

# Visualizing Unit Quaternions

*by* Nathaniel Richards

---

|                |                                                                                      |                 |       |
|----------------|--------------------------------------------------------------------------------------|-----------------|-------|
| FILE           | 15358_NATHANIEL_RICHARDS_VISUALIZING_UNIT_QUATERNIONS_78196_1910424940.DOCX (10.93M) |                 |       |
| TIME SUBMITTED | 21-APR-2016 12:03PM                                                                  | WORD COUNT      | 5929  |
| SUBMISSION ID  | 56190770                                                                             | CHARACTER COUNT | 34313 |



**Prifysgol Cymru**  
Y Drindod Dewi Sant  
**University of Wales**  
Trinity Saint David

**Visualizing Unit Quaternions**

by

Nathaniel James Priday Ricahrds

Supervisor: Dr Glenn Jenkins

Second Marker:

Project submitted as part of the requirements for the award of BSc(Hons) Computer  
Games Development

April 2016

## Abstract

The objective of this project is to show unit quaternion on the surface of a unit sphere environment/scene and to determine whether it can be done in a way that does represent it in a manner that useful. The methods used throughout this project are a literature review to analyse how to portray 3d information to give it correct context of the 3 dimensional objects. Analyse of how others have visualised unit quaternions or how others use unit quaternions to represent data in their papers. The project did show different ways to visualize unit quaternions but whether the representation is accurately portrayed is partially down the user of the program to deem usefulness. The method for developing the software was agile which allowed parts of the program to be isolated and tested before being used as one single program. The project does conclude that the representation of the unit quaternions on the surface of a unit sphere is accurate but there is room for improvement in the form of features that were not implemented. The recommend future works which include adding the features that were system requirements show greater ability to compare this version of the projects visualizations to splines and quaternion calculations.

## Contents

|                                   |    |
|-----------------------------------|----|
| Abstract.....                     | 2  |
| Introduction .....                | 5  |
| Literature Review.....            | 5  |
| Introduction .....                | 5  |
| Information Representation .....  | 6  |
| Projection.....                   | 8  |
| Orientation.....                  | 9  |
| Euler Angles .....                | 9  |
| Quaternion.....                   | 10 |
| Unit quaternion.....              | 12 |
| Unit sphere.....                  | 12 |
| Rotations Calculations .....      | 13 |
| Fixed and Moving Rotations .....  | 13 |
| Neural Networks .....             | 13 |
| Character Motion.....             | 15 |
| Curves and splines .....          | 17 |
| Aircraft Control .....            | 17 |
| Robotic Arms.....                 | 18 |
| Conclusion.....                   | 19 |
| Methodology.....                  | 19 |
| Design & Implantation .....       | 20 |
| First iteration.....              | 21 |
| Second iteration.....             | 23 |
| Third iteration .....             | 24 |
| Fourth iteration.....             | 24 |
| Fifth Iteration .....             | 24 |
| Sixth iteration.....              | 25 |
| Seventh iteration .....           | 26 |
| Final Interface .....             | 27 |
| Evaluation .....                  | 27 |
| Conclusion & Recommendations..... | 27 |
| Reflection .....                  | 29 |
| References .....                  | 30 |

Bibliography ..... 31

Appendix A ..... 33

User Requirements ..... 33

System Requirements ..... 34

## Introduction

The project is about showing different ways to portray unit quaternions on a unit sphere and how with the overall goal of showing how unit quaternions data is seen on unit sphere in a variety of ways to allow any user to understand unit quaternions.

The Literature review goes into how 3 dimensional information affect how it is seen by anyone viewing it and ways to make it believable also how unit quaternions have been viewed in work done before this project and also various other ways.

The methodology explains quickly what methodology was used for this project and why it is most suited to it.

The design and implementation explains the steps of creating the program and how certain iterations went well and others did not and also why they did not.

## Literature Review

### Introduction

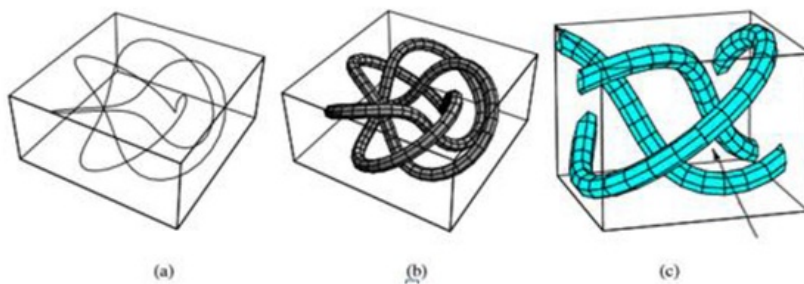
In order to understand the problem of visualizing unit quaternion to objects in 3 dimensional space or a unit sphere. There must be a understanding of how unit quaternions are portrayed and used in current fields which unit quaternions are not the aspect being visualized but are affecting how the objects is being represented in a rotational and positional sense.

Quaternions and by relation unit quaternions are a slightly confusing thing to understand and visualize due to the hyper complex nature and are normal used to represent rotation. These

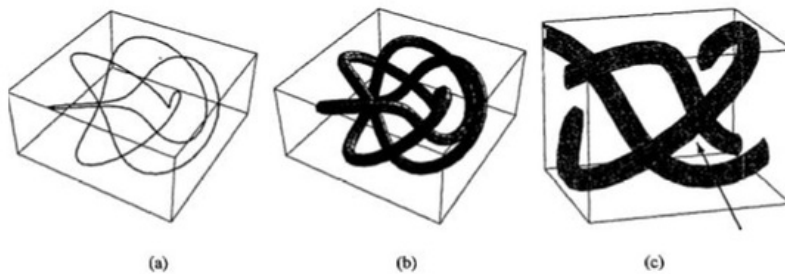
rotations are different from other forms of representing rotations but are used in similar algorithms like splines.

### Information Representation

The presentation of data is often more important than the data itself, the data can be represented without colour and presented with data which does not contribute to the point that is being put across.



