# Mesh to Voxel Transformations for Optimised Physics-Based Interactions

T. Lefley F. Ponjou-Tasse

Project Presentation, 2015

# Motivation

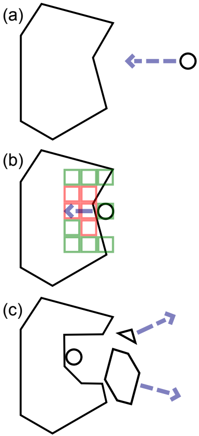

Figure: A simple collision and its volumetric resolution.

- Meshes only store surface information
- Reasoning on interior difficult
- Most destruction algorithms limited
    - Only work on convex shapes
    - Or have to split concave shapes first
- By precomputing volume data we have an O(1) check for inside/outside a shape

# Voxelisation

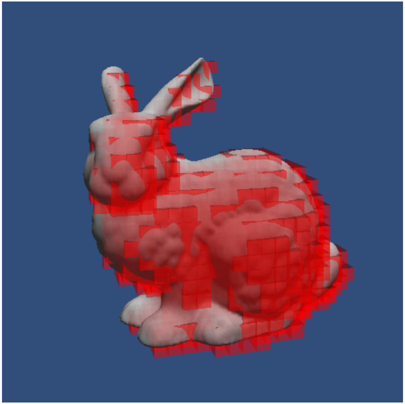

Figure: Voxelisation of the 'Stanford Bunny' model, composed of 69,666 triangles.

- HLSL GPU algorithm
- Based on GPU Pro 3 implementation of Schwarz and Seidel's method
- Achieves solid voxelisation of convex and concave bodies

# Destruction

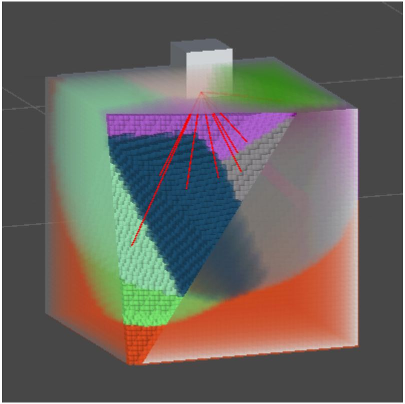

Figure: Labelling of voxels to their correct fragments.

- Process collision information
  - Find collision point
  - Calculate force
  - Generate fragment points
- Construct 3D voronoi diagram
  - Label each voxel by nearest point
  - Within radius
- Find islands
  - Flood fill

# Rebuilding the Mesh

## Different Approaches

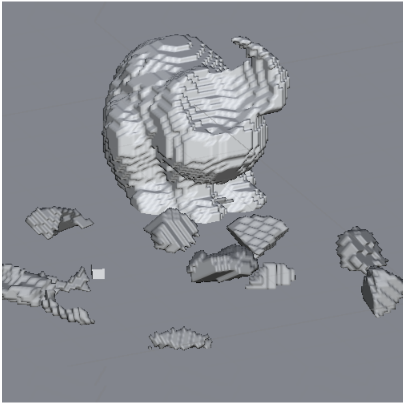

Figure: Fractured Stanford Bunny remeshed using the Marching Tetrahedrons algorithm.

- Marching Tetrahedra
    - Original mesh not preserved
    - Detail loss

# Rebuilding the Mesh

## Different Approaches

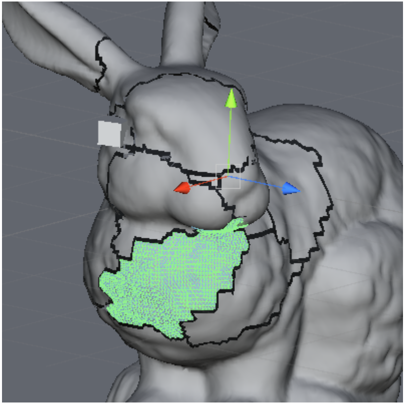

Figure: Fractured Stanford Bunny mesh partitioned using nearest neighbour.

- Marching Tetrahedra
  - Original mesh not preserved
  - Detail loss
- Nearest Neighbour
  - Find only fragment edge voxels
  - Add them to KDTree
  - For all vertices, map to nearest voxel

# Rebuilding the Mesh

## Different Approaches

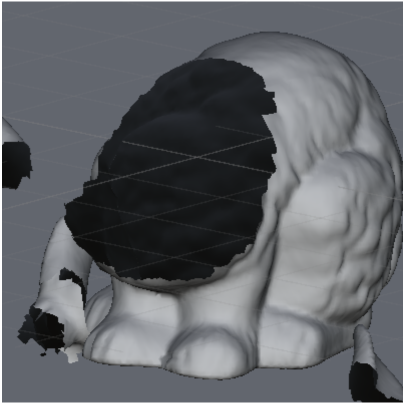

Figure: Fractured hollow Stanford Bunny.

- Marching Tetrahedra
  - Original mesh not preserved
  - Detail loss
- Nearest Neighbour
  - Find only fragment edge voxels
  - Add them to KDTree
  - For all vertices, map to nearest voxel
- Meshing interior...

# The Next Steps

- Finish interior meshing algorithm
- Optimise for speed
  - Multithreading
  - Remove unnecessary complexity
- Post destruction calculations
  - Fragment mass
- Evaluation
  - Framerate
  - Memory usage
  - Physical accuracy
- Write-up