RLE Sparse Level Sets

Ben Houston
Frantic Films
bhouston@franticfilms.com

Mark Wiebe
Frantic Films
mwiebe@franticfilms.com

Chris Batty
Frantic Films
cbatty@franticfilms.com

1 Introduction

This paper introduces a novel scalable level set representation, the RLE (run-length encoded) sparse level set, and discusses the application of representing implicitly a character animated via traditional means.

2 RLE Sparse Level Set Structure

The RLE sparse level set representation has many beneficial characteristics: (1) highly scalable, (2) fast to access randomly, (3) sequential traversal of the narrow band is optimal, (4) approximate distance values are provided throughout the bounding volume, and (5) easily adapted to work with the large majority of existing level set methods developed for regular grid structures. These advantages are not all shared by the octree, the sparse field method, or the sparse block grid scalable level set representations (reviewed in [Bridson 2003].)

2.1 Random Access. Efficient random access into the RLE encoded volume ([Curless & Levoy 1996] by contrast required linear search) is achieved by (1) restarting the RLE encoding at the beginning of each row and (2) augmenting the normal run-length encoding structure with two tables. A *run-start table* associates each run with its start voxel coordinate along the axis of compression. A second table associates the voxel coordinates of the two other axes to the first run of the corresponding run-length encoded row. Thus random access consists of: (1) finding the segment of the run-tables which corresponds to the row of interest using the second table described above, (2) using a binary search on the identified segment of the *run-start table* to find the run of interest, and, if the run is defined, (3) using the run data start offset to determine the data index of the voxel value. Random access time is thus reduced to $O(\log r)$ instead of $O(R)$, where r is the average number of runs in a single row and R is the total number of runs.

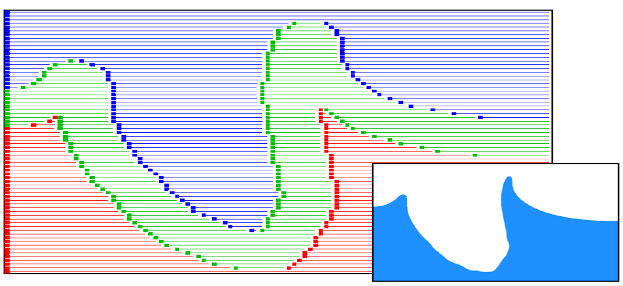

Figure 1. Each run is represented by a square followed by a line segment. $+\infty$, *defined*, and $-\infty$ runs correspond respectively blue, green and red.

2.2 Level Set Run Types. For the purposes of storing level sets we standardize three run types: *defined*, $+\infty$, and $-\infty$. If the narrow interface band is of width $2m$, then these run types denote respectively the cells of level set values: $|\Phi| \leq m$ (the narrow interface band), $\Phi > m$ (cells far outside of the level set defined region), and $\Phi < -m$ (cells far inside the level set defined region.) It is important to note that the *defined* run type differs from all other run types in that the values of such denoted cells are stored explicitly in an associated flat value array.

2.3 Encoding. Constructing an RLE sparse field structure has both space and time complexity of $O(n^2+R+D)$, where n is the side-length of the bounding volume, and D is the total number of defined voxels. Assuming that in any $O(n^3)$ grid there are only $O(n^2)$ cells close to the surface, as is the case for smooth enough geometry, the RLE sparse field structure scales with near optimal, $O(n^2 + R)$, space and time costs.

2.4 CSG Operations. Common level set CSG operations such as union, intersection and subtraction can be performed very efficiently, using only $O(R_a+D_a+R_b+D_b)$ operations when the encoding axes of the input RLE sparse level sets match.

2.5 Augmented Level Sets. Unlike many sparse structures, the RLE sparse field is decoupled from the storage of the defined values. Thus for level sets augmented with auxiliary data, such as the global occlusions representation introduced in [Houston et al 2003], a single RLE sparse field structure can be associated with multiple defined flat arrays of values, one for each auxiliary field desired.

3 Representing Animated Characters

Converting an animated character into a series of RLE sparse level sets in a robust and efficient manner while maintaining fidelity involves solving multiple challenges.

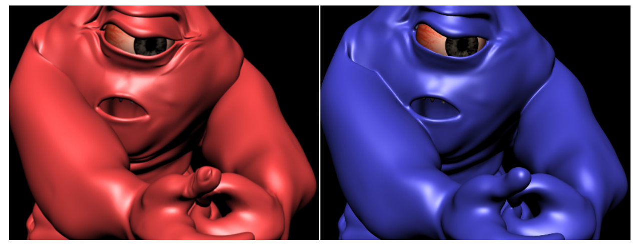

Figure 2. Original mesh in red, resulting RLE sparse level set defined isosurface in blue. Conversion resolution was 624x554x488.

3.1 Mesh to RLE Level Set. The structured ray casting method of converting meshes to dense level sets introduced in [Houston et al. 2003] can be adapted to RLE sparse level sets such that neither $O(n^3)$ intermediate storage nor $O(n^3)$ operations is required. This efficiency results from the ray cast operator's ability to provide upper bound signed distances for large linear sections of a volume at once. While structured low density ray casting doesn't yield an exact distance field solution by itself, the level set values of the identified narrow band can be improved to sub-voxel accuracy via slower but exact method as a post process.

3.2 Self-Intersection. In order to deal with self-intersections, which occur frequently with traditionally animated characters, we employ the following disambiguation scheme: Each ray cast which intersects a grid point results in an inside or outside vote as well as a corresponding signed distance value. After a number of passes, the distance corresponding to the sign with the most votes will be resolved as the final level set value.

3.3 Temporal Anti-Aliasing. Animated characters often have sub-voxel features such as sharp folds or edges, which when animated across a regular grid structure can result in visually displeasing aliasing artifacts. To remove these artifacts, without artist intervention or accommodation, meshes can be converted to RLE sparse level sets at twice the desired resolution followed by down sampling with a low pass filter to the desired resolution.

4 Results

A plug-in for 3dsmax has been developed that allows for artists to easily make use of RLE sparse level set operations in their every day work flow. Additionally, both the RLE sparse level set representation and the above described method of converting animated characters into level sets is used in the creation of the "Tar Monster" production effect described in [Wiebe & Houston 2004].

References

BRIDSON, R. 2003. Computational aspects of dynamic surfaces. PhD Thesis. Stanford University.

CURLESS, B., & LEVOY, M. 1996. A volumetric method for building complex models from range images. Computer Graphics 30, Annual Conference Series, 303-312.

HOUSTON, B., BOND, C., & WIEBE, M. 2003. A unified approach for modeling complex occlusions in fluid simulations. SIGGRAPH 2003 Technical Sketch Presentation.

WIEBE, M., & B. HOUSTON. 2004. The Tar Monster: Creating a Character with Fluid Simulation. SIGGRAPH 2004 Technical Sketch Presentation.