Figures (1) [1]

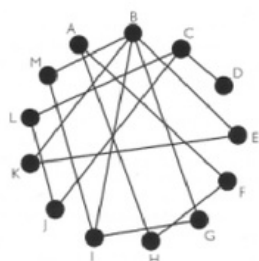


Figures(2) [1]

The pictures above are the same picture where one is from the paper as it normally is and the other is a black and white copy. The ability of the reader to understand what the paper is trying to put across is largely lost in the black and white copy due to the black and white

object information does not show the lines representing the faces of the object, the loss of the face data of the object being showed is the worst loss of data because the information telling the reader where that individual part of the object is going is lost, which means when the object crosses over itself it is almost impossible for the reader to tell how the shape actually looks.

In information visualization chapter 4 representation section called colour puts some great reasoning of when to use colour and that the decision of colour has to be chosen not just scaled from colour red to pink divided by the amount of the colour because the person looking at that visualization will not be able to tell the difference between the objects because the strength of the colour is not different enough to be able to see the individual objects as separate elements and that if elements are interacting with each other as in the in workings of a clock, the person would see different gears and leavers interacting but is each gear and lever had different colour so that if you looked at a certain location within the clock the person could understand which gear it is because of the location of the colour.



Figure(3) [2]

There is also a point of too much information to a scenario as figure 3 shows a set of inter connected node some send others receive aid during an emergency but not all of the nodes connect to each other in that is the problem with the diagram with first glance look at the

diagram the reader would think that the nodes are all linked but the reader should not, in the book information visualization the section shows that this diagram and others[2].

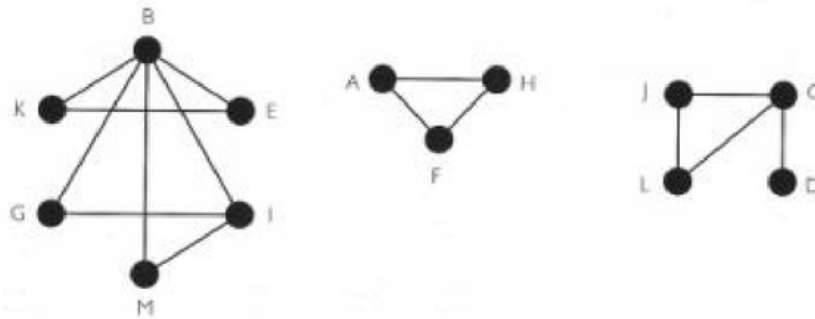


Figure (4) [2]

The other diagrams as shown in figure 4 are easier to understand and show that there is no connection between the separate nodes, making any information to do with the separate section of nodes do not affect each other. The point of the diagrams is to point out that there is no chance of supplies from each diagram arriving at the other diagrams nodes while figure 3 diagram which would take a time to find out when looking at it.

## Projection

When most computer programs are made the understanding of the scene that wants to be shown can be 2 dimensional for data like graphs which show one set of data interacts and affects the other set of data but when it comes to 3 dimensional data there is normal extra data in colour being affected due to its position to a light source. in information visualization chapter on dimensional data the 2 dimensional data is separated through colour but the 3 dimensional data is seen to have the same shadowing being done to each sphere but the shadow size on the sphere which gives across the information that a person would see in their

daily lives and can understand the proximity to the object even though the distance is almost equal for each representation of the shape to the person viewing the object. [2]The most important data that is made through projection is the way the shapes are changed depending on location to the camera and the screen goes from detailing the objects as if the objects are all the same distance away to varying distances, the alteration to the shape is what creates as sense of depth in a 2 dimensional environment.

## Orientation

Orientation and rotation are ways to define between the action and the result of the action, where by when you rotate you change from orientation to another orientation and rotation is the only way to change the orientation of the object. [3]

## Euler Angles

Euler angles have set of three rotations that will be acted upon at least 2 axis of x y and z and are stored in triplet, for each individual rotation it will only be done around only one axis and is simple represented as a roll around that individual axis at that time. Euler angles do suffer from a problem known as gimbal lock whereby through rotations 2 axis line up which means one degree of freedom is lost and that 2 axis rotations will cause the same affect, meaning that to rotate in 3 dimensions 1 of the set rotations in the will have to affect multiple axis to cause the rotation. [4]

## Quaternion

Quaternions are viewed in many ways, the least likely one to visualize is the four dimensional vector space of  $\mathbb{R}^4$  and most software which states that it is showing the fourth dimensional aspect of quaternions are normally showing a unit quaternion affecting an object or a unit quaternion with relation to a unit sphere.

The discovery of quaternions was done by a man named Sir William Rowan Hamilton who was at the time trying to extend complex numbers from:

$$a + bi$$

to

$$a+bi+cj$$

As in the complex numbers the character where  $i$  is the imaginary part so is  $j$  and it would also equal square root of minus 1, this part of complex numbers is known as the imaginary because square root of any number is positive since the number times itself is positive and so no square rooted number can ever exist normally.

Hamilton realised one day when walking day on the his way to Royal Irish Academy, that the imaginary part of the equation was not

$$Bi+cj$$

But actually

$$Bi+cj+dk$$

There was also a cycle pattern between each imaginary part, that is seen as  $i*j = k$  ,  $j*k = i$  and  $k*i = j$  but reversed cycle pattern  $k*j = -i$  ,  $j*i = -k$  and  $i*k = -j$ .

Quaternions have 2 common representations which are

$$Q = a + bi + cj + dk$$

And also

$$(S, \mathbf{v})$$

S is the real number and a scalar value and v is the imaginary part stored in a 3 dimensional vector form (i,j,k).

Quaternion does also have different ways of calculating

Multiplication

$$q_1 q_2 = (s_1 s_2 - \mathbf{v}_1 \bullet \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2)$$

This is the shorter version of quaternion multiplication and better shown through expanding it down to individual calculations

$$\begin{aligned} Q_1 Q_2 = & ((w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + \\ & (y_1 z_2 - y_2 z_1 + w_1 x_2 + w_2 x_1) i + \\ & (x_2 z_1 - x_1 z_2 + w_1 y_2 + w_2 y_1) j + \\ & (x_1 y_2 - x_2 y_1 + w_1 z_2 + w_2 z_1) k) \end{aligned}$$

The only difference in the representation is that the scalar value is stored in the w instead of s.

