into a specific data model, it can be easier to store it in a format of your choosing and read it in yourself.

There are some drawbacks to this method though. Depending on where you store your data, the data can be accessed by the user and be moved or deleted. Although on all systems, there are private data stores that can be used to keep your data safe. These files often sit close to the application code so that they share the same permissions. One of the other drawbacks is that it is more difficult to keep track of your data, and where it is stored. This could be remedied by using one of the other data storage methods though, as it would provide a 'link' to the location of the data.

Overall, the concept of storing our gesture models in their own files makes the process of reading just one gesture in more efficient, and allows us to more easily add and remove data throughout the development process.

### C. Key Pairs

Key pairs are an easy way to store small amounts of data, and are useful for things like usernames or device keys. They are very specific on the types of data that they can store, and generally allow for personalized data to be stored without resorting to creating a whole database or file system.

On iOS, these keys are stored in a '.plist' file within the app package. Because of the way the file is stored, there is a maximum file size of 4GB (file-system restriction). [3] This, along with the fact that the file has to be read in every time the app loads, or the first time data is requested, means that the amount of data needs to be limited, and shouldn't be used to store large data structures. The only difference with Android devices is that the 'Shared Preferences' objects are stored in an xml file within the system. [4] Otherwise they act the same as an iOS device. [5] The case is similar with windows devices, although their data is stored more centrally, instead of being put close to the rest of the app's data. Windows devices store their data in the 'appdata' folder of the user, and can be configured to store data as key pairs or as custom files. [6]

Overall, while this may be a good method for storing some user data, it is not the best way to store gesture models themselves, as they are made up of a larger table of points, and there are too many of them to efficiently store in a model like this.

### D. Best Option

After reviewing these methods, I believe that there are a couple different options we can pursue. The most promising option is to store our gesture models in external files, and creating references to these files and their locations in a set of app key pairs, or as static variables in our code. Doing this would allow us to format the data in a way that makes sense to us, all while maintaining a reference and organization to our files.

## III. CATEGORIZATION ENGINE - HOW TO FIND A GROUND TRUTH

### A. What is a Ground Truth?

In general, a ground truth is a data set that defines the base observations that all observations made after its establishment are based on. The concept of a ground truth is used extensively in image recognition, and is heavily related to a scientific assumption. In this way, it can be used as a basis of observation, and allows the observer to determine whether or not the data that they have gathered fits the model or not. [7]

### B. What is included in a ground truth?

At its base, the ground truth is the simplest, most basic, and neutral form of the data expected. However, one set of perfect data is not enough to define the base case of the model. A ground truth has to not only include the optimal solution, but it also has 'sub-optimal' solutions that are still in the threshold. [8] It also includes solutions that are not in the data-set, which help to filter out data that doesn't fall within the confines of the desired outcome.

In image recognition, the ground truth also has multiple layers. All of the layers help to filter the data, and make it easier to pinpoint what is different from the ground truth. By looking at things such as the diffusion, reflectance, shading and the specular components of an image, it is easier to see what part of the image differs from the ground truth, which makes it easier to correct later. [9]

### *C. How can we apply this?*

Detecting gestures is difficult for many reasons, but they all lead to one thing: no two movements are ever the same. A lot of this has to do with sensor accuracy, as the accelerometer in a phone is 'precise' but not very accurate. The little irregularities in motion sensing stack up, and very quickly make pulling the true motion out of the noise very difficult. The other complication is that humans never recreate the same exact gesture, and no two humans will perform the same gesture the same way. because of this, the motion itself is not accurate to the model. This is why a simple 'perfect' gesture can't be used as the ground truth. By using a ground truth in multiple axis, as well as tolerances and negatives, we can eliminate bad data, and generate the most probable path.

