B.Eng. Dissertation

## 3D PLOTTING OF AIRCRAFT ATTITUDE

Submitted by

MUHAMMAD OMER IQBAL

# ABSTRACT

Air accident investigations usually involve reading and interpreting large volumes of flight data. Visualising this data, especially in a geographic context makes its analysis easier. Therefore the Air Accident Investigation Board (AAIB) needed a tool that would render a 3 dimensional model of flight data from their datasets. The data includes measurements like latitude, longtitude, altitude, heading, roll, thrust etc over time. To meet this criteria I designed a Clojure based application that takes a dataset in csv format and renders aircraft attitude over time on a map. The application generates a KML (Keyhole Markup Language) file which can be opened on Google Earth and displays a 3d model of the data.

SUBJECT DESCRIPTORS:

- Specialized information retrieval

- Information visualization

- Geographic visualization

KEYWORDS: Clojure, KML, Visualisation, Aircraft Attitude, XML generation, 3d Modelling

# PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

Put your publications from the thesis here. The packages `multibib` or `bibtopic` etc. can be used to handle multiple different bibliographies in your document.

*We have seen that computer programming is an art,*
*because it applies accumulated knowledge to the world,*
*because it requires skill and ingenuity, and especially*
*because it produces objects of beauty.*

— **?** [**?**]

## ACKNOWLEDGEMENTS

[ November 4, 2013 at 4:02 – classicthesis version 2.0 ]

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LISTINGS

## ACRONYMS

DRY   Don't Repeat Yourself

API   Application Programming Interface

UML   Unified Modeling Language

KML   Keyhole Markup Language

AAIB   Air Accident Investigation Board

Part I

## THE PROBLEM

You can put some informational part preamble text here. Illo principalmente su nos. Non message *occidental* anglo-romanic da. Debitas effortio simplificate sia se, auxiliar summarios da que, se avantiate publicationes via. Pan in terra summarios, capital interlingua se que. Al via multo esser specimen, campo responder que da. Le usate medical addresses pro, europa origine sanctificate nos se.

# INTRODUCTION

## 1.1 BACKGROUND

Aircraft systems record large quantities of data during flight. A lot of this data is necessary in the system 's proper functioning. It takes an even more significant role when investigating air accidents. As this data establishes the causes behind those accidents, it aids measures to prevent such accidents from occurring. That is the primary reason for why recovery of Flight Data Recorders [1] is usually a very high priority for aircraft investigations.

The data itself features many components tracked over time, including but not limited to the following[2]:

- Latitude

- Longtitude

- Altitude

- Heading

- Pitch

- Roll

- Engine Thrust

- Throttle Lever Position

As is probably evident, analyzing this largely numeric data by itself would be a painstaking task. Primarily because finding patterns in numbers through observation is inherently difficult. Also because the data set is of a substantially large size [3].

Therefore the need for a tool to visualise this data becomes immediately obvious.

The Air Accident Investigation Board (AAIB), a branch of Singapore's Ministry of Transport performs analysis on such flight data, and requested an application that would render relevant flight data quickly and in a portable manner.

---

1 Popularly referred to as 'Blackboxes'
2 The following components were part of the sample dataset I was given for testing.
3 I will be cautious before calling it "Big Data".

3

## 1.2 SYSTEM REQUIREMENTS

AAIB needed the application to meet certain criterias and contain certain core features. These included:

1. GOOGLE EARTH Compatibility. Google Earth is a popular "virtual globe, map and geographic information program". It provides detailed 3d projections of Google's satellite images on a spherical globe, overlaid with huge quantities of textual and pictorial data varying from international boundaries to streets. There are many good reasons for this requirement including:

   - Cross Platform. It works and is supported on Windows, OSX and Linux [4]

   - Extensible through the Keyhole Markup Language (KML) [5]

   - Free[6]

   - AAIB's familiarity with it

   - Offline use. This is quite important as internet access is not guranteed on accident sites

2. CSV[7] based input. The system should be able to use large datasets encoded in CSV.

3. The system should display the data in its geographical location as vectors. The visualization should be able to reflect the actual data in an intuitive way. For instance, vectors at each data point should represent roll, heading, engine throttle

4. Users should be able to select which vectors they want to view.

5. The data points should be manually adjustable from within google earth. This is particularly relevant for data points near landing strips, where measurement errors may cause an offset in the rendered visualization

---

4 Atleast the desktop version is supported on all three major Operating Systems.
5 A lot more on this later
6 As in beer, not speech
7 Comma Separated Values

Part II

## DESIGNING A SOLUTION

You can put some informational part preamble text here. Illo principalmente su nos. Non message *occidental* anglo-romanic da. Debitas effortio simplificate sia se, auxiliar summarios da que, se avantiate publicationes via. Pan in terra summarios, capital interlingua se que. Al via multo esser specimen, campo responder que da. Le usate medical addresses pro, europa origine sanctificate nos se.