Conjugate

$$Q^* = w - \mathbf{v}$$

Magnitude

$$\begin{aligned}|Q|^2 &= QQ^* \\ &= Q^*Q \\ &= w^2 + \mathbf{v} \cdot \mathbf{v}\end{aligned}$$

The magnitude is one of the ways to get the length of a quaternion.

Division normally called Inverse

$$Q^{-1} = \frac{1}{|Q|}Q^*$$

### Unit quaternion

Unit quaternion is like normalized vector and is calculated in a similar manner since the same steps are followed. Times each part of the equation by itself and add them all together and you get the length of vector or quaternion now if you divide each part of the quaternion by the length you will get the unit quaternion which is the same as for vectors.

### Unit sphere

Unit sphere is representational of 3 Dimensional surface this because the unit sphere does not represent a sphere in 3 Dimensional space but the surface of a  $R^4$  hypersphere. This is useful for showing a unit quaternion since the radius of unit sphere is length of 1 and a unit quaternion is a quaternion of length 1 and that it can never exceed the boundaries of the unit sphere.

## Rotations Calculations

Rotations calculation and by extension quaternion calculations have a well-known operation difference when compared to normal mathematical calculations since they are non-commutative which means that quaternion  $q_1$  times  $q_2$  does not equal  $q_2$  times  $q_1$ .

This is important when it comes to quaternions and is something that causes complete different result than what is intended.

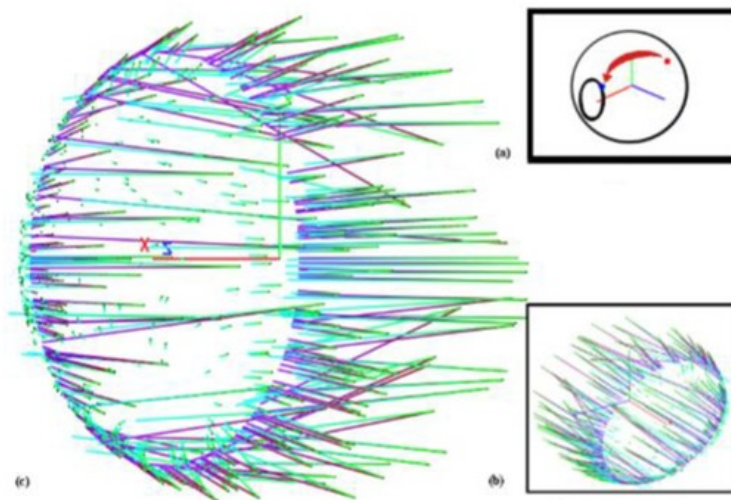
## Fixed and Moving Rotations

There are 2 ways to apply rotations to an object in graphics scene, fixed is when rotations are done around the origin of the world and moving is when the objects origin is rotated and doesn't affect the other objects in the scene.

## Neural Networks

The neural network that use quaternions is based around joint restrictions where by a set of quaternions are inputted into individual nodes and then the nodes go through training that is represented as a quaternion and then output quaternion is compared against the training quaternion to see if the input quaternions have been changed correctly[4].

The representation of the quaternion data in Evolved Neural Network Approximation of discontinuous vector fields in unit quaternions space ( $S^3$ ) for anatomical joint constraint by Glenn Llewellyn Jenkins



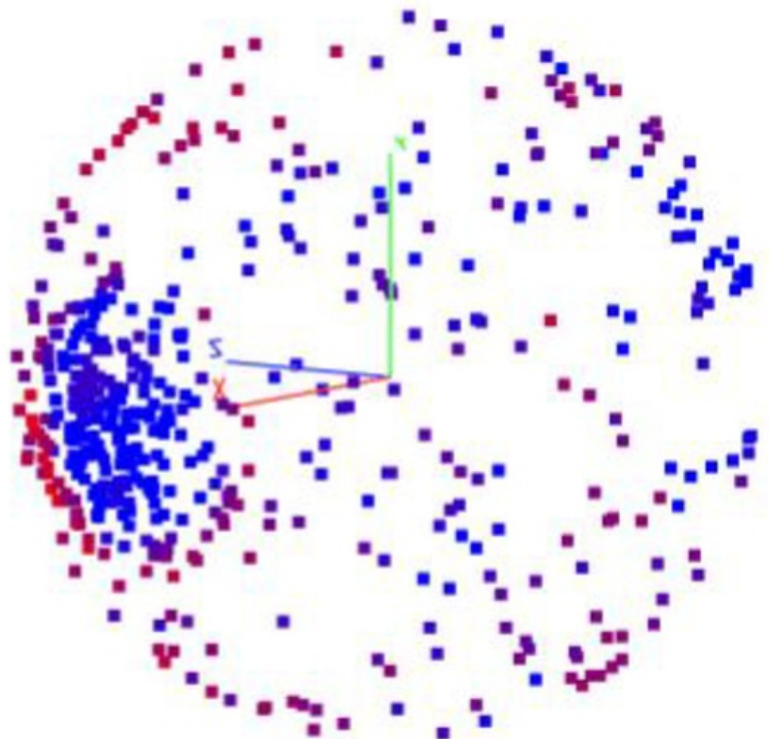
[4]this picture represents quaternions before and after being trained by a neural network. It shows the quaternion points being brought close or completely into the unit sphere.

The picture given with context makes sense and shows that yes the neural network does bring most of the quaternions closer to the unit sphere but the element of the unit sphere the 3 Dimension data is hard to understand and not obvious to the reader, and this is mainly because there is no way to tell if the object is 3 except for the axis of the centre of the sphere which is partly obscured by the vector lines that show the change of quaternion position.

