Example or APA format?

**Rigid Body Destruction in the Houdini Engine: A Literature Review**

Jason Beal

East Tennessee State Universi

**(CENTER) Rigid Body Destruction in the Houdini Engine: A Literature Review??**

Purposeful destruction of solid objects has become an interesting part of video games and film. These effects enhance gameplay mechanics to add layers of interest. Films, particularly those involving superheroes, include the wide-scale destruction of entire buildings across multiple city blocks. These kinds of scenes require an intense simulation on platforms like Houdini that have a robust array of tools to control the details of these amazing scenarios. Houdini’s procedural node-based programming allows for complex simulations to be changed quickly by changing just a few attributes. This allows for finely tuned art direction without long hours of repetitive tweaks. Tools called Houdini Digital Assets (HDA) can be used in other programs SUCH AS the Unreal Engine to speed up workflow. JUSTIFIED LEFT

To inform the research into the destruction of rigid bodies and their collision calculation within the Houdini engine, various relevant sources have been selected to answer the following questions:

1. What tools or methods were researched to lay the foundation for the methods and tools that produce destruction effects in Houdini?
2. What tools have been implemented in Houdini to create interesting, dynamic, and accurate destruction of rigid body objects?
3. What methods were used to optimize the production of these effects?

**~~Body~~**

**Methods And Tools That Produce Destruction Effects In Houdini**

There are multiple advances to rigid body destruction that may apply to Houdini given the right plug-in. The development of modern technologies and methods used in Houdini today were dependent on research that continues to move forward. Osher and Sethian (1988) did work for the National Aeronautics and Space Administration (N.A.S.A.) to “track the motion of a front whose speed depends on the local curvature” (OSHER&SETHIAN!!, 1988, p. 1) for flames and hot gasses. This work centered around the “formulation of the correct equation of motion for a front propagating with curvature-dependent speed” (OSHER&SETHIAN!!, 1988, p. 1). They were able to visualize topological changes in moving objects by viewing the surface as a level set.

Research into level sets continued to progress and improve the optimization of three-dimensional simulations. One such level set that combined DT-Grid with Run-Length Encoded (RLE) level set was Hierarchical Run-Length Encoded (H-RLE) level set. H-RLE allows for “compact storage of sequential nonnarrowband regions while the dimensionally recursive encoding along each axis efficiently compacts nonnarrowband planes and volumes” (Houston, 2006, p. 1). ONE!! can store a level set with 45 billion voxels in just 1 GB using this process. This made the Tar monster from Scooby Doo 2 possible (Houston, 2006, p. 163). Houston (year) and Chitalu (year) agreed about the limitations of bounding volumes used for parallelizing collision detection and that more research will be necessary to make advancements in this technical process.

Researchers at the Institute of Electrical and Electronics Engineers (specific format for thesis) found exciting methods that “generate a volumetric level set representation for each small shard” (Zhaosheng, 2007, p. 371) for the destruction of thin-shelled rigid bodies and deformation of objects penetrated by a projectile in a ductile pattern. Using the volumetric level set, they determined the “fracture occurs when the principal components of the element stress are above a certain threshold and that the fracture surface is locally planar” (Zhaosheng, 2007, p. 375). They managed to keep the deformation realistic by using a method called time averaged stress as opposed to the general snapshot method of collision detection that “can be arbitrarily high causing spurious fracturing” (Zhaosheng, 2007, p. 374). Level sets are a database that “exploits the spatial coherency of uniform grids to effectively and separately encode data values” (Museth, 2009, paragraph 2) for use in simulating volumetric shapes like clouds or water. The constantly changing shape of volumetric objects makes them time-intensive to render so level set’s “compact and very fast volumetric data structures allow us to work at unprecedented grid resolutions”(Museth, 2009, paragraph 1). Modal analysis “takes into account the geometry of the impacted body, estimates contacts duration, and chooses appropriate time steps to simulate the deformations during contacts” (Glondu, 2013, p. 201) and manipulates implicit surfaces to represent the fracture surfaces. Voronoi, a type of fracturing, measures strain on an object and creates a higher number of smaller pieces at the points of greatest strain. An energy criterion can be used to limit the “sampling of Voronoi centroids if the energy of the indirect created surface exceeds the initial strain energy available” (Glondu, 2013, p. 202)

