

Deformable Objects Using Fast Lattice Shape Matching

Marwan Kallal

1 Introduction

I attempted to use the paper by Alec R. Rivers and Doug L. James, explaining Fast Lattice Shape Matching (FLSM), to make our already fuzzy bunny deformable as well. I have omitted collision and other physics in the interest of time and am focusing on the deformations as single points move. I will also be using the unoptimized version SLSM (Slow LSM), to focus on the way that deformations happen.

2 Explanation of FLSM

2.1 Constructing the Lattice

The first step to FLSM is to build a lattice, or grid, that encloses the mesh. We start by creating a bounding box around the mesh. From here we round the dimensions of the bounding box up to fit an integral number of grid squares. From here we need to check which grid squares are actually in contact with the inside of the mesh. We can do this using a triangle mesh voxelization algorithm [Rosenburg()].

2.2 Creating Particles

particles happen

2.2.1 Assigning Vertices

give particles vertices

2.3 Creating Regions

make regions

2.4 Dynamics

stuff moves

2.4.1 Particle Movement

particles move

2.4.2 Region Movement

regions move

2.4.3 Setting Particle Positions

you put them there

3 My Implementation

```
int i = 0;
```

4 Challenges

doesn't work

5 Optimizations

make faster

References

[Rosenburg()] David Rosenberg. Triangle mesh voxelization. URL <http://blog.wolfire.com/2009/11/Triangle-mesh-voxelization>.