The diagram also does not show the scalar part of quaternion just the vector data as the points and also there is way to identify that the points that are be changed 3 dimensions.

Another figure from the same work shows the clustering of the of the quaternions also within a unit sphere but has a problem could be avoided by doing what the first figure did and show another perspective because this would show that the points are actually clumped together also what could of showed the grouping to be more easily believed is by colouring the points on their distance away from the centre of the unit sphere itself this would have provided the

viewer of the information context to the relationship between the points and the centre of the sphere.



Another problem with the second picture is that even though the viewer knows the data is given colour based upon its chance for being wrong, there is no way to tell the actual location of the square point. This means for the work that is being done the person trying to understand whether the chance for error has relation depending on certain layers of the sphere, or in a certain cone area of the sphere[4].

### Character Motion

In the dissertation Exploiting Quaternions to Support Expressive Interactive Character Motion by Michael Patrick Johnson the paper is about character motion and looks into creating an

interactive engine that will have a middle ground between inverse kinematics wide range of interaction and quality of animated or motion capture quality of realism being the lack of jerky actions which are known in inverse kinematics. The engine that is put forward use of quaternions to store the rotational data because of its computational capabilities.

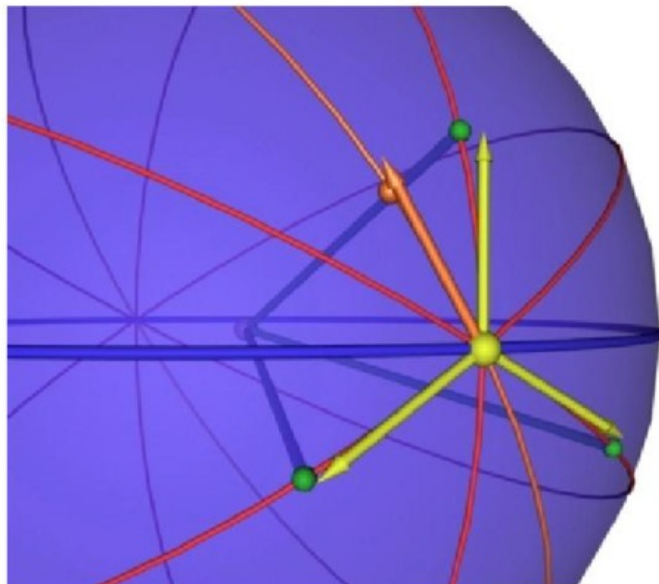


Figure 7-3: The slime algorithm linearly blends the tangent vector description (yellow vectors) of the geodesics (red great circles) of the examples (green spheres). As an example, the orange vector is an arbitrary weighted blend of the three example vectors. This blend is actually specifying a particular different geodesic through the reference point — the orange great circle. By integrating this blended angular velocity forward unit time from the reference (using the exponential), we get the blended quaternion (orange sphere).

The images above with its description give a great deal of information firstly the description tells you what each line and object represent. This is also improved by how the lines that rap around the sphere change colour once they are obscured by the sphere itself but are still visible because the sphere is transparent would be similar to looking through stained glass. With the glass effect it allows the viewer to understand the quaternion curve going around the sphere even though the red and yellow lines look the same when looking through the sphere itself the way the way the lines meet allows the viewer of the information to determine which

line is showing the quaternion curve. From this clever use of visualizing 3 dimensions there is no loss in understanding with what is being shown in comparison to the neural network figures.

### Curves and splines

Splines and curves that use unit quaternions where by the control points for the splines are placed upon a sphere which in some papers [5] where the sphere is in  $r^3$  and the quaternions are used as control points for splines like Bezier and b spline in which they interpolate between the quaternion points on the 2d surface of the sphere. Interpolation across a curve has a problem of angular acceleration which causes the amount of points towards the centre of the curve to be less because of the acceleration [6]. Curves are also used for animating objects and in where by the algorithm known as Slerp is used to change the scene which is very similar to splines that use control points but with Slerp there is only two points to interpolate in-between because Slerp is based on linear interpolation and forces the lines from the points to the surface of the sphere but this would also suffer from acceleration problems because the amount the points are changed is greater depending on how far the points are away from the start and end points which are on the sphere.[7]

Even though in the paper Fast Construction of Accurate Quaternion Splines uses quaternions instead of unit quaternions, the quaternions could be normalised so the rotation can be compared with no magnitude. Also that rotational data could be compared to curves which are made from quaternion Slerp and quaternion curves.

### Aircraft Control

Most uses of quaternions in aircraft control papers, use dual unit quaternions which can be understood as one unit quaternion represents the rotation of the aircraft to some normal

orientation and another unit quaternion controls the positioning of the aircraft itself[8][9][10][11].

The representation of the aircrafts in flight is seen as a spiral but normally has not axis showing how the aircraft changes as it goes along the path being shown in a 3 dimension view . The 3 dimensional visualization of the aircraft is shown be having axis surround the spiral to give context to the aircrafts movement in the 3 dimensional space.

One paper [11]show how a set of quadrotors used dual quaternions to store position and rotation like the other papers to do with aircraft but also pass the information to each quadrotor which then changes its own rotation and position to keep with the quadrotor and the layout of the quadrotor is shown as a tree of nodes and resembles an aircraft flying formation.

Dual unit quaternions do have a different way of calculating each other and are not done independently of each other meaning the calculations for dual unit quaternions are calculated as eight numbers times eight numbers but are quaternion normalized so should be able to be seen in a unit sphere scene and should be accurate representations.

## Robotic Arms

In robotics the joints of the machine will wear down over time due to the various rotations that being done. In the paper [12], the use of Slerp to calculate a smoother rotational interpolation between the two orientations and was found to be a better interpolation then unit quaternion interpolation in that the unit quaternion interpolation was found not smooth in its acceleration and velocity.

## Conclusion

The research papers have shown that there are some credible ways to show unit quaternion on a unit sphere and also ways not, as seen in the neural network paper where by the lack of the third dimension does not stop what is meant to be portrayed from being understood but the use of colour and depth techniques which were partially seen in the character motion would of allowed the understanding of the points within the unit sphere which would have given greater context to each of the points.

