Mandelbulb with CUDA

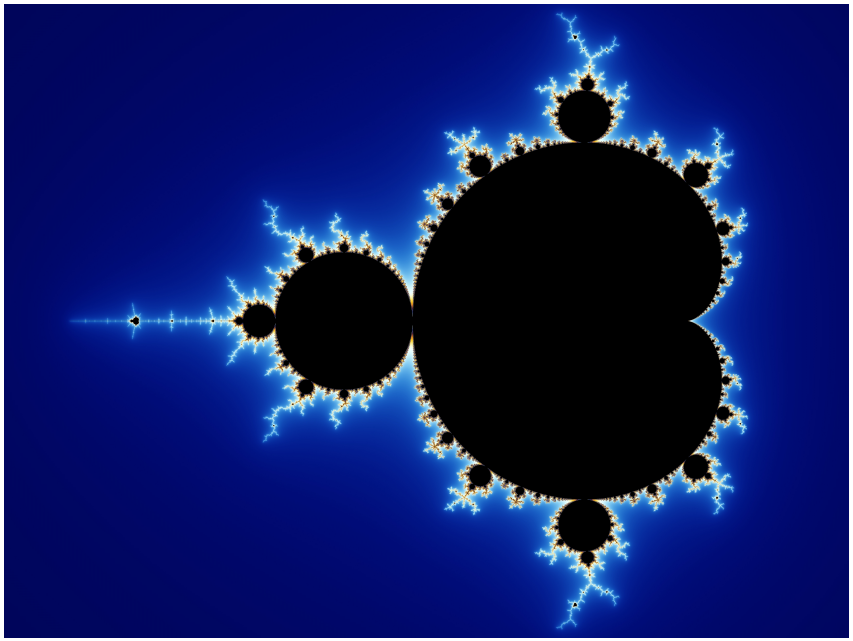
Alexander Kuczala

June 11, 2015

Outline

- Mandelbrot set
- Mandelbulb set
- Ray marching algorithm
- Parallelization
- Performance

Mandelbrot Set



$$z_{i+1} = z_i^2 + c$$

$$z = x + iy$$

Accept if $|z|^2 = x^2 + y^2 < r^2$ after n iterations

3D analogue?

