Parallelizing the Particle Level Set Method

Wen Zheng
Stanford University

Introduction
The particle level set is widely used as an accurate interface tracking tool in simulation, computer vision and other related fields. However, high computation cost prevents applying this method to real-time and interactive scenarios. This work intensively used parallel design patterns that are implemented in the thrust library, like compaction, reduction and scattering, to parallelize the particle level set method in order to attain real-time performance.

Outline
The flowchart shows the main steps of the particle level set method. The steps in red boxes are non-trivial to be parallelized, which will be discussed here in detail.

Particle seeding
- Compact cell indices within the thickness of the interface
  `thrust::remove_if`
- Randomly generate particle positions as in the Monte Carlo example (each cell has a unique seed)
- Compact and remove those failed seeding
  `thrust::remove_if`

Level set correction
- Compact particle indices that indicate errors in the level set
  `thrust::remove_if`
- Duplicate each particles into 4 cells near it, and assign the cell indices as their keys
- Collect particles with the same key together
  `thrust::sort_by_key`
- Compute a correction value for each duplicated particle
- Attain the maximum correction value per each cell
  `thrust::reduce_by_key`

Results

Comparison of Rigid Rotation Test
A notched disk rotates as a rigid body. The top row used the particle level set method; the bottom row used the pure level set method.

Timing Info
Resolution: 1024x1024

<table>
<thead>
<tr>
<th></th>
<th>Serial</th>
<th>Parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Marching</td>
<td>128.621 ms/frame</td>
<td>9.214 ms/frame</td>
</tr>
<tr>
<td>Particle Correction</td>
<td>348.315 ms/frame</td>
<td>4.332 ms/frame</td>
</tr>
</tbody>
</table>