(Volume-preserving) Soft Object Denting

Blender Script

by Marcel Quanz

Problem and Motivation

Consider: soft object deformation in response to another (hard) object.

E.g.: A soft ball is grabbed and squished by a hand.

- Area underneath the fingers is pushed down.
- Additional displacement around this area (depending on rigidity/elasticity)
- Displaced volume is added to the rest (depending on internal pressure)

Materials science terms: Elastic modulus, Poisson's ratio

Existing “Solutions” (in Blender)

- Softbody simulation
  - \+ Physically accurate (more or less)
  - \+ Automatically updates
  - \- Requires animation (depends on time)
  - \- Hard to control (depends on many parameters)
  - \- Sometimes too realistic for artistic expression

Existing “Solutions” (in Blender)

- Sculpting:
  - \+ Allows full artistic expression
  - \- Requires a lot of time and work
  - \- Results may vary

Existing “Solutions” (in Blender)

- Dynamic Paint:
  - \+ Relatively quick to use
  - \+ Automatically updates
  - \- Requires animation (to some extent)
  - \- Only displaces (no volume preservation)
  - \- Can cause artifacts

Existing “Solutions” (in Blender)

- Boolean:
  - \+ Displacement is quite exact
  - \- Only displaces (no volume preservation)
  - \- Creates additional geometry
  - \- Either destructive, or otherwise problematic

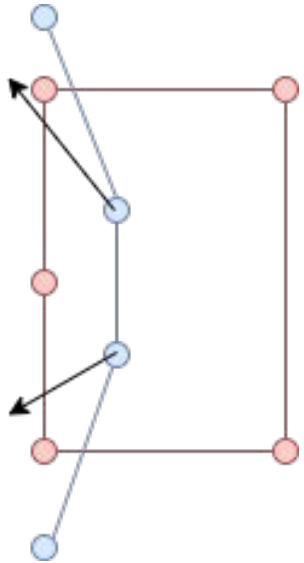
This Implementation...

- Preserves volume (approximate)
- Is not based on time (one-click solution)
- Does not add additional geometry
- Tries to keep artifacts to a minimum
- Has parameters to control artistic (not physical) properties
- Works non-destructively on shape keys (can be animated)

Part I a: Pushing in

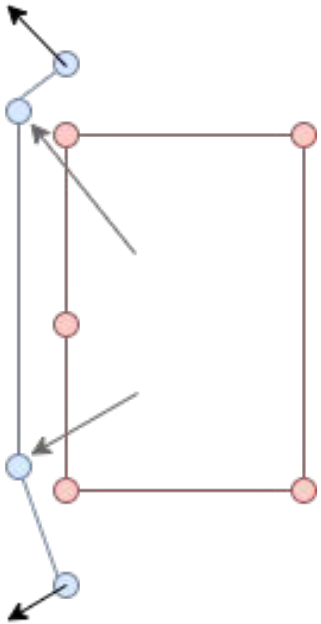
1. Find all soft object vertices inside the hard object
2. For each vertex inside:
    1. Find some average displacement direction depending on the surrounding hard object vertices
    2. Get the closest surface point on the hard object surface along this direction
    3. Put the vertex on this point (+ some optional, additional displacement)

We save the total displacement amount for later.



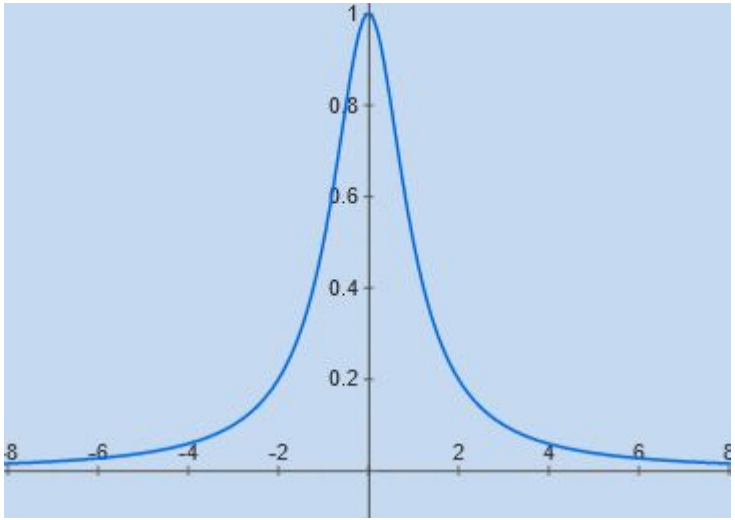
Part I b: Pushing around the area

- Depending of the rigidity/elasticity parameter, we also displace vertices around the initial area
- This displacement amount is based on two factors:
  - The distance to the initial area (inverse square)
  - The rigidity/elasticity factor (percentage of the displacement of closest vert)