Almost all of the papers where the unit quaternion is being represented with a unit sphere it is normally on the surface of the sphere in a 2 dimensional sense and is a valid way to show the spherical lines and splines but also shows the 3 dimensional rotation because of the position of the point on the sphere.

## Methodology

Since the project is about creating a programme to visualize unit quaternion and evaluating the programme on whether it has succeed or not depending on the user requirements of the programme from that reasoning the philosophy of the project is positivism and an approach of deductive.

A problem with the programme being deduced whether it is correct by the creator means that the view can be seen as biased, this should be removed by creating the requirements for the project with the supervisor. The requirements for the system should be concise and easy to

judge and if not broke down into individual requirements so that requirement can be judge correctly.

The project started out as an experimental research strategy but by reading literature related to quaternions, unit quaternions and generally visualizing graphics it was decided to change the research to development and create a programme that is intended to visualize unit quaternions in multiple ways to make it easy to understand unit quaternions in relation to unit sphere.

Then versatility of agile methods in design and implementation of software allows for some experimental development at the start of the design and implementation phase of programming which means that any ideas to visualizing unit quaternions can be tried and evaluated without taking time away from developing the software.

## **Design & Implantation**

The development process that was used was the iterative method of ropes which is accurately shown in figure(5) . It starts on the right hand side where the requirement and system analysis done on the user and system requirements which have already been done before the design starts. From the requirements the design is created and then implemented and tested to show that it works on its own.

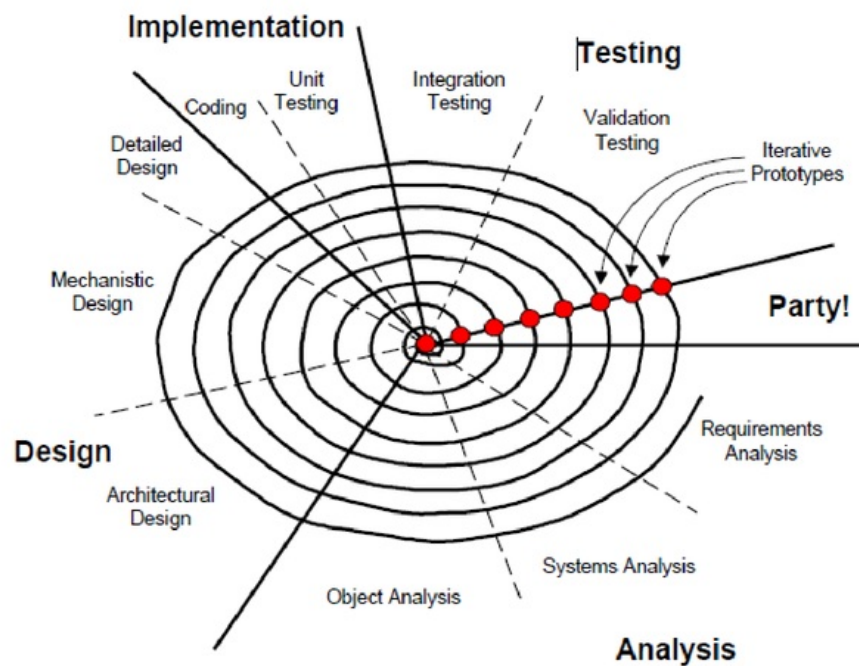


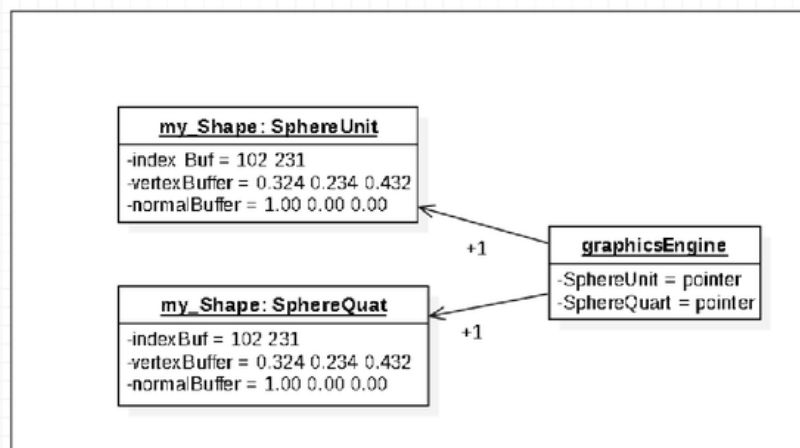
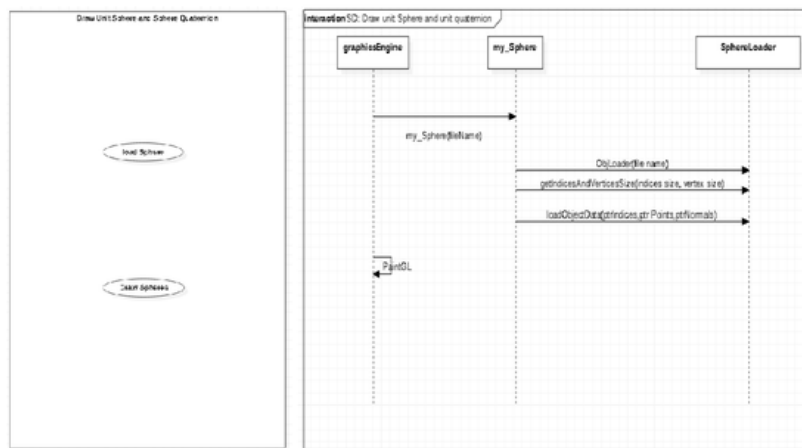
Figure (5) [13] Shows ropes method of development.

