Abstract
Most fractal generation software uses shortcuts and optimizations to run efficiently on a CPU. However with the rising power of graphics processors, the calculation and display of fractals would be much more easily done on a graphics card. The work is to start applying the shortcuts and functionality of free fractal software to code that runs on the GPU using the CUDA programming language.

Fractal Creation
A fractal is generally an image that exhibits self similarity. The fractals that we are concerned with are generated using recurrence relations. Some complex number $z(n)$ is calculated using $z(n)$ as well as whatever point on the complex plane we are currently looking at. If $z$ stays within a certain bound after a set number of iterations, it is considered to be a part of the set. If it breaks out of the bound, then it is not part of the set. These are called escape time fractals.

Each point can be tested independently of all of the other points on the complex plane. Thus we can hand off each point to a different processor. On the GPU each point would be tested sequentially. A GPU should be much faster because it can do many points at once.

Timings
Test Machine: XPS15 running Windows 7 with a GT540m graphics card (96 CUDA cores) with an i7 2720qm processor (2.2GHz with boost up to 3.3GHz) and 8GB of RAM.

Optimizations with dimension 1024x1024, 200 iterations and blocks of 16x16 threads:
• 30 fps without optimizations
• 33 fps with cardioid and period 2 bulb checks
• 22 fps with periodicity checking

Runtime with CPU version of software with dimension 1024x1024, 200 iterations and cardioid and period 2 bulb checks:
• 3.8 fps without optimizations
• 12.8 fps with cardioid and period 2 bulb check
• 4.5 fps with periodicity checking

Moving Forward
• Optimize threads per block
• Implement better periodicity checking
• Implement other shortcuts
  • Solid guessing
  • Dynamic resolution

Acknowledgements
• NSF grant 1156509
• Dr. Enyue Lu
• NVIDIA and CUDA

References
• http://locklessinc.com/articles/mandlebrot/
• CUDA By Example