Polar coordinates

$$z = re^{i\phi} = r\cos(\phi) + ir\sin(\phi)$$

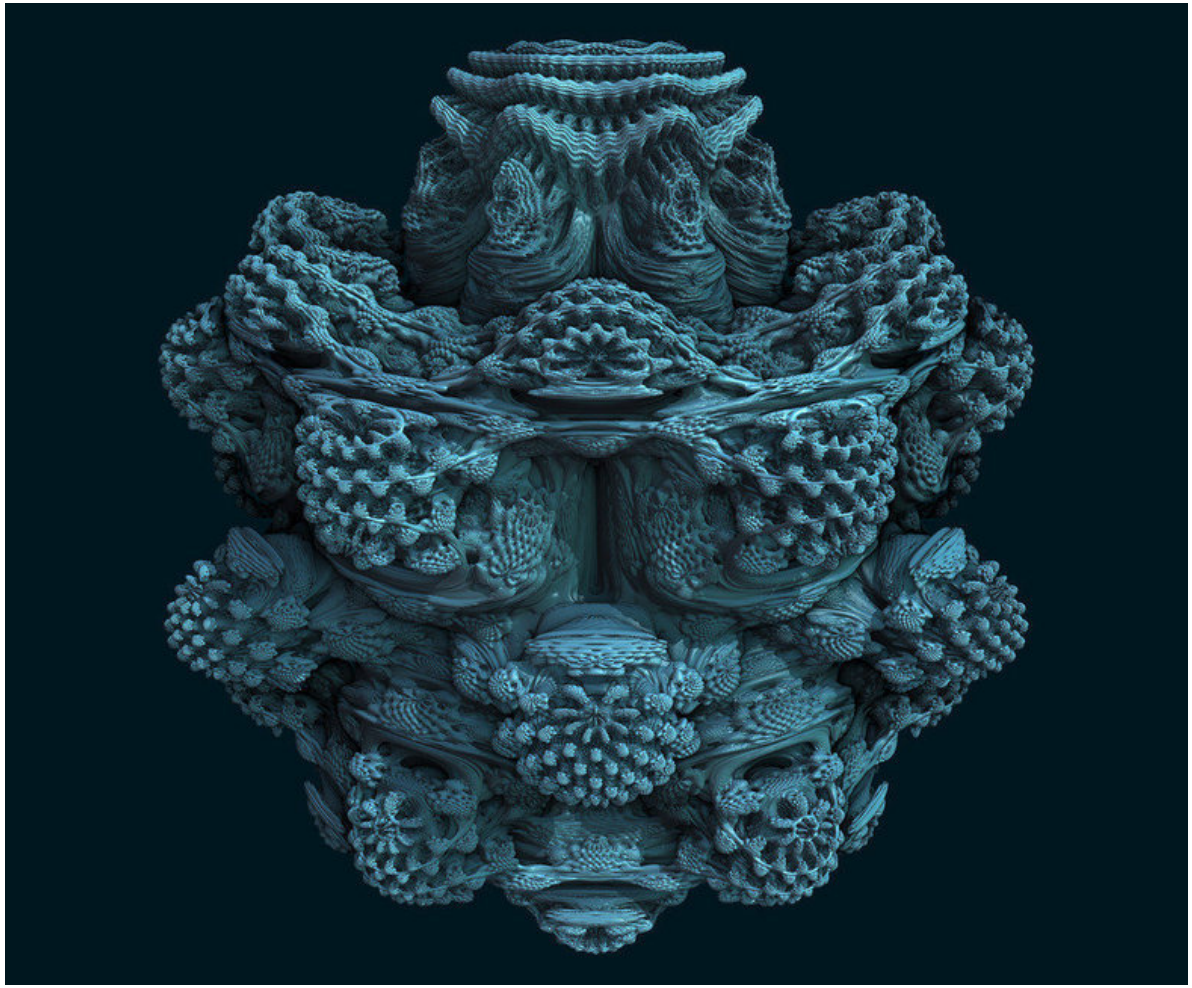
$$re^{i\phi} \rightarrow r^2 e^{2i\phi} + c = (r\cos(2\phi) + a) + (ir\sin(2\phi) + b)$$

