

Developing a Morphing Algorithm That Can Transform One 3D Object Into Another

Conclusion

The final algorithm that has been implemented is able to transform one object into another in both 2D and 3D. Currently, the algorithm is only able to use a limited number of 3D objects, however the morphing sequence is smooth and visually pleasing.

Future Work

In future, the algorithm will be improved in order to allow for any object to be morphed. Also the algorithm is going to be implemented into a virtual reality environment, which was the original goal of this project.

Introduction

The proposed algorithm in this project has been influenced by existing algorithms. The algorithm which was proposed by Kent et al explains their method of correspondence as inflating two objects like balloons until they become spherical. So that each point of the object is mapped on to a specific point on the surface of the sphere. This creates a one-to-one correspondence between the objects [1]. This seemed to be an effective method to begin correspondence and as such, heavily influenced the design of this algorithm.

Aim

To develop a working 3D morphing algorithm that can transform any simple convex 3D object into another.

Objectives

- Create a correspondence between two 2D objects.
- Interpolate between the two 2D objects.
- Create a correspondence between two 3D objects.
- Interpolate between the two 3D objects.
- Render objects using OpenGL.

What Is Morphing?

Morphing is a type of key-frame interpolation that transforms one object into another. This can be done in both 2D and 3D. However a 2D implementation is generally used on images, rather than objects. 3D morphing can be used for video games, video entertainment, and even educational purposes. Many different algorithms have been proposed to morph 3D objects. Having said that, most of them are computationally expensive, not aesthetically pleasing, or are limited to the type of object that can be morphed. The proposed method in this project is intended to be the first step towards creating a visually pleasing algorithm which is inexpensive to run and can morph a wider range of objects.



Different stages of morphing using algorithm by Kent et al [1]

Implementation

- Two chosen objects are translated so that the centre is at the origin and scaled to unit length.
- Objects are combined into a single array to create a super mesh.
- Super mesh is duplicated and both new objects are reshaped to look like original objects.
- A link is made between each object for every point on both objects.
- Every point is interpolated with the point it was linked with to create in-between frames.
- In-between frames are rendered one at a time to create an animation.
- Rendered objects using OpenGL and added user controls for ease of use.

Results

The algorithm produces a successful morph sequence which runs smoothly and is visually pleasing.

References

[1] - J. R. Kent, W. E. Carson, and R. E. Parent, "Shape Transformation for Polyhedral Objects," 1992.