# RENDERING STRATEGIES

As with any project, the first few stages of the project involved researching on existing technologies that either implemented the features specified in by AAIB in their requirements, or aided in developing those features. As Google Earth compatibility was a major requirement for the project, my research started with exploring methods to render custom data on Google Earth's platform.

## 2.1 KEYHOLE MARKUP LANGUAGE

The Keyhole Markup Language (*KML*) is a subset of XML, used to display geographic data in an Earth Browser, like Google Earth. It uses a tag based structure with nested elements that represent certain geometric and descriptive constructs. The KML reference is very reasonably documented with sample code demonstrating it's usage.

KML supports rendering the following features:

1. Placemarks

2. Descriptions

3. Ground Overlays

4. Paths

5. Polygons

KML uses latitudes and longtitudes as a coordinate system for positioning elements, which works very well with geographic data.
However KML is primarily designed as a means to display data, as opposed to performing interactions with the data. The only interactions it allows are those that Google Earth features, i.e. zooming, panning and moving to a certain point on the globe. Clicking on placemarks with descriptions opens a modal that renders the description, which can be written in a subset of HTML.

## 2.2 GOOGLE EARTH'S BROWSER PLUGIN

With web browsers becoming increasingly powerful and technologies like WebGL that allow efficient 3d rendering on the browser, Google launched the Google Earth Browser Plugin. The browser plugin by itself allows pretty much the similar core functionality as Google

Earth's Desktop Application. The primary difference is that the virtual globe can be embedded within an HTML web page.

By itself the browser plugin may not seem to accomplish much. However Google also released the Google Earth API, a javascript library that provides access to various features in Google Earth. The browser plugin when combined with Google Earth's API becomes a very powerful tool to produce complex applications that render and allow interaction with geographic data.

For instance one of the sample toy applications provided as documentation for the API is a car simulator, where you drive a car over the virtual globe's surface.

Such user interactions are possible by combining interactions provided by the DOM, and using them to render models using the Google Earth API.

However, the browser plugin is not a silver bullet. One of it's most lacking features is offline usage. The browser plugin does not work offline, unlike the desktop application which caches data for offline use.

## 2.3    CHOOSING A RENDERING STRATEGY

After researching on rendering methods available I had to choose which method was a better fit for this project. From my research the following rendering strategies seemed initially viable:

1. Generate KML and use it's Path and Polygon constructs to render the various vectors that AAIB requested

2. Use Google Earth's javascript API to render the same data.

After revisiting AAIB's application requirements, I decided that between producing KML and using Google Earth s Browser Plugin, generating KML was a significantly better option. The rationale guiding the decision was based on the following factors:

- Offline usage. As mentioned before, the browser plugin does not feature offline usage. This is in contrast to rendering KML documents on Google Earth's Desktop client which works well offline. Considering offline usage was an important part of the application requirements, this was a major reason.

- Portability. KML is a well defined format that can be rendered on many other earth browsers. Furthermore, the browser plugin is only supported for Windows and Mac OS. Linux [1] was not

---

1  Which I am a huge fan of

yet supported, and I did not want to add that constraint to the application without good reason. In contrast, Google Earth is supported on Windows, Mac OS and Linux.

# CHOOSING THE STACK

This decision took a while. And rightfully so, as it would affect all decisions from that point. Choosing the right language and framework could have more than substantial effects on the system's architecture.

After quite a bit of research and debate, I decided to implement the system using CLOJURE, a relatively new lisp class language than runs on the JVM.

## 3.1 CLOJURE?

The choice of implementing this project in clojure may seem peculiar. However there are a number of features that clojure and it's ecosystem that fit very well with this particular project. Before jumping to those features, a quick overview of clojure is in order.

Clojure is a lisp dialect, created by Rich Hickey. It is a general-purpose, functional programming language that runs on the Java Virtual Machine (JVM) [1]. Clojure focuses on programming with immutable data structures and provides a number of persistent hash-trie based data structures that allow efficient functional programming.

Clojure was designed to be a real world language, as opposed to a purely academic one.

---

1 Clojure runs in other environments like Microsoft's Common Language Runtime (CLR) and Javascript engines as well

Part III

APPENDIX

# A

APPENDIX TEST

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

## A.1 APPENDIX SECTION TEST

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

*More dummy text*

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse

15

| LABITUR BONORUM PRI NO | QUE VISTA | HUMAN |
| --- | --- | --- |
| fastidii ea ius | germano | demonstratea |
| suscipit instructior | titulo | personas |
| quaestio philosophia | facto | demonstrated |

Table 1: Autem usu id.

Listing 1: A floating example

```
1  for i:=maxint to 0 do
   begin
   { do nothing }
   end;
```

viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

## A.2 ANOTHER APPENDIX SECTION TEST

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

# DECLARATION

Put your declaration here.

*Singapore, November 2013*

Muhammad Omer Iqbal,
November 4, 2013