Spherical coordinates

$$\zeta = r(\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$$

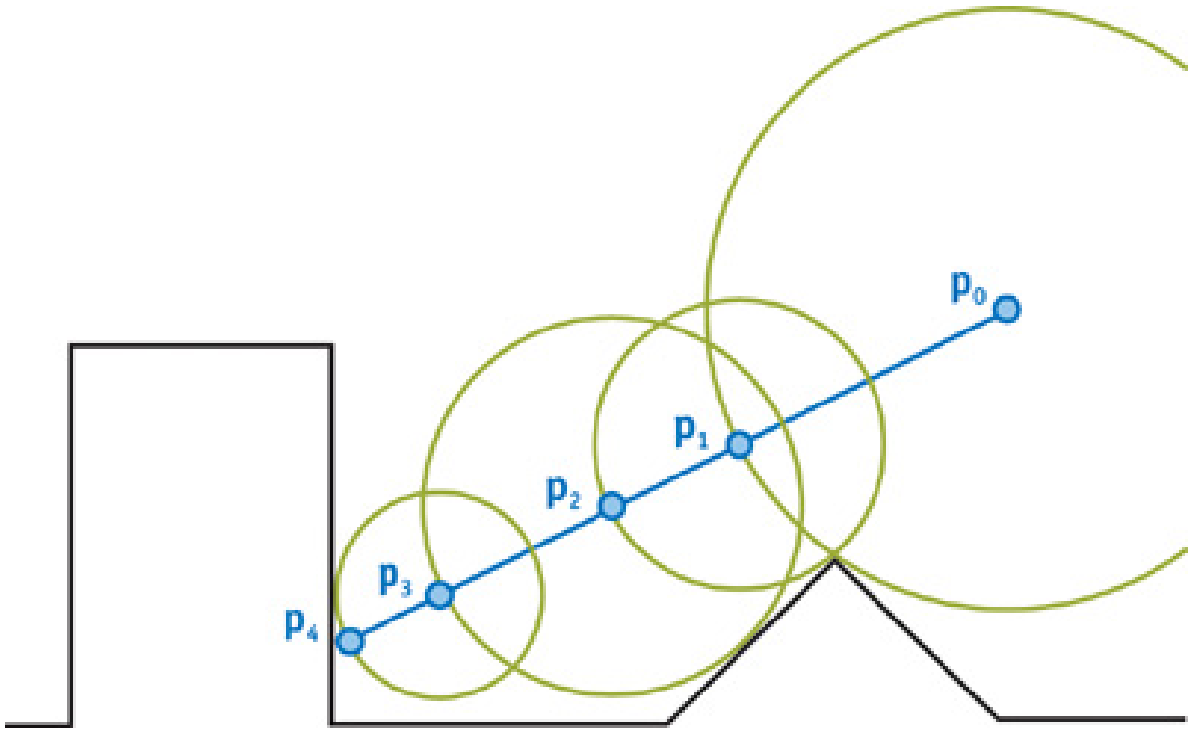
$$\rightarrow r^2(\cos 2\phi\sin 2\theta, \sin 2\phi\sin 2\theta, \cos 2\theta) + \vec{c}$$

Mandelbulb



$$\zeta(r, \phi, \theta) \rightarrow r^p(\cos p\phi \sin p\theta, \sin p\phi \sin p\theta, \cos p\theta) + \vec{c}$$

How to render?



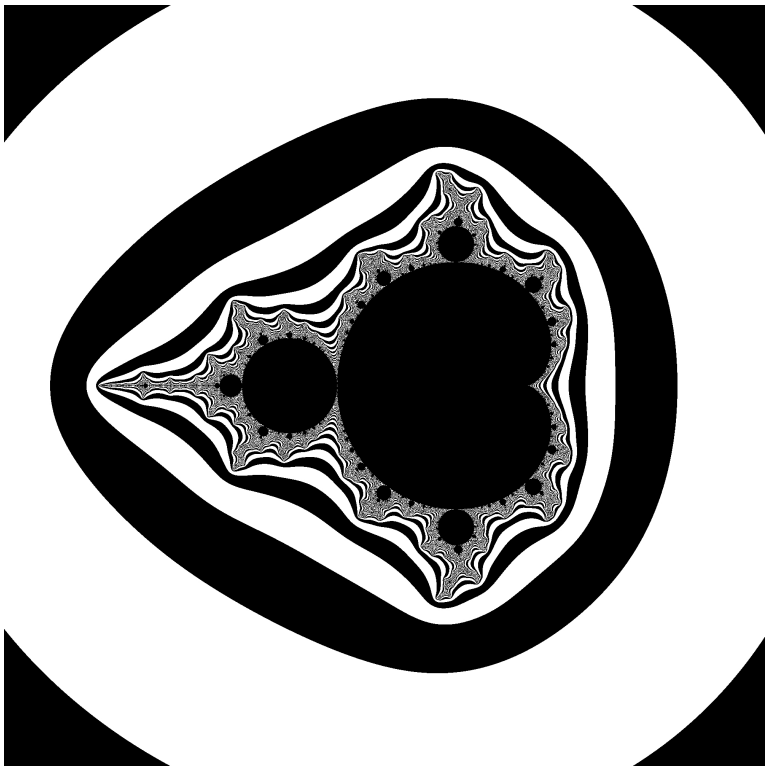
- Need to find distance to fractal along each ray
- Use ray marching algorithm
- Need to estimate distance to fractal at each step to bound marching distance