## IV. GRAMMAR STATISTICS (FREQUENCIES AND COMMON PATTERNS)

### *A. Keyboards*

One of the important things when designing an input device is the layout of the input 'keys'. Many traditional keyboard layouts were designed around the English language, and the letter frequencies that are used to type some of the most common words. One of the most common examples of this is the Devorak keyboard, which prioritizes vowels and commonly used consonants. [10] we will have to do a similar thing with our keyboard, but we will also have other issues to attend to. Because of the nature of gestures, there are certain gestures that will be difficult to detect if they come after another gesture, and some that are easy and flow into one another. In this way, we are dealing with some of the issues that came along with typewriters (frequent jamming), and the supposed reason that the qwerty layout was invented. In reality, the layout was invented to slow down faster typists, which is exactly the opposite effect that we want to have. [11]

### *B. Frequency in Words*

Instead, we should pair more readily accessible gestures with the most commonly used characters, and associate the least used characters with 'more out of the way' gestures. We also need to look at the characters that commonly come after one another, and make sure the pair of gestures work well together. A study done by the College of Math at the University of Illinois shows how likely a given character is to come after another. In their study, there were a large number of characters that never appeared together, and a list of the most frequent ones, which included 'th', 'he', and 'an' as the top three. [12] This makes a lot of sense, as the three most common words in the English language are 'the', 'of' and 'and'. [13]

### *C. Conclusions*

If we can use this information to map our gestures to letters, then we can improve the efficiency of typing on our keyboard. By taking into account all of the lessons learned in keyboard design, and incorporating them into our app, we can create a better experience for our users.

## REFERENCES

- [1] “What is SQLite? Top SQLite Features You Should Know,” 2019. [Online]. Available: <https://www.sqlitetutorial.net/what-is-sqlite/>
- [2] “Use a SQLite database in a UWP app - Windows UWP applications,” Nov. 2018. [Online]. Available: <https://docs.microsoft.com/en-us/windows/uwp/data-access/sqlite-databases>
- [3] “iOS Data Storage Guidelines,” 2019. [Online]. Available: <https://developer.apple.com/icloud/documentation/data-storage/index.html>
- [4] Tarun Kumar, “Top 10 ways you should know to store data in Android,” Mar. 2018. [Online]. Available: <https://www.loginworks.com/blogs/top-10-ways-know-store-data-android/>
- [5] Obaro Ogbo, “How to store data locally in an Android app,” Sep. 2016. [Online]. Available: <https://www.androidauthority.com/how-to-store-data-locally-in-android-app-717190/>
- [6] “ApplicationData Class (Windows.Storage) - Windows UWP applications,” May 2019. [Online]. Available: <https://docs.microsoft.com/en-us/uwp/api/windows.storage.applicationdata>
- [7] Danilo Pena, “Ground Truth Versus Bias,” Apr. 2018. [Online]. Available: <https://towardsdatascience.com/ground-truth-versus-bias-99f68e5e16b>
- [8] S. Krig, *Computer vision metrics: survey, taxonomy, and analysis*, ser. The expert’s voice in computer vision. New York, NY: Apress, 2014, oCLC: 881828978. [Online]. Available: [https://www.embedded-vision.com/sites/default/files/apress/computervisionmetrics/chapter7/9781430259299\\_Ch07.pdf](https://www.embedded-vision.com/sites/default/files/apress/computervisionmetrics/chapter7/9781430259299_Ch07.pdf)
- [9] Roger Grose, Micah K. Johnson, Edward H. Adelson, and Willian T. Freeman, “Ground truth dataset and baseline evaluations for intrinsic image algorithms,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, Feb. 2009. [Online]. Available: <https://www.cs.toronto.edu/~rgrosse/iccv09-intrinsic.pdf>
- [10] Randy Cassingham, “The Dvorak Keyboard,” 2019. [Online]. Available: <https://www.dvorak-keyboard.com/>
- [11] Leah Welborn, “Why QWERTY?” Jun. 2011. [Online]. Available: <https://www.mentalfloss.com/article/27938/why-qwerty>
- [12] Jeffrey S. Leon, “Frequency of Character Pairs in English Language Text,” University of Illinois, Tech. Rep., 2008. [Online]. Available: [http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/char\\_freq2.pdf](http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/char_freq2.pdf)
- [13] “100 Most Common English Words (Learn 85% of English in 100 Days),” 2019. [Online]. Available: <https://www.rypeapp.com/most-common-english-words/>