

# Technology Review

Group#65

## Krisna's Part

### User Interface Toolkits

#### A. Options

There are three options for the user interface toolkits. The first option is the visual studio user interface toolkits. Visual studio user interface toolkits are built-in toolkit in visual studio that used for programmer to create a window application user interface. The second option is GTK+ toolkits. GTK+ toolkits are open source cross-platform widget toolkit for creating graphical user interfaces [1]. The last option is the IUP portable user interface. IUP is a computer software development kit that provides a portable, scriptable toolkit to build a graphical user interface [2].

#### B. Goals

We will use a user interface toolkit to help us create the graphical user interface of this project. The goal of having user interface is to create a presentable representation of the alignment data. The algorithm will output a raw alignment data and will be hard to understand without the help of graphical user interface. This will also make our project presentable to our clients and the expo. The user interface will be a crucial part for our project presentation.

#### C. Criteria

The criteria for user interfaces are complexity, accessibility, time commitment, and cost. These criteria are used to determine the value of using a particular user interface. Complexity is referring to the additional complexity being added to the code by using a particular user interface. Accessibility is referring to the price of the software, accessibility of the library, accessibility of the software, and the adaptation of a particular user interface toolkit in the programmer community. The time commitment is the amount of time needed to learn and get a better understanding on how to use a particular user interface toolkit. The cost will refer to overall cost and benefit of using a particular user interface toolkit.

#### D. Comparison Table

Toolkit Name	Complexity	Accessibility	Time Commitment	Cost
Visual Studio Toolkit	Least Complex	Accessible for free. Already have it in our computer.	Minimal	Least powerful toolkit, with the least learning curve. .
GTK+	Complex	Accessible for free online. The most popular toolkit for graphical user interface. .	Medium	More powerful than visual studio, simpler than IUP.
IUP	More complex	Accessible for free online. Less popular than GTK+ toolkit.	High	Most powerful option. A complete development tools. Have the most cost and learning curve.

#### E. Discussion

Visual Studio toolkit will have the least complexity because we will use Visual Studio as our development tool for the software. Hence, using the built-in toolkit will not require us to download any additional library, include file, and syntax when creating the user interface. Visual Studio is accessible for free with our ONID account. Our team will have visual studio installed in our computer from previous programming

classes. The learning curve for using this toolkit is shallow. Our team are familiar with the Visual Studio environment. This toolkit will require the least time commitment from us. Visual Studio Toolkit is the least powerful toolkit out of the three options. However, the cost of using this toolkit is minimal, which makes this toolkit an appealing option for our project. GTK+ will surely add more complexity to our project. However, GTK+ is a toolkit rather than a full development tool like IUP, this makes GTK+ less complex than IUP. GTK+ is an open source software that licensed under the GNU Lesser General Public License, which makes this software accessible for our project. This toolkit is accessible for free and has been one of the most popular toolkit for creating graphical user interface. There will be a learning curve when using GTK+ toolkit. I am not familiar and never use GTK+ before. Using GTK+ will required me to learn it from a basic level. GTK+ is a more powerful option that Visual Studio with a less complexity than IUP. IUP is a software development tool. This will add a lot of complexity to our project. IUP is accessible for free online. The learning curve is high, since we are now using a completely different software development tool. Since it will add a huge complexity to our project, the cost of using IUP is really high.

## F. Selection

Although a user interface is a critical part in our project presentation, this functionality is not a critical part of the project as a whole. Our clients strongly emphasize the quality of our algorithm. The most important piece of this project is the dynamic alignment algorithm and our clients encouraged us to spend most our time in developing and refining the dynamic alignment algorithm. Thus, using the toolkit with the least learning curve is encouraged for our project. This makes us decided to use the built-in visual studio toolkit to create our project's graphical user interface.

## Programming Language

### A. Options

Our clients give us the freedom to choose the programming language that we want to use to create this project. There are three options for programming language that we can use for the this project. The options for the programming languages are C, C++, and Assembly language.

### B. Goals

We are looking for a better programming language that will improve the quality of our project. The goal of using a particular programming language is to reduce the complexity of writing code for our project, while having an acceptable speed and memory spaces.

### C. Criteria

The criteria to evaluate those programming languages are complexity, speed, availability, and memory spaces. Complexity is the complexity of the syntax, function, method, and coding style of a particular programming language. Speed is the speed of a compiled software written in a particular programming language. Availability is referring to the compatibility with software and other development tools that we might use in working on this project. Memory is the size of compiled software written in a particular programming language.

### D. Comparison Table

Language Name	Language orientation	Complexity	Speed	Availability	Memory
C	Object oriented	High Level Language (More Complex)	The same with C++	More compatible with the hardware	Less
C++	Object oriented	High Level Language (Less Complex)	The same with C	Less compatible with the hardware	More

Assembly	No orientation	Low Level Language (Most complex)	Fastest	Completely compatible with the hardware	Least
----------	----------------	-----------------------------------	---------	---	-------

## E. Discussion

### F. Selection

## Statistical Analysis Method

### A. Options

There are three options of statistical analysis methods that we can use for this project. The first one is confidence interval. The second option is credible interval. The third option is tolerance interval.

