Seamless Fracture in a Production Pipeline

Akash Garg[*] Kyle Maxwell[†]

DreamWorks Animation

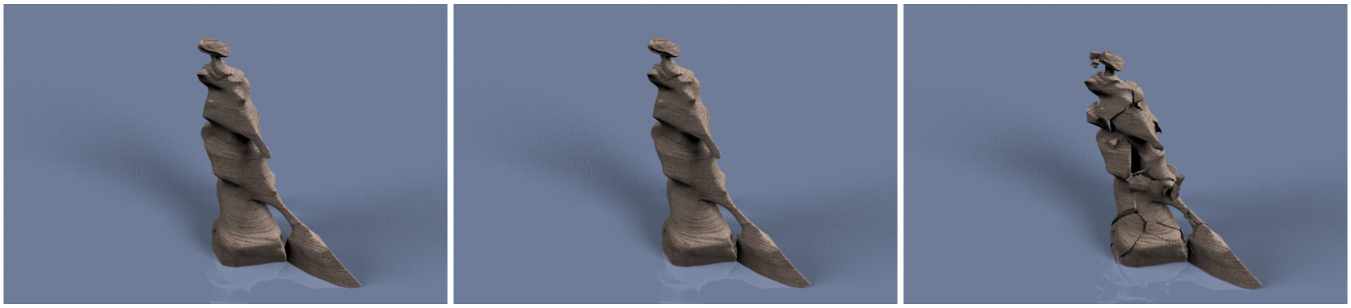


Figure 1: *Fracturing a cliff. Left: The original surface geometry, Middle: the scan-converted, fractured, and reconstructed geometry at frame 1, Right: the fractured geometry at frame 25. Note the fidelity to the original geometry and preservation of sharp features.*

Contributions To alleviate the laborious task of generating volumetric fracture pieces by hand while preserving the integrity of the artistically modeled shape, we develop a practical, robust, and effective solution for artistically directed volumetric fracture. Unlike previous approaches our procedural fracture system maintains intrinsic attributes (e.g., texture coordinates), preserves discontinuities such as texture seams, and generates procedurally fragmented "chunks" with *seamless* boundaries. The key steps in our fracture pipeline are: robust scan-conversion from an input geometry, generation of fragmented fracture pieces with artistic control, and high fidelity mesh generation of the disjoint fracture fragments.

Scan-conversion The input to our fracture tool is a base geometry, usually a textured polygonal or NURBS surface. To handle self-intersecting, open, and non-manifold geometry, we use a robust scan-conversion algorithm based on the work of Nooruddin & Turk [Nooruddin and Turk 2003]. The scan-converted volume stores 2 key pieces of information: (i) inside/outside markers for fast CSG operations and (ii) surface normals and attribute information at the edges of voxels that intersect the surface. Storing this data intrinsically in the volume aids faithful mesh reconstruction of the input geometry, while preserving sharp features and attribute discontinuities, such as texture seams. The high fidelity between the pre-fractured and post-fractured geometry allows fracturing complex geometry with little to no post-processing, see Figure 1.

Artistic Driven Fracture Once we have scan-converted the original geometry, we use the resulting volume to generate disjoint fragments. The user has two options for controlling the creation of these fragments. The first, gives the user fine grained control over the fracture locations via an intuitive 3D painting interface; the second creates a multitude of fragments by automatically generating Voronoi cells in R^3, useful for large-scale destruction. Once created, boolean intersections are carried out between these fragments and the scan-converted volumetric object. The resulting fractured pieces are then reconstructed as polygonal meshes, used for animation/simulation and rendering.

Seamless Mesh Generation Once the volume has been fragmented using boolean intersections, each individual fracture piece is output explicitly as a mesh. To avoid rendering artifacts, it's important that no seams be present when neighboring fragments adjoin. Traditional tessellation algorithms like the one used in [Museth and Clive 2008] leave small gaps between adjacent fracture pieces. To this end, we use the surface normals stored during scan-conversion and augment Ju et. al's Dual Contouring method [Ju et al. 2002] to tessellate individual fragments. Our dual contouring tessellator ensures not only that sharp features are preserved but also that no gaps are present between adjoining fracture pieces (see Figure 2).

In addition to preserving geometric seams between adjacent fragments, attribute values also need to vary smoothly across fragment boundaries. Typically a post-process such as closest point projection is used, which although efficient, is problematic for high curvature geometry. To share attribute values across disjoint fragments, we introduce *ghost-faces*, faces that are never displayed and are only used as a means to transfer attribute information. Consider the following scenario on the left. The red vertex is shared among two different fractured pieces. Consistent interpolation of attributes can only be achieved if data from the adjacent faces in the adjoining fracture piece is present. To this end, The black face consists of the 3 blue ghost-faces such that the vertex's attribute value can be interpolated accurately.

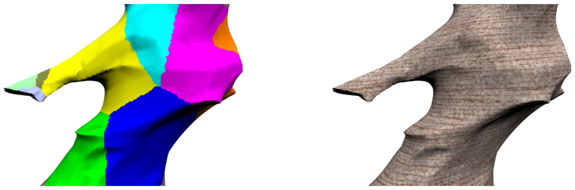
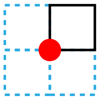


Figure 2: *Left: each fractured piece colored differently. Right: the fractured pieces with textures applied. Note that both textures and geometry are continuous across fracture boundaries.*



References

Ju, T., Losasso, F., Schaefer, S., and Warren, J. 2002. Dual contouring of hermite data. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 339–346.

Museth, K., and Clive, M. 2008. Cracktastic: fast 3d fragmentation in "the mummy: Tomb of the dragon emperor". In *SIGGRAPH '08: ACM SIGGRAPH 2008 talks*, ACM, New York, NY, USA, 1–1.

Nooruddin, F. S., and Turk, G. 2003. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics 9*, 2, 191–205.