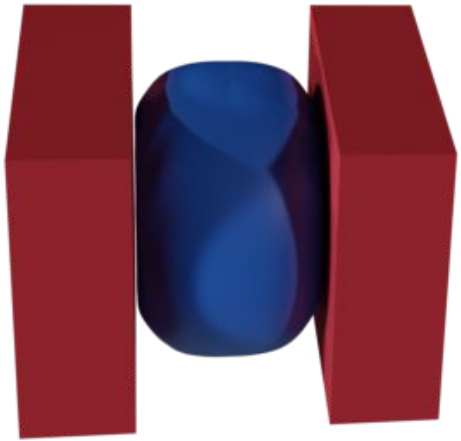
Part II: Adding back the volume

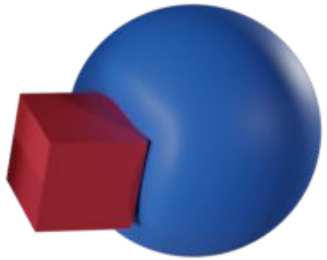
1. Summing up the inverse square sum of all verts not affected in Part I
   For each vert: $1 / (1 + (dist\_min - dist\_inf)^2)$ where:
   dist_min: distance to the closest vert of the overlap area
   dist_inf: minimum of all dist_min
2. Calculate the inverse square factor x by dividing the total displacement distance from Part I by the inverse square sum
3. Move each vert by $x * 1 / (1 + (dist\_min - dist\_inf)^2)$ along the vertex normal