**What tools have been implemented in Houdini to create interesting, dynamic, and accurate destruction of rigid body objects?**

Houdini is an advanced modeling and simulation engine that excels at “particles and integrated dynamics, such as rigid bodies, wire, cloth, and fluids.” (Side effects makes Houdini 9.5 appear, 2008, paragraph 4) These simulations are created using a node-based procedural workflow. This procedural workflow cuts most of the repetitive tasks out of the process allowing users to quickly build and render high-quality products with greater flexibility to be used in games or film.

Houdini has many tools built-in to solve problems in particular scenarios. For example, the Bullet Solver node was used in LAIKA’s film, “The Missing Link”, to break apart a bridge made of ice while a clay-animated character stands on it (Horsley, 2019). The Bullet solver node solves all of the velocity and collision detection calculations in one package. (RBD Bullet Solver geometry node n.d.). A simple constraint network was used to hold up the ice blocks of the bridge until they were moved during the collapse. A Point Deform Surface Operator (SOP) node was used for a “sagging effect” (Horsley, 2029, p. 2). The difficulty of mixing clay animation and a CGI effect required a change in the workflow of this shot so the filmmakers “started with the bridge animation and used that to drive the character animation” (Horsley 2019, p. 2). For any Houdini rigid body destruction simulation, the “Dynamics Network [is] where all the fun takes place” (Magee, 2010, p. 80). It loads any objects involved in the destruction into a single node to “establish the connections between them, and configure solvers for simulating their interactions” (Dynamics nodes dynamics node, n.d., paragraph 1). Every model can be adjusted with fine-tune control over how it breaks using a scatter node to spread points along the model, so it can be “evaluated frame by frame, and are broken in reaction to the impact of colliding objects” (Magee). In contrast to Houston’s use of non-narrowband planes with H-RLE, LAIKA decided to use an adaptive narrow band to simulate a thin layer of particles for a rogue wave.

Houdini has different nodes for different types of fractures. For a basic fracture, like a rock, “fractures create jagged edges along various cracks” (Elkins, 2020, p. 6). For wood, a splintered fracture will have “random splintering along the cut” (Elkins, 2020, p. 6). Using a shatter-type fracture on a thin piece of glass creates “an almost spiderweb-shaped break, with the smaller pieces generating at the contact point” (Elkins, 2020, p. 6). The Rigid Body Dynamics (RBD) material fracture node accepts inputs for high-resolution geometry, the constraints on that geometry, and proxies to stand in for the geometry. Constraints can be used to link pieces together until a threshold of force has been reached. This allows the object to break apart realistically instead of all at the same time. RBD Pack and Unpack merge and split these three inputs into one output for simplicity.

The foundation of Houdini’s workflow is the nodes that can contain nodes within themselves with all of the code needed to generate procedural content. The Surface Operator (SOP) node isn’t for modeling as much as “build[ing] a process for making a model” (Zerouni, 2012, Ch. 2, paragraph 1). There are SOPs for numerous shapes but also tools like Copy SOP, Transform SOP, Add SOP, and Subdivide SOP. Some nodes like Softpeak SOP can be used for some more direct modeling. When it comes to destruction with an RBD simulation, you first need to scatter points on an object with Connectivity SOP and Partition SOP to “put each chunk into its own group” (Zerouni, 2012, Ch. 2, paragraph 11). Houdini has some limitations with UV layering such as “the Layer value does not affect the AttribCreate nor the AttribPromote SOPs” (Zerouni, 2012, Ch. 2, paragraph 69). A method for saving a cooked SOP in memory for faster recall when it is needed with the Unload tool. (Zerouni, 2012)

