GAMES103 Lab 3: Bouncy House

Due Date: 01/03/2021, 11:59PM

In this lab assignment, we will implement the finite element method for elastic body simulation. For simplicity, we will consider the simple St. Venant-Kirchhoff (StVK) model for elasticity. We will use explicit time integration, so the key component is the calculation of every vertex force.

To begin with, please download and import the example package. This package contains the code for a tetrahedral mesh of a house model. It also contains some functions useful for the assignment, such as singular value decomposition of a 3-by-3 matrix. The example script also contains the code for creating a single tetrahedron. You may use that for testing purposes.

1 Basic Tasks

1.a. Basic setup (2 Points) In the `_Update` function, write the simulation of the house as a simple particle system. Every vertex has its own position $\mathbf{x}$ and velocity $\mathbf{v}$, and the velocity is under the influence of gravity. Please also implement frictional contact between every vertex and the floor. Since this project uses a relatively small time step, the `Update` function calls `_Update` ten times. After that, the function contains the code to send vertex positions into the house mesh for display.

1.b. Edge matrices (2 Points). Next, write a `Build_Edge_Matrix` function that returns the edge matrix of a tetrahedron. In the `Start` function, call this function to calculate `inv_Dm`, the inverse of the reference edge matrix for every tetrahedron.

1.b. Elastic Forces (4 Points). Follow the slides to implement the elastic forces, based on the finite element method. Specifically, for every tetrahedron, calculate the deformation gradient $\mathbf{F}$, the Green strain $\mathbf{G} = \frac{1}{2}\left(\mathbf{F}^\top\mathbf{F} - \mathbf{I}\right)$, the second Piola-Kirchhoff stress $\mathbf{S} = 2s_1\mathbf{G} + s_0\mathtt{tr}(\mathbf{G})\mathbf{I}$, and finally the forces for the four vertices.

1.d. Laplacian smoothing (1 Point). Since we use explicit time integration, the simulation is susceptible to numerical instability. Damping implemented in 1.a is crucial to numerical stability, but it cannot eliminate high-frequency oscillation. To address this problem, please implement Laplacian smoothing over the vertex velocities. To do so, for every vertex, sum up all of its neighbor's velocities and then blend it into its own velocity. Can you do this without using any neighborhood topology?

2 Bonus Tasks (4 points)

An alternative way of calculating the stress is to treat the strain energy density as a function of matrix invariants, or principal stretches. This allows us to more conveniently adopt arbitrary hyperelastic models. Given the SVD function already being provided, can you implement this idea with the StVK model and show that the result is equivalent? (Hint: The instructor's website has CUDA code for GPU-based hyperelastic simulation in C++. You can use that for reference.)

3 Submission Guideline

Save all of your files, including scene and script files, and export them into a package. Submit your package by the SmartChair system.