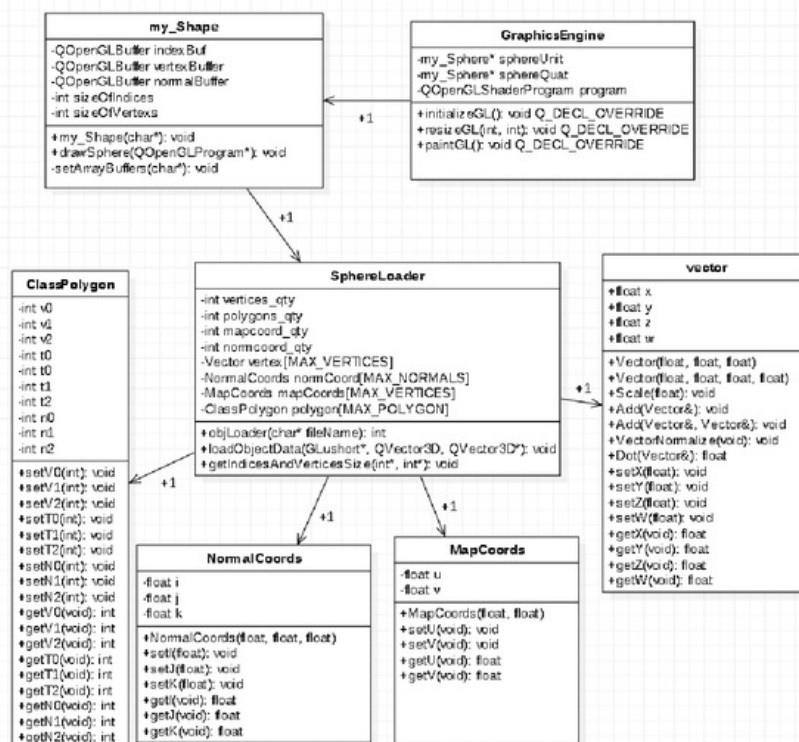
Before the iterations of the design and implementation, there was an overall design which was altered throughout the project as new problems were raised during the iterations. The overall design was created from the requirements in appendix A.

#### First iteration

The first iteration of the implementation was to draw a unit sphere with a single unit quaternion within or on the 2 dimensional surfaces. This iteration was completed within around half a day of coding and testing. The reasoning behind doing this part of the design first is because it can be done separate from other parts of the design like file loading and can be added to in later iterations.

The behaviour that generated this iteration is shown in figures (3)





Figures(6)

This class diagram shows that the file information is stored in classes of vector, classpolygon, normalCoords and mapCoords which store the obj file information. It could have been stored as local arrays but the classes were taken from previous work and had been tested already.

### Second iteration

The second iteration of the implementation was to load a list of quaternion that was stored in a quaternion file into a new independent build. This was completed and tested to show that the list of quaternions stored in pat and csv files were loaded and checked to be within

accuracy. This part of the design is similar to the first where by it could be done separately from the other sections of the design.

### Third iteration

The third iteration added axis lines to drawing the spheres and worked relatively well but had problems of understanding QOpenGLShaderProgram which allows the adding of vertex and fragment shaders but does not seem to have the ability to switch between vertex and fragment shaders pair. To solve this problem graphics engine class was implemented with 2 program shaders and has to be bind before drawing lines and sphere object. The reasoning behind adding the axis to the sphere drawing program before putting the quaternion list and sphere parts together is that it was thought to cause less problems then the fourth iteration which did bring the two parts together.

### Fourth iteration

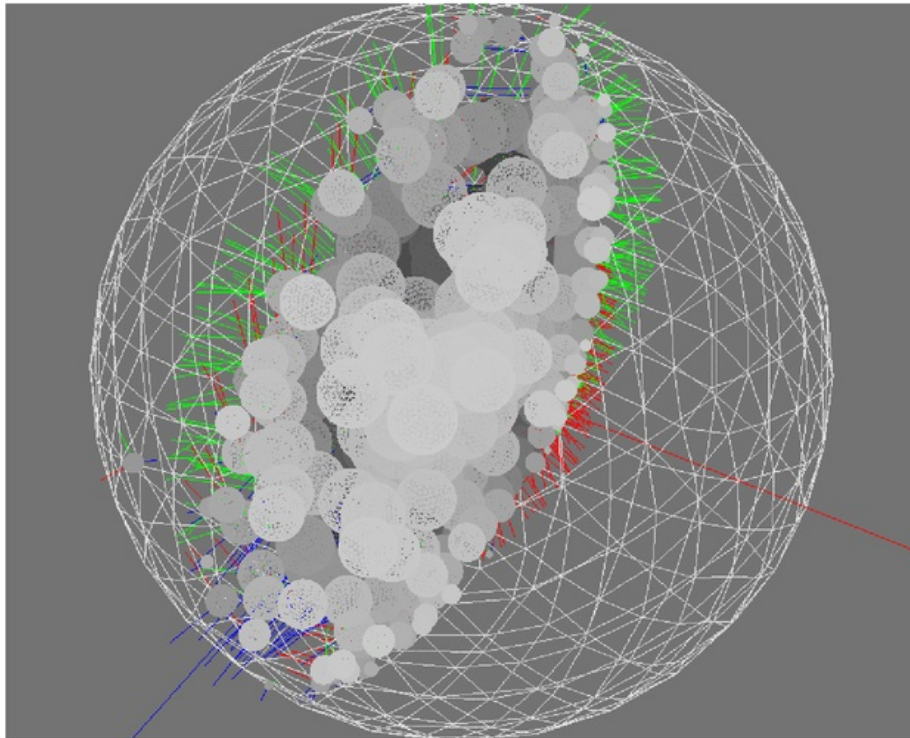
The fourth iteration added the second iteration to the third and also used the quaternion location to determine the location of the unit quaternions on the sphere. This iteration worked correctly and displayed all the orthogonal axes on all the spheres correctly.