Equipotentials

- Continuous iteration function

$$\phi(z) = \lim_{n->\infty} \frac{\log |z|}{2^n}$$

- Integer equipotentials of function are correspond to iterations of set



Distance estimator

- Distance estimator

$$d(z) = \frac{\phi(z)}{\phi'(z)}$$

$$d(z) \approx \frac{1}{2} \frac{r(z)}{r'(z)} \log r(z)$$

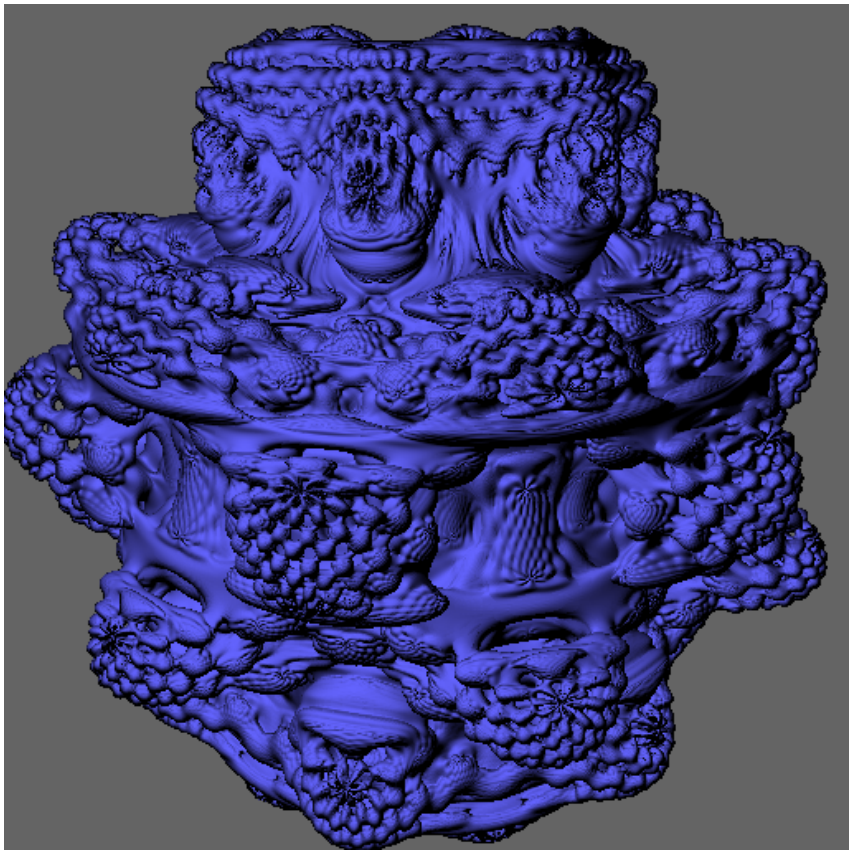
- Exact for $n \to \infty$, accurate for z near boundary
- Derivative of iteration function