The Particle Operator node runs all the effects of particle simulations and is embedded in the Dynamics Operator (DOP) for caching. RBD objects are included in the DOP node and can be fractured based on the locations of the particles dividing it and the type of fracture. This can include an assortment of force nodes that move the objects in ways that can cause them to collide with other objects or each other. These impacts generate collision impacts that apply stress to the fractured pieces. Once a threshold has been passed, the constraints holding the pieces together are broken and they then follow their physics separate from the core object (Zerouni, 2012, Ch. 3).

The DOP generates gravity and a ground plane by default to save time on the setup. Within this node, a UI called “the Affector Matrix” (Zerouni, 2012, Ch. 4, paragraph 6) is color-coded to communicate the relationship between the pieces of the RBD objects in the simulation. The RBD state node contains the settings for the glue constraints. Glue constraints can only be broken automatically by collision contact, not by any kind of force. With keyframes, a user can set the glue constraints to break manually at the appropriate time. A tool, the “dopnumrecords() function” (Zerouni, 2012, Ch. 4, paragraph 77), exists to create a reaction decided by the user when a collision occurs. This can be used for an array of other effects generated dynamically. There is a limit to the memory of this tool because it only exists on the frame where the collision takes place. With these tools, only the user’s imagination can limit the possibilities.

**What methods were used to optimize the production of these effects?**

Destruction leaves behind an even bigger problem to solve, collision. These collision calculations can be very expensive, in rendering seconds, to keep up on every frame with hundreds possibly thousands of tiny pieces. Chitalu found that using “implicit binary trees in broad-phase collision detection” exhibited significant refinement in optimization (2020, p. 3). Methods like “bounding volume hierarchy (BVH), which is a widely used tree data structure for collision detection”(Chitalu, 2020, p. iv) have been implemented to vastly increase the speed that Houdini renders these complex scenes. This data structure stores data about the triangles that cover the surface of objects in a game environment. One of the advantages of using a BVH is the ability to balance parallel collision detection calculations across all of the GPU threads. (Chitalu 2020, p. 10) This data is referenced to quickly determine when these triangles will cross over each other and thus, return a collision. A new method called “re-meshing free” or boundary elements” has offered an alternative to the “finite elements” (FEM) method that is already used. Re-meshing free allows for calculating cracks in an object but may “prove limited on objects with large surface-area-to-volume ratios” (Chitalu, 2020, p. iv).

A toolkit was developed for Houdini called a voxel database (VDB) for “highly efficient implementations of level set operations like advection/propagation, re-normalization, narrow-band rebuilding, boolean operations, morphological operations, adaptive meshing, morphing, collision detection, mesh-to-level-set conversion” (Museth, 2009, para. 3). This was made possible through the use of an “open-source C++ library comprising a novel hierarchical data structure and a suite of tools for efficient storage and manipulation of sparse volumetric data discretized on three-dimensional grids.” (Museth, 2013, paragraph 1).

**Conclusion**

Rigid body destruction is an advanced and spectacular process that includes a quickly growing body of research. The eye-catching scenes that are produced enhance the most exciting parts of both films and games. The tools and methods such as an RBD Bullet Solver geometry node, a Point Deform SOP node, a Dynamics Network node, basic fracture, splintered fracture, shatter fracture, Voronoi fracture node, boolean fracture node, glue constraints, an RBD Pack node, an RBD Unpack node, bounding volume hierarchy, level sets, re-meshing free, boundary elements, finite elements, H-RLE, modal analysis, and voxel database node are what make these effects possible. Advancements in the workflow, efficiency, and quality of rendered simulations have made these effects more common and more shocking. Scientific research made these possible and experiments are ongoing. The findings often can be applied within Houdini to improve the power developers have to create their visions in the way they desire. There are experiments done every year on numerous methods, hardware, and software to make creating destruction easier to make, quicker to render longer sequences, and higher in visual fidelity and quantit

