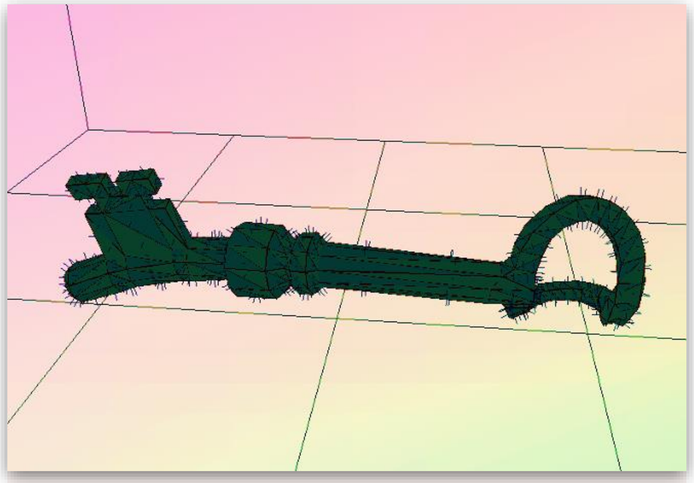
Soft Body Dynamics on the GPU using Shells

Student: Noé Pagés Puertas | Supervisor: Dr. Benjamin Kenwright | Second Marker: Prof. Jon Kerridge

Edinburgh Napier UNIVERSITY

April 2015



Key model deforming after colliding with the floor.

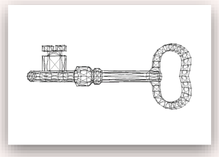
Aims

- Combining *shells* and *position based dynamics* to create a new technique to simulate *soft bodies* in *real-time*.
- *GPU* parallelisation of the technique.
- Study of collision detection techniques with a focus on the *narrow phase*.

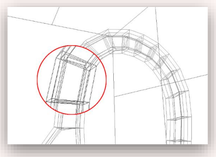
Background

Soft bodies are a field of computer physics animation in continuous demand in the fields of video-games and computational surgery. The simulation of cloth has been used for real-time recreational purposes for a long time. *Position based* approaches are easy to understand, permit attaching soft bodies to characters and are effective when solving collision detection through the *penalty based* system. *Shells* are mesh projections that create concentric layers commonly used to optimise the simulation of *fur* in *real-time*.

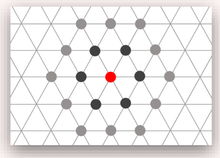
Methodology



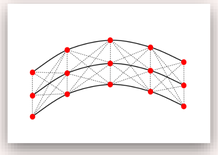
3D mesh.



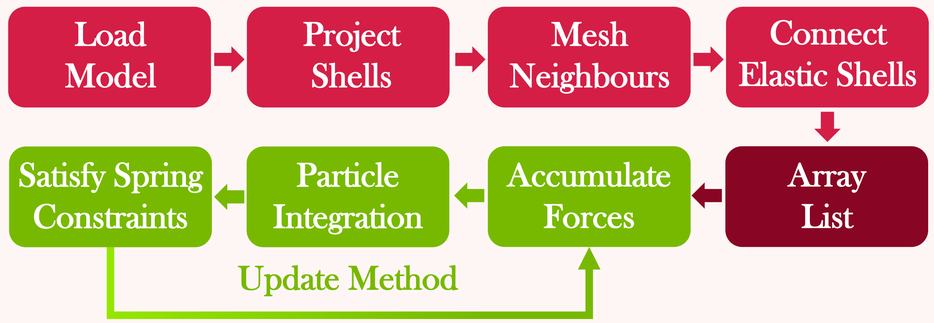
Concentric Shells.



Particle neighbourhood.

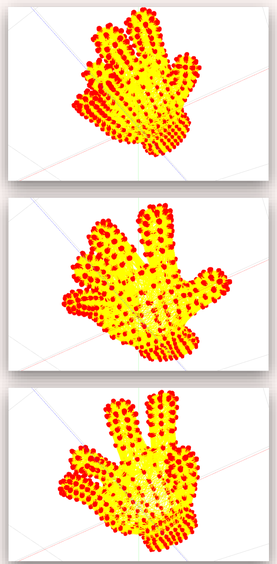


Neighbour connections.



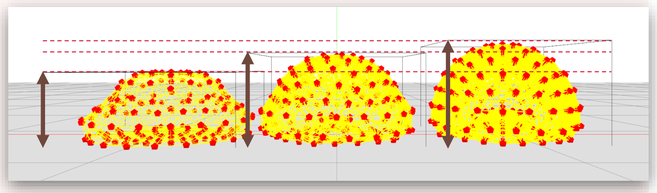
Results

A Deformation



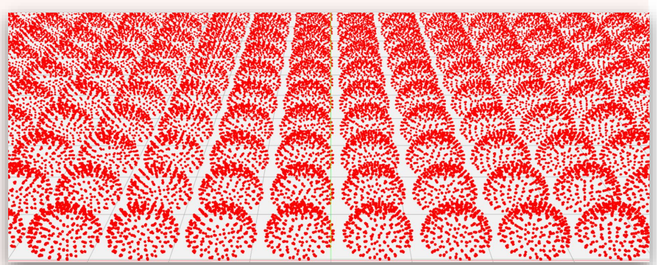
Soft body glove.

B Shape recovery based on shells



From left to right: one, two and four shells.

C GPU version



Soft body spheres on the GPU.

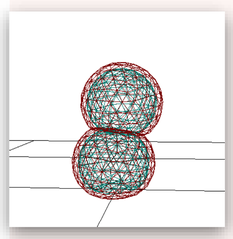
This project has been sponsored by NVIDIA. NVIDIA.

Conclusions

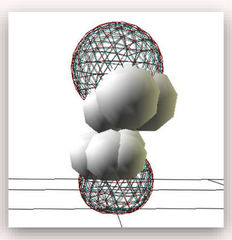
- *Shells* can be parallelised on the *GPU* to simulate soft bodies.
- Configuring the network of shells impacts the model's shape *recovery capability*.
- The diverse *topology* of 3D models defines the procedure to connect shells for parallelisation.
- *Soft body dynamics* is a mature field with a large amount of techniques and it is early to determine whether this project makes any valid contribution in the field.

Future Work

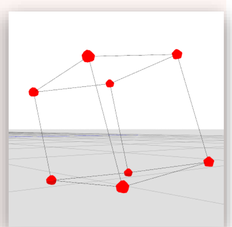
- Combining *soft body* dynamics and *fur* could be a new approach for using *shells*.
- A *GPU*, *shared memory* model could contribute on achieving *speedup*.
- Implementing *API GPU interoperability* would greatly optimise the method to stream data.
- Distance and angular constraints could be combined in a hybrid type of constraint.
- *Collision Detection* has been studied in the broad and the narrow phase. However, a solution that can deal with *concave* angles must be explored.



Narrow Collision Detection.



Spatial partitioning.



Angular constraints.