Geometric fracture modeling in BOLT

Figure 1: *Left, middle: Rhino's ball is riddled with cracks as a metal gate crushes it down. Right: A roadway is torn up by Bolt's "superbark".*

1 Introduction

Modeling the geometry of solid materials cracking and shattering into elaborately shaped pieces is a painstaking task, which is often impractical to tune by hand when a large number of fragments are produced. In Walt Disney's animated feature film *Bolt*, cracking and shattering objects were prominent visual elements in a number of action sequences. We designed a system to facilitate the modeling of cracked and shattered objects, enabling the automatic generation of a large number of fragments while retaining the flexibility to artistically control the density and complexity of the crack formation, or even manually controlling the shape of the resulting pieces where necessary. Our method resolves every fragment *exactly* into a separate triangulated surface mesh, producing pieces that line up perfectly even upon close inspection, and allows straightforward transfer of texture and look properties from the un-fractured model.

2 Crack geometry generation

The input to our system consists of a closed triangulated surface defining the (uncut) solid object to be fractured and one or more additional triangulated surfaces defining the geometry of the cracks. The geometry of this "crack surface" is not constrained by the shape of the material object itself; cracks are free to extend outside the material into the empty space, and can have non-manifold shapes, topological junctions, or even intersect themselves. Leveraging this flexibility, we can automatically create a fracture surface which partitions the ambient space into any given number of regions by simply introducing the same number of seed points and computing the 3D Voronoi regions of those seed locations. The fracture surface is then defined as the union of all boundaries between Voronoi cells (see Fig. 2, top right). In practice, we approximate these regions by laying down a background, procedurally generated tetrahedral mesh, and computing the Voronoi cells via a flood-fill on the tetrahedral mesh, starting from the elements containing the seed points.

The shape of the fracture surface can be controlled by specifying the number of seed points, or even by volumetric painting of seed point densities, to control which regions will shatter into more, smaller fragments. We also control the smoothness of the boundaries between Voronoi regions by jittering the background tetrahedral mesh prior to the flood-fill, and selectively smoothing the boundary surface where smoother cracks are desired. Finally, in certain situations (e.g. the cracks of the hamster ball in Fig. 1), more specific artist control of the fragment geometry is desired. For these cases, we extruded sets of artist-drawn 3D curves in the direction normal to the object surface to create a fracture surface that cuts through the material and reflects the crack design intended by the artists.

3 Automatic fragment mesh generation

We adapt the cutting algorithm of [Sifakis et al. 2007] to automatically generate triangulated meshes for the material fragments defined by the object and fracture geometries of the previous section. In particular, their method begins with a tetrahedral volumetric representation of the material to be fractured, as opposed to the triangulated boundary geometry we assumed as input to our system. We handle this representation discrepancy by first generating a tetrahedral mesh that fully covers the object to be fractured . The triangulated surface of the uncut object itself is used as the first cut in an application of the algorithm of [Sifakis et al. 2007], effectively sectioning the background tetrahedral volume into the "material" and "void" regions. The fracture surface is then applied as the second cut, resulting in the separation of the material volume into separate fragments. The cutting algorithm computes the triangulated boundary of every volumetric fragment in a way that every triangle of a fragment is contained inside a triangle either of the uncut object, or of the fracture surface (Fig. 2, bottom). As a result, texture and look properties can be remapped simply by embedding each resulting triangle barycentrically into either the material or the fracture surface respectively, and looking up the properties of the embedding triangle. Finally, although our framework is currently used for geometric modeling, we aim to employ it in conjunction with simulation in the future, to model time-dependent crack propagation.

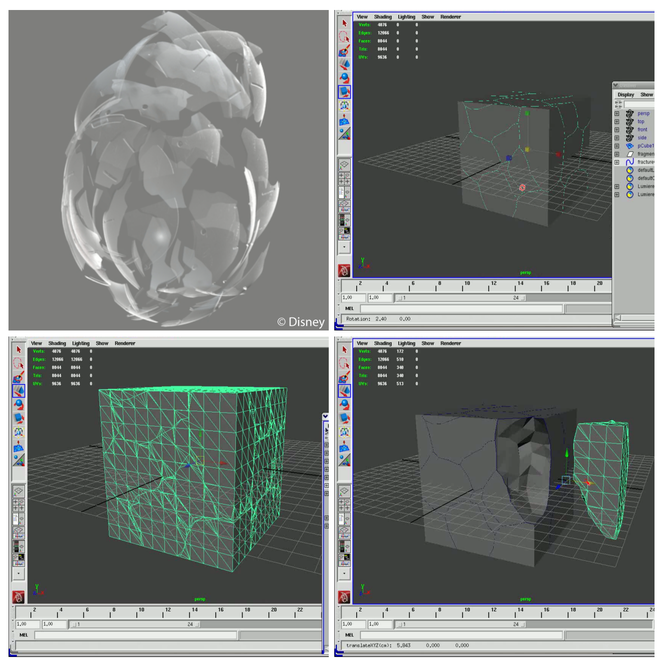

Figure 2: *Top left: Simulation of the shattered fragments of Rhino's ball. Top right: Fracture surfaces defined as the boundaries of Voronoi regions in 3D. Bottom: The fragments are fully resolved as independent surface meshes, and can be separately manipulated.*

References

SIFAKIS, E., DER, K., AND FEDKIW, R. 2007. Arbitrary cutting of deformable tetrahedralized objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 73–80.