### Fifth Iteration

The fifth iteration added the functionality of rotating the unit sphere using mouse commands. This iteration encountered a problem where the rotation of the unit sphere caused jittering of the objects in the scene. To solve this problem instancing was introduced to the program to reduce the amount of draw calls which then intern reduces and stops the jittering affect. Instancing is where by the data stored in openglbuffer , the stored information is passed through to vertex shader and is similar to creating attribute varaibles but except for the use of the command:

```
glVertexAttribDivisor( {attributeLocation} , 1 );
```

The command tells the pipeline that the attribute at the attribute location is instanced and the second parameter states that the data will be read every so many instances. This also has to be used with draw calls that are instanced where by the last command states how many instances there are. The amount of instancing passed through the in the draw command must be less or equal to the amount of instanced data.



Figure(7)

### Sixth iteration

The sixth iteration added the keyboard commands and also the file loader. The file loader is a Qt example of file loading and was altered to only display the text of valid file format and also the text was docked to the right hand side of the screen and the opengl widget docked to the left hand side. One problem caused in this iteration is that there is a window formed over the

menu in the top left which is how the user loads in new quaternion files. The keyboard commands allowed the user to change the representation of the unit quaternion with reference to the unit sphere. The colours chose for the sphere is basically using numerical value for each one determines its colour on the white to black scale. It was originally meant to do a simple shadow effect with front one colour and the back another colour to reinforce the 3 dimensional natures of the points.

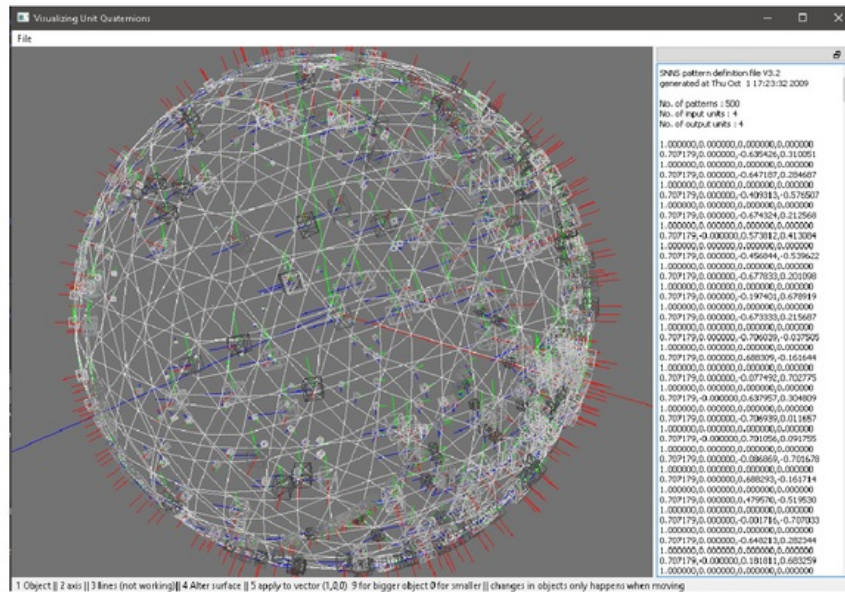


Figure (8)

### Seventh iteration

The seventh iteration was adding the drawing of lines to the quaternions from the centre of the unit of the unit sphere. This should have been done before iteration five but was forgotten about. This was not achieved due to time constraints and also the lack of understanding the problem since the code made sense.

## Final Interface

Figure() shows the final interface it allows the user to select any PAT or CSV file to be loaded using the open option menu that is located in a normal location but it does not normally load a window over it for no apparent reason. This is an annoyance to the user but is easily rectified by the closing or moving the window. The displaying of the file next to the visualization gives good context to the quaternions stored in the file but does not fully clarify which quaternion is represented by those numbers it would be useful and possibly future work.

## Evaluation

To state whether the program actually proves a better visualization of unit quaternions in relation to a unit sphere in comparison to the research would be a bad way to think of determining the success. A more relevant question to this project would be if it would be able to visualize the unit quaternion data that is seen in the literature review and the answer from using the requirements would be partially only because the lines from the unit quaternions to the centre of the unit sphere were not drawn correctly and if they were drawn there would be not major failing points of the program. The figures in this report on the program itself also point out how difficult it is to show 3 dimensional objects with transparency and still portray a 3 dimensional environment.

## Conclusion & Recommendations

The project has analysed different ways to show unit quaternions in relation to unit sphere and compiled some interesting ways to show rotational information of unit quaternions to the

screen. The design of the project was not of a high quality and is a detrimental effect on the report and even though it is explained it should have had greater visual aid to understand the iterations of the program. The program was implemented as intended and even though the loss of one of the major requirements it still did show a representation of unit quaternions.

There are eleven ways that the project aimed to do but did not achieve, this is mainly because of time restraints or problems with the implementation. The draw line to unit quaternions has already been explained that it was coded but the reason for not drawing is unknown. The unknown problem should be the first goal to solve in future work. There are also the other less important requirements that could be added to the program. The scalar being represented by colour does not seem a complex problem to solve and if passed into the vertex shader it could be handled by just passing the size data in which would most likely be instanced since it would be per sphere data. The twist of the unit quaternion could be shown as a colour that represents the axis becoming more predominant depending on the angle change when compared to a starting rotation of nothing. Quaternion splines being shown on the unit surface and a 2 Dimensional surface would be interesting since this project has no interpolation between two points and curves which hold similarities. Blurring between two quaternions points would be an interesting representation of unit quaternion rotations. If the current implementation and spline with interpolation was implemented it would be interesting how the different ways of representing the unit quaternion rotations vary and whether there is any correlation between them. The last few requirements are to do with either creating a unit quaternion or quaternion calculations which would show more of the unique calculations that are seen in quaternion mathematical corner.

## Reflection

Through my discovery of unit quaternions and quaternions literature I have discovered an interesting field of representing rotations that I previously never knew existed and have learned to respect them for their great way of showing rotations. I have also learned some of Qt and discovered its great library of commands are very easy to understand because of simple searches of the Qt variables we normally lead to well documented descriptions of the commands for the said variables and they commonly have a good context example to go with it. The problem that was in counted with calling to many draw calls will give me cautions to use the OpenGL draw commands without consulting can it be done more efficiently. I do believe that for any programs that I make in the future or have input on I will push for the use of quaternions to be used instead of Euler just because the amount of times I read that quaternion based systems run faster. I do believe I have failed myself in time management and with only few more weeks spent on the project I could have completed all my requirements. I do hope I pay more respects to topics like quaternions where most people reading and using them normally give them a waning and that next time I see that in a field of research I hope I concentrate on what the people in the papers or verbal are trying to get across.

