

Investigation into the different representations of 4D shapes

Joe Subbiani

2377990S

Project Outline

Motivation

4 Dimensional objects are something that cannot be created physically and cannot be visualised in their true state. The motivation for the project is to further a persons understanding of high dimensional space and to test their understanding of geometry that cannot be wholly visualised.

Aims

This project focuses on the development of a wide range of 4 Dimensional shapes and a variety of ways to display these shapes in a way humans can best interpret them. As such the usability of each representation will be experimentally validated in order to find the most intuitive and effective way to represent higher dimensional shapes.

The user will be asked to identify 4D objects, the rotations they undergo and to manipulate and match the pose of 2 shapes. The accuracy of which the user can identify these properties and act accordingly will be measured against the representation being shown to them.

Representations include:

- A control (standard 3D cross section of a 4D shape)
- A multi-rotational view of the object to see how it looks when viewing it from all 3 4D perpendicular angles of rotation
- A W-axis timeline to show how the objects cross-section changes moving forwards and backwards across the 4th dimension.
- A 3D view port of a 4D object to demonstrate 4D rotation with respect to 3D rotation

Progress

- Created a ray marcher in unity to visualise 4D shapes
- I will be testing users on these shapes:
 - Cube
 - Sphere
 - Cone
 - Tetrahedron
 - Torus
 - Capsule
- Created 5 different representations (including control) - will proceed with 4
 - A control (standard 3D cross section of a 4D shape)
 - A multi-rotational view of the object to see how it looks when viewing it from all 3 4D perpendicular angles of rotation
 - A W-axis timeline to show how the objects cross-section changes moving forwards and backwards across the 4th dimension.
 - A 3D view port of a 4D object to demonstrate 4D rotation with respect to 3D rotation
 - (NOT INCLUDED) 4D onion skin
- Used a variation on tri-planar projection where I project a pattern across each axis onto the object to distinguish faces
 - RGBW (XYZW colouration in the positive direction is coloured with Red, Green, Blue and White respectively)
 - RGBW with pattern

- Blue+ Red- with pattern (Blue in the positive direction and red in the negative direction along the w axis)
- Created a modular scene to enable and disable UI components and switch representations for the experiment
- I can read and write data to a persistent location
- Implemented rotor rotation, however it can be quite glitchy on hard surfaces
- Rotation can be controlled through 2 input methods
 - grab ball: click along each arc to rotate along the objects axis.
 - swipe rotation: swipe in the direction to rotate the object along the global axis.

Problems and Risks

Problems

- **Deriving the signed distance functions for a hyper cone and a pentachoron (4D tetrahedron).**
 - Existing SDFs for a 3D cone where not really extendable to 4D.
 - Had to go through multiple ideas and attempts before successfully deriving a 4D tetrahedron.
 - A pentachoron was necessary as currently the cube was the only hard edged object making it very obvious to identify against the other curved objects.
 - This did not hold up the project too much.
- **Onion skin effect was not very intuitive.**
 - The onion skin shader was a plan to show overlaying translucent cross sections, forward and backward, along the W axis to show how the shape would change if you travelled forward and backward along the W axis.
 - To make it more intuitive I made the forward and backward cross sections only show incremented steps, rather than a consistent offset such that it is easier to watch the shape become the future and past cross sections as you move along the W axis.
 - Appeared to be a waste of effort, but was an interesting exploration nonetheless.
- **Implementing intuitive rotation with a grab ball.**
 - Having the grab ball rotate in a way that matches the mouse gesture proved quite challenging. Even in well established engines such as Unity it is not very clear what will happen.
 - The grab ball rotates when the mouse is dragged along the tangent of the grab ball ring. This was still not immediately obvious as it feels like the mouse should follow the arc of the ball.
 - Implemented a tangent line to show up when an arc is clicked to show the user what direction to move the mouse to rotate the object along the axis.
 - This had a moderate impact on progress, but was not too bad.
- **Implementing 4D rotation.**
 - Euler matrices are unstable and have gimbal lock problems in 3D. This is exponentially worse in 4D.
 - Explored Marc ten Bosch's solution - Geometric Algebra Rotors.
 - Multiple derivations with slight tweaks over weeks that were not successful. Massive delay in progress.
 - Used `sympy` and `galgebra` to produce a closed form expression for rotation
 - The working 4D rotation produces some very glitchy visualisations within the shader that need reviewing.

Risks

- **Developing stable, *visually clear*, rotation**
 - Currently the object has unexpected visual glitches when rotating the object in the 4th dimension.
 - I will mitigate the impact by designing a “skip” button for the pose matching test. If the user cannot see a shape that is not impaired, they should skip the tests.
 - Given time I will review and experiment with the rotor equations, and see if I can produce more stable results.

- **May not get enough data from experiments**
 - There are many ways to represent the objects, from how it is layed out, to the texture applied to it.
 - To mitigate too much data requiring more participants, each element of the experiment will be ordered in a way to prioritise different data.
 - Method of rotation will be something that will be specific to each participant, so they can get familiar with manipulating objects
 - Each set of tests will be conducted per representation - how the object is displayed. This is the most important piece of data - which representation is best
 - The texture applied to the object will be random for each test. I believe this will effect the results the least, and therefore is not being segregated for further research. The texture is more of a guide to the user to provide a point of reference as they handle the object.

Plan

Christmas

- Record tutorial explanation videos
- Continue to develop and implement the rest of the test system
 - Build survey scene
 - Store current experiment status and load it between surveys
 - Store answers of survey
 - Implement random rotation mechanic in the modular scene

Semester 2

Week 1 & Week 2

- Finish implementing test system

Week 3

- Begin developing data analysis tools, most likely with a jupyter notebook
- run some preliminary tests to decide if there is any data or visualisations I want to cut

Week 4 & Week 5 & Week 6

- Run experiments
- Begin dissertation
- Further develop data analysis tools

Week 7

- Evaluation of experiments

Week 8 & Week 9 & Week 10

- Write up draft dissertation