Results

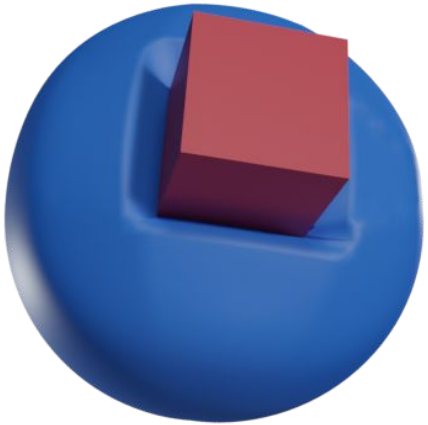
Not too bad…

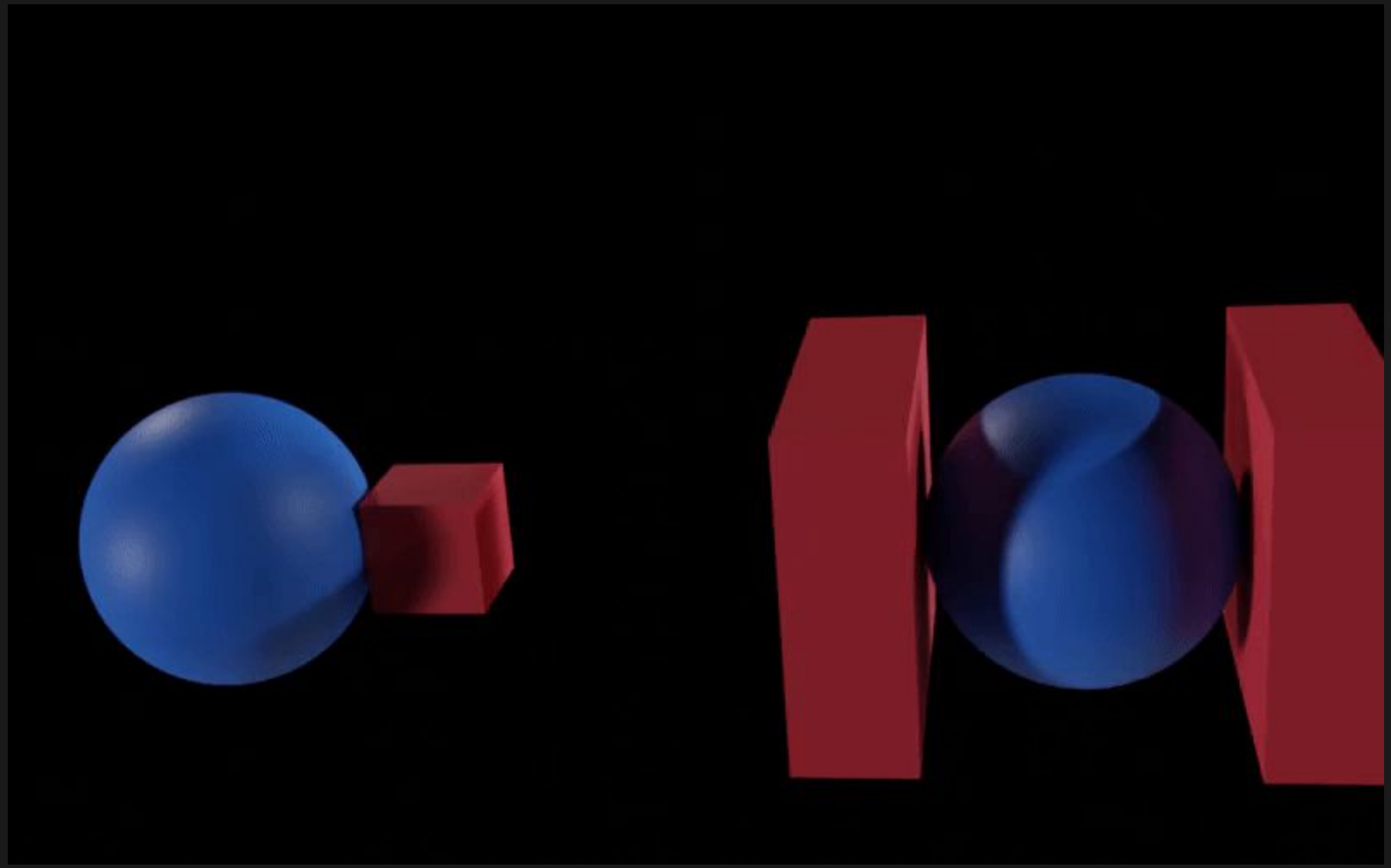




Results

…but not great





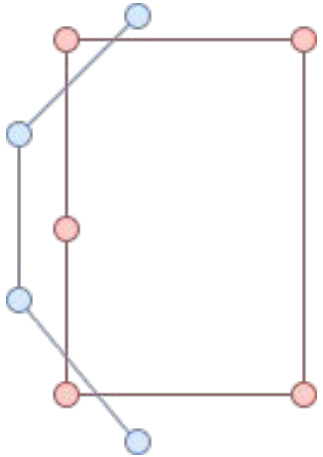
Current Problems

1. Because pushing in (Part I) is based on overlapping vertices, faces/edges may still overlap.

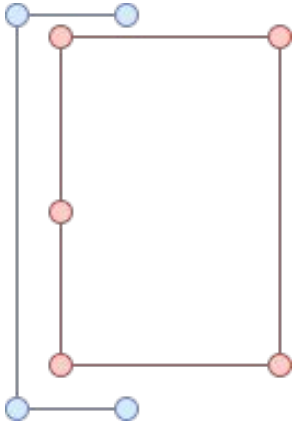
   Two ways to fix this:

   a. Easy: Also displace vertices around the initial area (already done to some extent through rigidity/elasticity)

   b. Hard (but better): First try to find defining features of the hard object (convex hull + 1° planar decimate) and displace verts onto those, then continue as normal.



We want this:

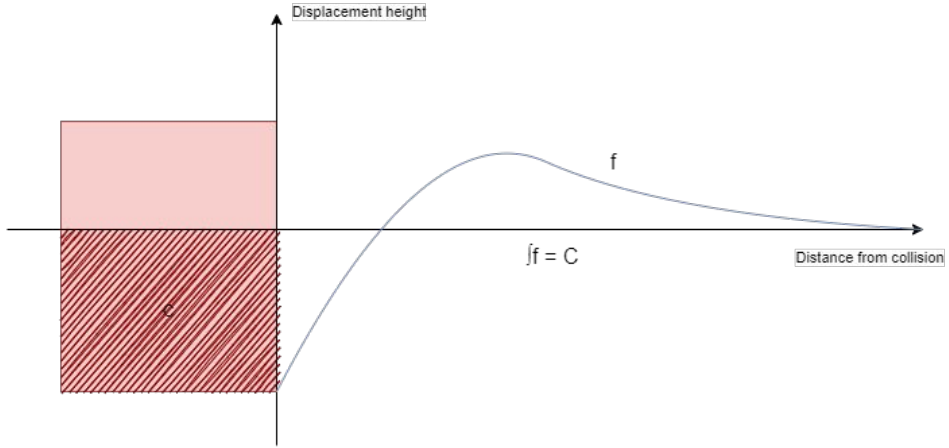


Current Problems

2. The functions for elasticity and volume preservation are not continuous when combined

Interpolation could help, but somehow it makes it look worse.

In theory, a single function for both would be best, but we would have to solve for the integral…



Current Problems

3. Because (Part I) vertex displacement is based on the area around it, it can be affected by vertices it should not be affected by.
   This only happens if there multiple overlap sections.

   How to fix this:

   Isolate the impact/overlap areas and work with them separately (can also be done with multiple hard objects)

Current Problems

4. Volume preservation is purely based on the displacement distance, not on volume. As such, it only works well for even meshes.

   Not actually much of a problem, but better volume estimation might help

Questions?