References

Chen, Yao, M., Feng, R., & Wang, H. (2014). Physics-inspired adaptive fracture refinement. *ACM Transactions on Graphics, 33*(4), 1–7. <https://doi.org/10.1145/2601097.2601115>

Chitalu, F. (2020). *Accelerating and simulating detected physical interactions* [The University of Edinburgh]. <https://doi.org/10.7488/era/388>

Elkins, Ethan B., "Simulating Destruction Effects in SideFX Houdini" (2020). *Undergraduate Honors Theses*. Paper 524. <https://dc.etsu.edu/honors/524>

*Art directing your destruction in Houdini.* (2017). Vimeo. Retrieved on February 28th, 2023. <https://vimeo.com/198266808>.

Glondu, Marchal, M., & Dumont, G. (2013). Real-Time Simulation of Brittle Fracture Using Modal Analysis. *IEEE Transactions on Visualization and Computer Graphics, 19*(2), 201–209. <https://doi.org/10.1109/TVCG.2012.121>

Harrower, Kyme, P., Scheepers, F., Rice, M., Tollec, M., & Moaveni, A. (2018). SimpleBullet: collaborating on a modular destruction toolkit. *ACM SIGGRAPH 2018 Talks*, 1–2. <https://doi.org/10.1145/3214745.3214814>

Horsley, & Stuart, P. (2019). The rigid body and fluid dynamics of LAIKA’s “Missing Link.” *ACM SIGGRAPH 2019 Talks*, 1–2. <https://doi.org/10.1145/3306307.3328143>

Houston, Nielsen, M., Batty, C., Nilsson, O., & Museth, K. (2006). Hierarchical RLE level set: A compact and versatile deformable surface representation. *ACM Transactions on Graphics, 25*(1), 151–175. <https://doi.org/10.1145/1122501.1122508>

Lai, Latham, W., & Leymarie, F. F. (2020). *Towards Friendly Mixed Initiative Procedural Content Generation: Three Pillars of Industry.* <https://doi.org/10.48550/arxiv.2005.09324>

Magee, R. (2010). Rigid body dynamics. *3D World*, 137, 76–81.

Museth, K. (2009). An efficient level set toolkit for visual effects. *SIGGRAPH 2009: Talks*, 1–1. <https://doi.org/10.1145/1597990.1597995>

Museth, K., Lait, J., Johanson, J., Budsberg, J., Henderson, R., Alden, M., Cucka, P., Hill, D., & Pearce, A. (2013). OpenVDB: an open-source data structure and toolkit for high-resolution volumes. *ACM SIGGRAPH 2013 Courses,* 1–1. <https://doi.org/10.1145/2504435.2504454>

Side effects makes Houdini 9.5 appear. (2008, September). *Computer Graphics World*, *31*(9), 4. <https://link.gale.com/apps/doc/A185331031/AONE?u=tel_a_etsul&sid=bookmark-AONE&xid=071ac47b>

SideFX. (n.d.). RBD Bullet Solver geometry node. *SideFX.* <https://www.sidefx.com/docs/houdini/nodes/sop/rbdbulletsolver.html>

SideFX. (n.d.). Dynamics nodes dynamics node. *SideFX.* <https://www.sidefx.com/docs/houdini/nodes/dop/index.html>

Zerouni, C. (2012). *Houdini On the Spot: Power User Tips and Techniques*. CRC Press. <https://doi.org/10.4324/9780080551470>

Zhaosheng Bao, Jeong-Mo Hong, Teran, J., & Fedkiw, R. (2007). Fracturing Rigid Materials. *IEEE Transactions on Visualization and Computer Graphics, 13*(2), 370–378. <https://doi.org/10.1109/TVCG.2007.39>