### B. Goals

The goal of using a statistical analysis method is to find the credibility of the data that we got from the algorithm.

### C. Criteria

The criteria for the statistical analysis method are the complexity, difference, and the result provided by each statistical method.

### D. Comparison Table

Method Name	Type of Statistics	Complexity	Difference	Result
Confidence Interval	Frequentist			
Credible Interval	Bayesian		.	
Tolerance Interval	Can be both			

## E. Discussion

### F. Selection

Our clients ask us to use confidence interval.

## Jiongcheng's Part

### Hardware selection: Microcontroller

A microcontroller plays an important role in the system, which a microcontroller will have following functionalities in the system:

- Process input from the IMUs and output to the computer/display
- As a carrier-board to process the alignment algorithm

Therefore, it is critical to choose the most appropriate microcontroller for this system since there are limitation onto the both hardware and software. One of Arduino Pro Mini and Adafruit Pro Trinket will be most ideal boards and we have to consider the follow criteria for choosing the “best” one:

1. **Support Language**, we expect to use a general programming language for the microcontroller since using familiar programming language will reduce our time on addition research and the save time on working on the algorithm itself.
2. **Clock Speed**, the alignment algorithm is required to within around 500 milliseconds time from taking the input from the IRUs to the output of aligned data, that require the board has a fast speed and running any processes.
3. **I2C Protocol**, this is necessary to have on the board since the MPU-9250 (our selected IMU) require I2C protocol to communicate.
4. **Connection with PC**, this is necessary to have on the board since we will use computer to any software program.
5. **Size**, size is less important since this system is for demonstration purpose.
6. **Cost**, this is somehow important but as long as the price is within expected budget will be acceptable.

Model	Arduino Pro Mini	Adafruit Pro Trinket	ONE MORE
Core Chip	ATmega 328	ATmega 328	
Support Language	C	C	
Clock Speed	8MHz (3.3V mode)	12MHz (3V mode)	
I2C Protocol/Number	Yes/1	Yes/1	
Connection with PC	USB	FTDI/USB	
Size	33mm x 18mm	38mm x 18mm x 2mm	
Cost (U.S Dollar)	\$9.95	\$9.95	
Links	<a href="https://www.sparkfun.com/products/11114">https://www.sparkfun.com/products/11114</a>	<a href="https://www.adafruit.com/product/2010">https://www.adafruit.com/product/2010</a>	
Advantages	<ul style="list-style-type: none"><li>• Small size</li><li>• Easy to acquire</li><li>• Abundant resources</li></ul>	<ul style="list-style-type: none"><li>• Faster clock speed</li><li>• Have two option to connect with PC</li></ul>	
Shortages	<ul style="list-style-type: none"><li>• Slower Clock Speed</li><li>• Limited connection option with PC</li></ul>	<ul style="list-style-type: none"><li>• Larger Size</li><li>• Fewer resource of guidance</li></ul>	

By comparison, the Adafruit Pro Trinket will be more preferable since it has outstanding clock speed compares to the Arduino Pro Mini, although it has a larger size but this doesn't affect to its actual performance based on our project requirement, other than that, all other concerned criteria are same as the Arduino Pro Mini.

#### Hardware selection: Represent MEMS IRU

Model	InvenSense MPU-9250 CA-SDK Reference Board	Sparkfun IMU MPU-9250	DiyMall Mpu-9250 9dof Module
Accepted Power Supply (Voltage)	3.7V (rechargeable batteries)	2.4 - 3.6V	3-5V
Output	24-bit CRC Error correction	16 bits ADC	16 bits ADC
Communication Protocol	SPI/I2C/Bluetooth	SPI/I2C	SPI/I2C
Size	33mm x 18mm	Unknown	14.3 mm * 20.5 mm
Cost (U.S Dollar)	\$440.00	\$14.95	\$15.99
Detail Specification	<a href="https://store.invensense.com/datasheets/invensense/MPU-9250CA-SDK.pdf">https://store.invensense.com/datasheets/invensense/MPU-9250CA-SDK.pdf</a>	<a href="https://www.sparkfun.com/products/13762">https://www.sparkfun.com/products/13762</a>	<a href="https://www.amazon.com/Mpu-9250-Nine-axis-Attitude-Acceleration-Magnetic/dp/B017VT9TF4/ref=sr_1_3?s=industrial&amp;ie=UTF8&amp;qid=1479193223&amp;sr=1-3&amp;keywords=mpu+9250">https://www.amazon.com/Mpu-9250-Nine-axis-Attitude-Acceleration-Magnetic/dp/B017VT9TF4/ref=sr_1_3?s=industrial&amp;ie=UTF8&amp;qid=1479193223&amp;sr=1-3&amp;keywords=mpu+9250</a>
Advantages	<ul style="list-style-type: none"> <li>Embedded Microcontroller</li> </ul>	<ul style="list-style-type: none"> <li>Less expensive</li> </ul>	
Shortages			

#### Hardware selection: Represent MEMS IRU