$$z'_n(c) = 2z'_{n-1}(c)z_{n-1}(c) + 1$$

$$dr_n = 2dr_{n-1}r_{n-1} + 1$$

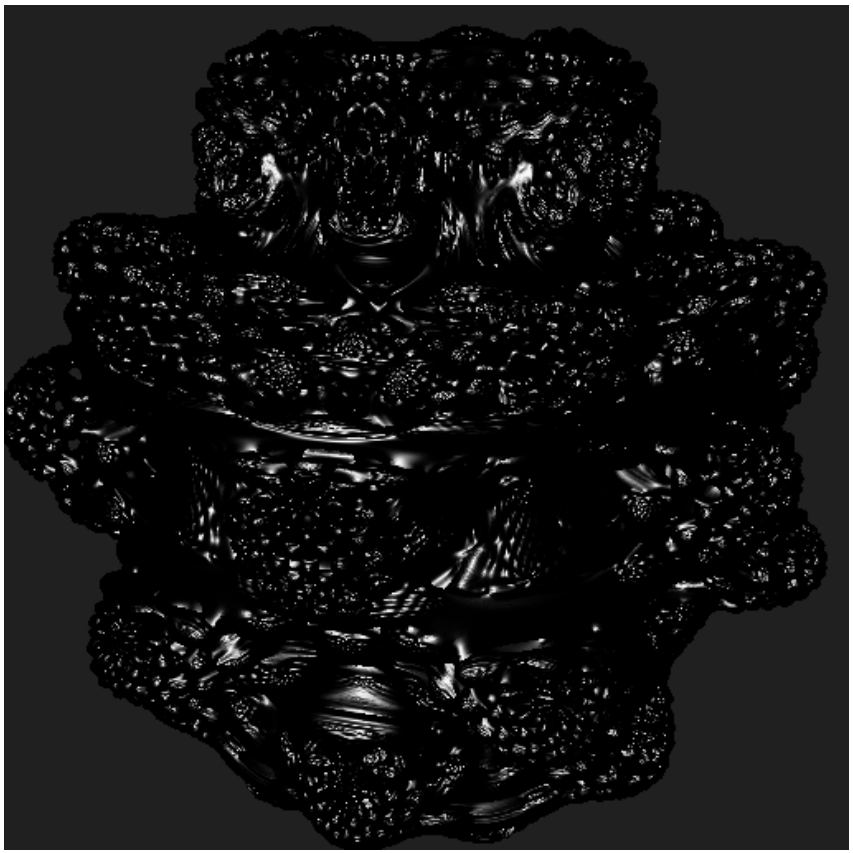
- Exact for mandelbrot
- Somehow also works for mandelbulb

Lambert Shading



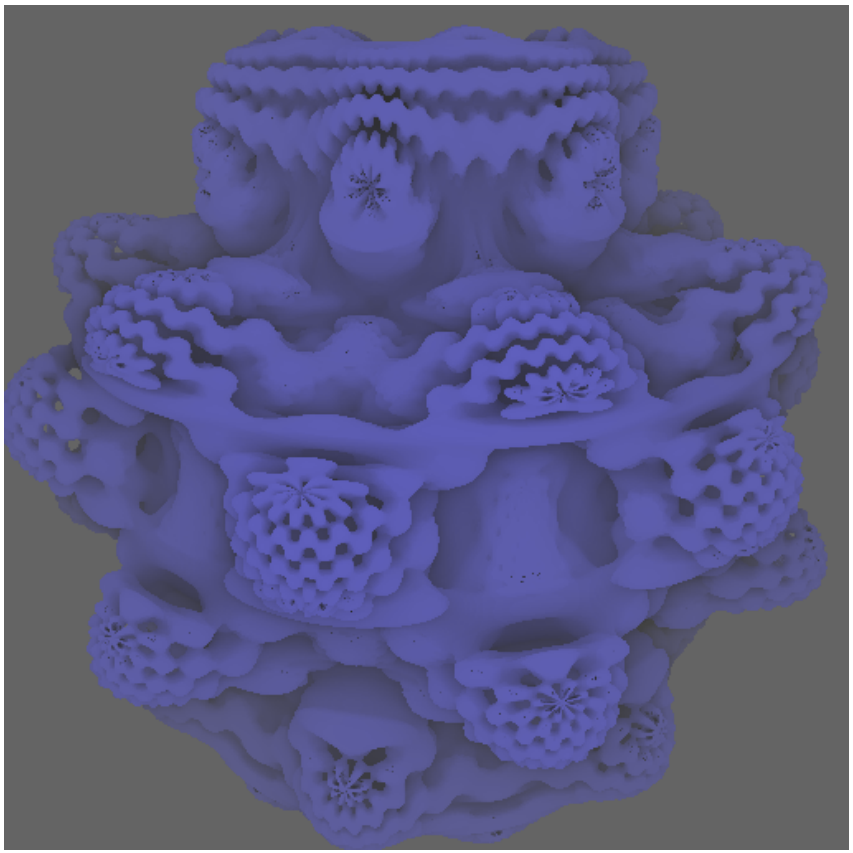
- Approximate normals by finite difference
- Lambert shading - shade using angle between light and normal

