Squashing Cubes: Automating Deformable Model Construction for Graphics

Doug L. James Jernej Barbič Christopher D. Twigg (Carnegie Mellon University)

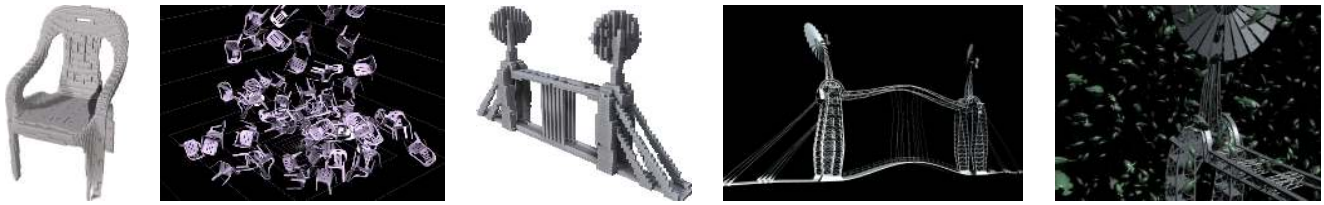

Figure 1: *Squashing Cubes models and simulations:* (Left) a flexible plastic chair; (Right) a complex swaying bridge superstructure.

Introduction

The vast majority of geometric meshes used in computer graphics are optimized for rendering, and not deformable object simulation. Despite tools for volume (or surface) (re)meshing of geometric models to support physical simulation, in practice, the construction of physically based deformable models from arbitrary graphical models remains a tedious process for animators.

Squashing Cubes (SC) automates the construction of physically based deformable objects from arbitrary geometric models. During preprocess, the geometric model (typically a surface mesh) is voxelized into tiny elastic cubes, i.e., the *squashing cubes model*. Second, a generic deformable object simulator is used to deform the SC model. Finally, the resulting deformations are interpolated back onto the original model, thus producing the final animation. Such domain embedding schemes are familiar to graphics [Pentland and Williams 1989; Faloutsos et al. 1997].

SC is simple to implement, practical for complex models, supports any geometric representation, and the SC deformable models are trivial to simulate. Although SC geometry is approximate, reasonable approximations of deformation displacement fields can be obtained for many animation purposes. One practical benefit of voxelization is that geometric features and defects smaller than the voxel scale are merged; consequently, intersecting and topologically disconnected polygons (so common in graphical models) are deformed appropriately. SC is especially appealing for complex superstructures, such as the bridge, for which the geometric mesh is not "structurally sound." For example, the bridge model contains numerous modeling shortcuts, such as improperly attached cables, intersecting geometry, small "gaps," badly shaped (skinny) triangles, and isolated polygons. In short, Squashing Cubes allows animators to focus on animation, and less on physically based modeling details, such as the engineering of bridges using properly connected beam, truss, and shell elements.

Squashing Cubes Model Construction

The resolution of the SC deformable model can be chosen independent of model geometry, and is typically less complex (see video). Given a fitted 3D voxel grid, the SC model contains all voxels occupied by the geometric model (or the largest connected component). These cubes can be found by scan conversion, or via (hierarchically-optimized) polygon-cube intersection tests. For simplicity, in this sketch we consider a point-based sampling approach that is trivial to implement, and suitable for any geometric representation.

Consider a large number, N, of near uniform surface (or volume) point samples, where each sample consists of a position p_j, a tiny mass δm_j (e.g., m/N), and a list of local material properties, e.g., (E_j, ν_j). For each sample hashed to a voxel, a squashing cube object is either created or retrieved, and used to accumulate any material properties. For example, an elastic cube's mass is simply the total point sample mass; this mass is evenly distributed to vertex nodes in vertex-based simulation schemes. The bulk material parameters for each elastic cube are chosen as the weighted average of material values for all contained samples.

Simulation

Given the list of squashing cubes, the list of adjacent vertices (or nodes) can be obtained for numerical simulation. Discretization and dynamics integration of the elastic cubes can be approximated using virtually any deformable object simulator, including approaches based on finite differences, finite elements (brick elements), or mass-spring particle lattices; a cloth or shell simulator could even be used to simulate a SC model. At each time step, the deformation of the SC model is interpolated back onto the original geometric model; the displacement at any point within a squashing cube is determined using trilinear interpolation of cube vertex displacements. For modal analysis [Pentland and Williams 1989], e.g., chair model, the SC model's eigenmodes can be interpolated back onto the original geometry during preprocessing, and the SC model discarded.

Collision detection can be approximated using the SC model, although in our simulations we use the underlying geometry directly. Contact forces applied to underlying geometry are (barycentrically) distributed to the nodes of the corresponding cube, and the resulting SC forces are used in dynamics integration.

Discussion

Limitations of coarse voxel resolutions are (a) close geometric components may be undesirably joined, and (b) the bending of thin surfaces can reflect voxel "stair-casing." Although we considered surface models, volumetric models can also be sampled, or computed using flood-fill voxel operations. Finally, high cube counts can occur for volumetric objects, but are very reasonable for thin structures, e.g., bridge; nevertheless, the computing costs should be weighed against increased animator productivity.

References

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 1997. Dynamic Free-Form Deformations for Animation Synthesis. *IEEE Trans. on Vis. Comp. Graph. 3*, 3, 201–214.

PENTLAND, A., AND WILLIAMS, J. 1989. Good Vibrations: Modal Dynamics for Graphics and Animation. In *Computer Graphics (Proceedings of SIGGRAPH 89)*, vol. 23, 215–222.