## References

- 1.) Handon,A,j. In *Proceedings of the conference on Visualization '98 (VIS '98).Constrained Optimal Framings of Curves and Surfaces using Quaternion Gauss Maps,,1998,375-382*
- 2.) R,Spence. *Information Visualization, 1<sup>st</sup>,Addison wesley,2001*
- 3.) Johnson.M.P, Exploiting Quaternions to Support Expressive Interactive Character
- 4.) Jenkins,G,L. EVOLVED NEURAL NETWORK APPROXIMATION OF DISCONTINUOUS VECTOR FIELDS IN UNIT QUATERNION SPACE (S3) FOR ANATOMICAL JOINT CONSTRAINT,2007
- 5.) Kim.M,Kim.M,Shin.S,Y, SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, *A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives,1995,369,376*
- 6.) Ramamoorthi,R. Barr,A,H. Proceedings of the 24th annual conference on Computer graphics and interactive techniques. *Fast Construction of Accurate Quaternion Spline.1993,287-292*
- 7.) Handon,A,j. In *Proceedings of the conference on Visualization '98 (VIS '98).Constrained Optimal Framings of Curves and Surfaces using Quaternion Gauss Maps,,1998,375-382*
- 8.) Isenberg.D.R, 41st Southeastern Symposium on System Theory, *Input and Output Transformations for a Space Robot Modeled with Quaternions.,2009,255-260*
- 9.) Nam.S, Lee.D,Lra.J,Chai.Y. VRST '00 Proceedings of the ACM symposium on Virtual reality software and technology., *Dual Projection-based VR System for the Light weight Motion-based Driving Simulator.2000.,195-198.*

- 10.)Thein.M.L, American Control Conference., *Attitude and Angular Rate Estimation Using the Sliding Mode Observer With Additive Quaternion Corrections.*,2006,3302-3307.
- 11.)Lin.Z,Yu.C,Wnag.X., IEEE Transactions on Robotics., *A Dual Quaternion Solution to Attitude and Position Control for Rigid-Body Coordination.*,2012.1162-1170.
- 12.)Kong.M,Chen.CJi.Z,Li.R., nformation and Automation (ICIA), 2013 IEEE International Conference on., *Application of Orientation Interpolation of Robot Using*,2013,384-389
- 13.)Douglass.B.P, *Rapid Object-Oriented Process for Embedded Systems.0 - 2000*

## Bibliography

- [1] A. J. Hanson, "Constrained Optimal Framings of Curves and Surfaces using Quaternion Gauss Maps."

- [2] R. Spence, *Information Visualization*, 1st ed. Addison Wesley, 2001.
- [3] M. P. Johnson, "Exploiting Quaternions to Support Expressive Interactive Character Motion," 2003.
- [4] G. L. Jenkins, "Evolved neural network approximation of discontinuous vector fields in unit quaternion space (s 3 ) for anatomical joint constraint," no. August, 2007.
- [5] M. Kim, M. Kim, and S. Y. Shin, "A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives 1," pp. 369–376.
- [6] R. Ramamoorthi and A. H. Barr, "Fast Construction of Accurate Quaternion Splines," 1997.
- [7] A. J. Hanson, "Constrained Optimal Framings of Curves and Surfaces using Quaternion Gauss Maps," vol. 98, 1998.
- [8] D. Isenberg, "Input and Output Transformations for a Space Robot Modeled with Quaternions," pp. 255–260, 2009.
- [9] S. Nam, D. Lee, J. Ira, and Y. Chai, ". Dual Projection-based VR System for the Light weight Motion-based Driving Simulator," pp. 195–198.
- [10] M. L. Thein, "Attitude and Angular Rate Estimation Using the Sliding Mode Observer With Additive Quaternion Corrections," pp. 3302–3307, 2006.
- [11] X. Wang, C. Yu, and Z. Lin, "A Dual Quaternion Solution to Attitude and Position Control for Rigid-Body Coordination," vol. 28, no. 5, pp. 1162–1170, 2012.
- [12] U. Quaternion, M. Kong, C. Ji, Z. Chen, and R. Li, "Application of Orientation Interpolation of Robot Using," no. August, pp. 384–

389, 2013.

- [13] B. P. Douglass, "ROPES : Rapid Object-Oriented Process for Embedded Systems," pp. 1–40, 1999.

## Appendix A

### User Requirements

- Be able to run on windows and Linux
- Draws at least on unit quaternion
- Draws the unit sphere
- Shows the scaler value of a unit quaternion
- Show the quaternion splines
- Show quaternion calculations
- Easy to use
- Load in quaternion file format

## System Requirements

- Must be written in Qt
- Must be drawn using OpenGL with its shadier language
- Must draw Unit quaternions
- Must draw Unit Sphere
- Must draw Axis for unit sphere
- Must draw Axis for each quaternion
- Must show the scalar value through at least one interpretation
- Must draw lines from unit sphere origin to all unit quaternions
- Must be able to view unit sphere from multiple orientations
- Should show unit quaternions affecting a single axis
- Should use colour to represent scalar
- Should use size to represent scalar
- Should accurately represent 3 dimensional surface
- Should load in different files that hold quaternion data
- Should show the twist of the unit quaternion rotation
- Could draw quaternion splines
- Could draw blur of two unit quaternion calculation
- Could show the difference between a 2d and a 3d surface
- Could show quaternion multiplication
- Could show conjugate
- Could show Magnitude of a quaternion
- Could allow selection of a unit quaternion for calculations
- Could create a unit quaternion from an Axis angle

- Could create a unit quaternion from an Euler Angle
- Could show the quaternion file that is loaded