Houdini 13.0 Finite Elements

deforming and fracturing solid objects

Michiel Hagedoorn

SIDE EFFECTS SOFTWARE

Overview

- Introduction
- Getting Started
- Creating Tet Meshes
- Embedding High-Resolution Geometry
- Understanding Internal Forces

Overview

- Fracturing Solid Objects
- Animating Rest and Target States
- Applying Constraints
- Controlling Collisions
- Manipulating Attributes

Introduction

- solid object = solid mechanics
- solid object deforms (non-rigid)
- finite elements: simulation method
- tetrahedrons + attributes
- model: elasticity + energy Loss

Specific Features

- support for isotropic + orthotropic (e.g., wood)
- accurate volume preservation
- material space controlled by UVW
- global collision response

Specific Features

- energy preservation
- smooth embedding of tets and polys
- seamless interaction with thin shells (cloth)
- directable using soft and hard constraints

Start by using the shelf tools!

Tips

- avoid really thin of flat shapes
- don't use more tets than necessary
- tets should resemble regular tetrahedra
- use simplified mesh for simulation

Tips:

- Inspect Interior Tets using Clip SOP
- Use the Quality Option on Tetrahedralize SOP
- Use VDB From Polygons & Convert
- Use the Remesh SOP and its Settings
- Use File SOP when Multiple Platforms

Embedding Features:

- embed both polygons and tetrahedrons
- smooth interpolation (not barycentric)
- fracture along with simulated tets
- pieces sticking out is OK
- bounded memory use
- connected components are kept separate

Main Parameters:

- overall stiffness: forces against deformation
- overall damping ratio: forces against deformation rate
- volume mass density: mass per cubic length unit

Relative Parameters:

- deformation types: stretch, shear, and volume
- stretch & shear stiffness: Young's modulus if stretch = shear
- volume stiffness: related to Poisson's ratio
- anisotropic: strength in UVW material directions (e.g., wood)

Resolution Independence

- orientation and size of tets should not matter much
- results depend mostly on shape, not how it's “tetted”
- lores is good prediction of hires

But

- avoid poorly shaped tetrahedrons for solve speed and stability
- differences more pronounced for lores meshes

Fracturing Solid Objects

- fracture threshold: strength, likeliness of fracture
- solid fracture SOP can create parts
- alternative: specify 'fracturepart' attribute directly
- enable fracturing on both solver and object
- embedded geometry breaks along

Tips

- always create unbreakable parts
- increase global substeps to increase quality
- use embedding for extra details
- embedded polygons suffice in unfractured regions
- shape embedded tets by tetrahedralizing between transforms

Animating Rest and Target States

Rest

- determines internal forces
- independent of rotation and translation

Target

- determines constraint forces
- goal per point: rotation and translation matter

Applying Constraints

- constraints use point/vertex attributes: “targetP” and “targetv”
- import target parameter can create these
- global control: target stiffness and damping
- local control: target constraint or attributes
- supported by spring constraint as well
- constraints work with fracturing

SIDE EFFECTS SOFTWARE

Controlling Collisions

- global collision response
- geometry based or volume based
- geometry based: full geometry, swept collisions
- volume based: per point, using signed distance field (SDF)

Attributes read by solver

- many solid object parameters have multiplier attribute
- includes stiffness, damping and mass density
- don't vary these too much
- constraint attributes: “pintoanimation”
- fracture attributes

Tips for learning

- use the shelf tools
- study nodes and settings created by shelf tools
- play with the examples from this presentation
- read the help documentation
- start simple, add complexity gradually
- when in trouble: increase “Min Substeps”

Questions?