

Fluid Simulation on the GPU using OpenCL

Tal Stramer
tstramer@stanford.edu

December 15, 2016

1 Summary

The goal of my project was to build a production-quality fluid simulation on the GPU using OpenCL. The hope was that I would be able to simulate many more particles and build a more realistic fluid simulation by moving computation to the GPU.

My specific accomplishments were to parallelize the major parts of the PBF algorithm, specifically:

- Particle computations: For every place in the algorithm that starts with ‘for all particles’ (line 1, 5, 9, 12, 16, and 20), I used the GPU to evaluate the computation for each particle in parallel.
- Nearest-neighbor search: I created a parallel version of the particle binning algorithm by first computing each particle’s bin based on its position in parallel. Then, using a separate thread per bin, I looped over the particles and assigned particles to bins. Finally, in parallel I assigned each particle its neighbors based on neighboring bins.

The code is written in C++ and OpenCL. I also used openFrameworks, a C++ framework that provides many useful libraries providing abstractions over OpenGL / OpenCL / etc for more easily writing graphics code.

2 Results

The resulting improvement in efficiency over my non-parallel implementation from project 1 is considerable. For project 1, the maximum number of particles I could render on my Macbook pro was around 4,000. With the parallel implementation, I am able to render up to 1,000,000 particles. After this, I run out of memory on my graphics card. Below are specific timings:

Particles	FPS (max 60)
1,000	60
5,000	55
10,000	50
50,000	9
100,000	4
500,000	.7
1,000,000	.2

One of the biggest bottle-necks in the algorithm is the nearest-neighbor search. I originally had planned to implement fast fixed-radius nearest neighbor search but ran out of time.

3 References

MACKLIN, MILES, AND MATTHIAS MULLER. "POSITION BASED FLUIDS." ACM TRANSACTIONS ON GRAPHICS (TOG) 32.4 (2013): 104.