Phong Specularity



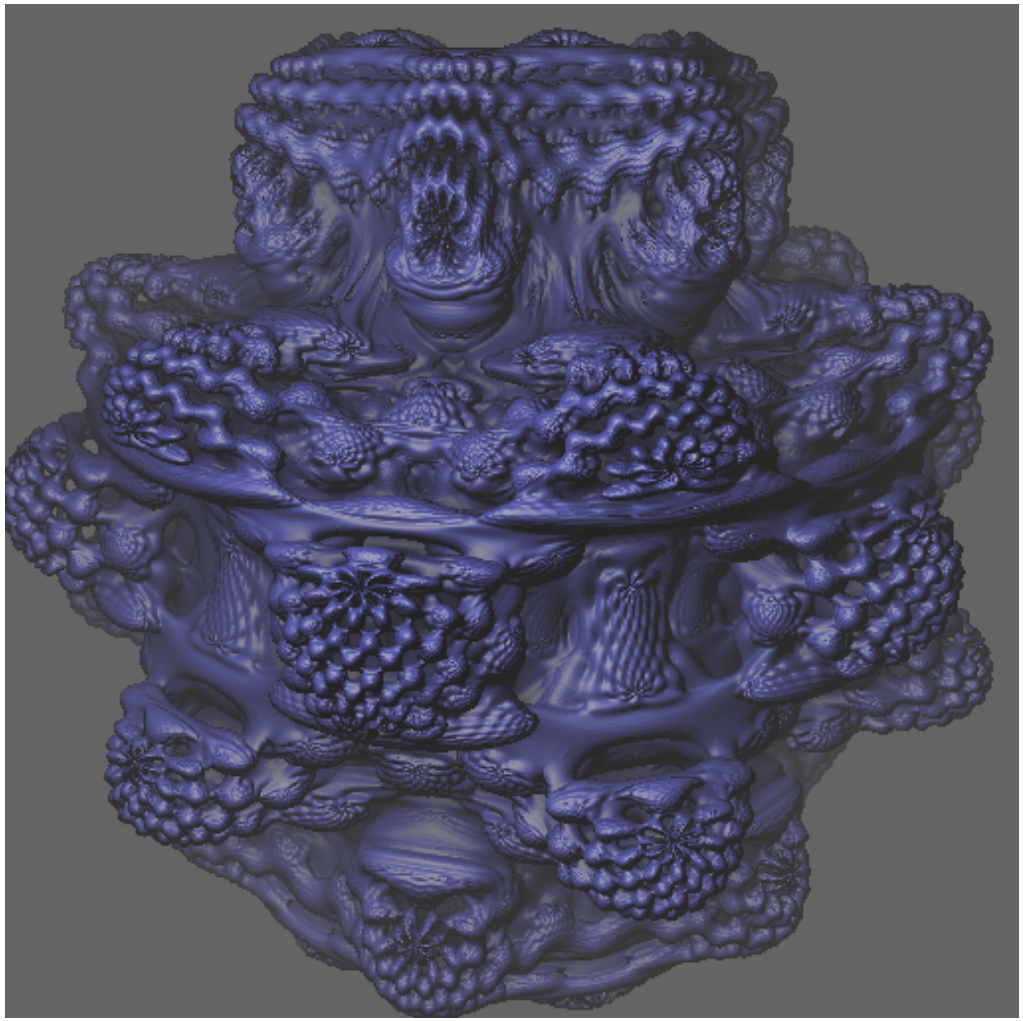
- Depends on angle between reflected light and ray

Ambient Occlusion



- Darken areas that take more ray steps to reach

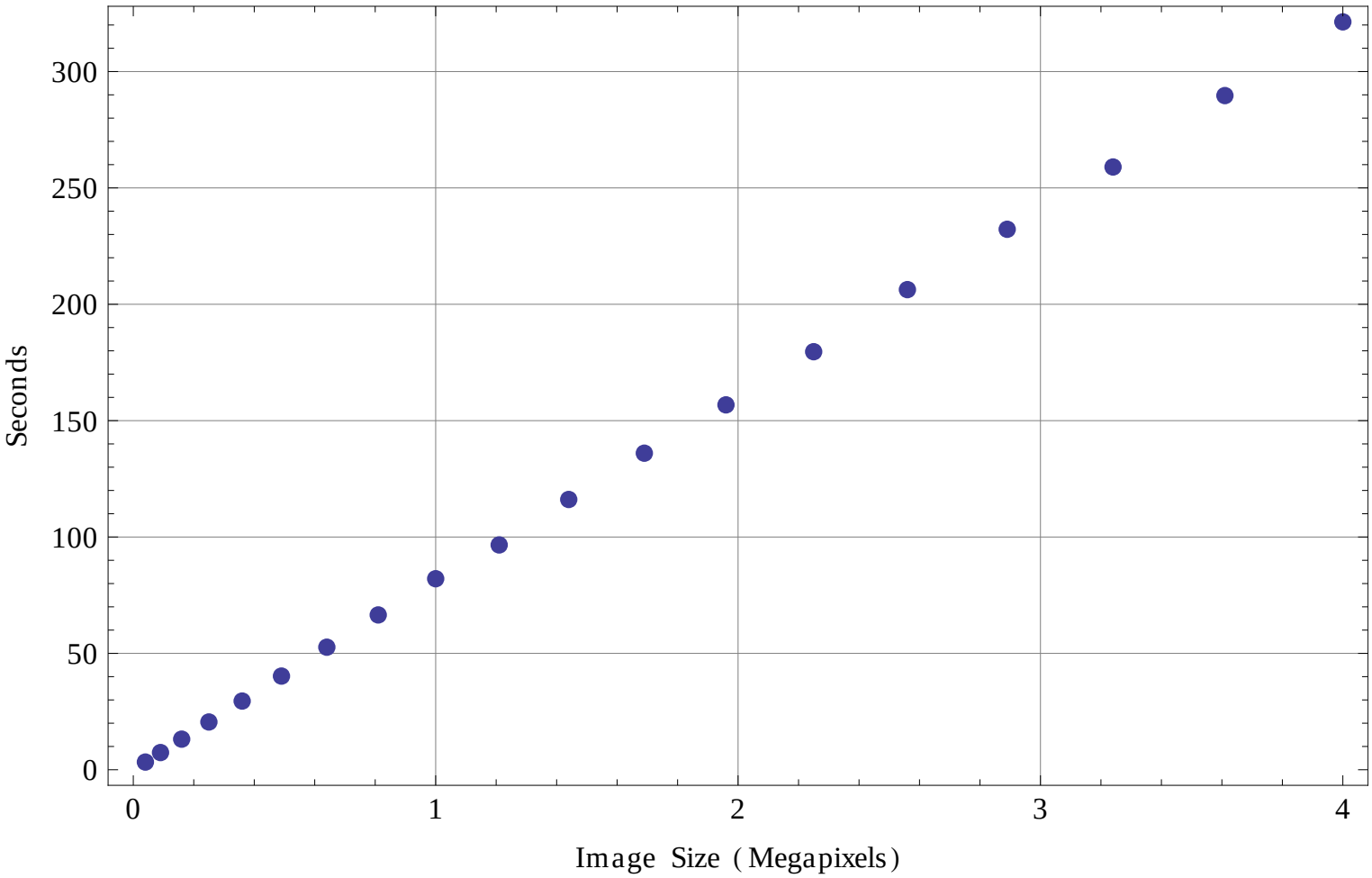
Combined + fog



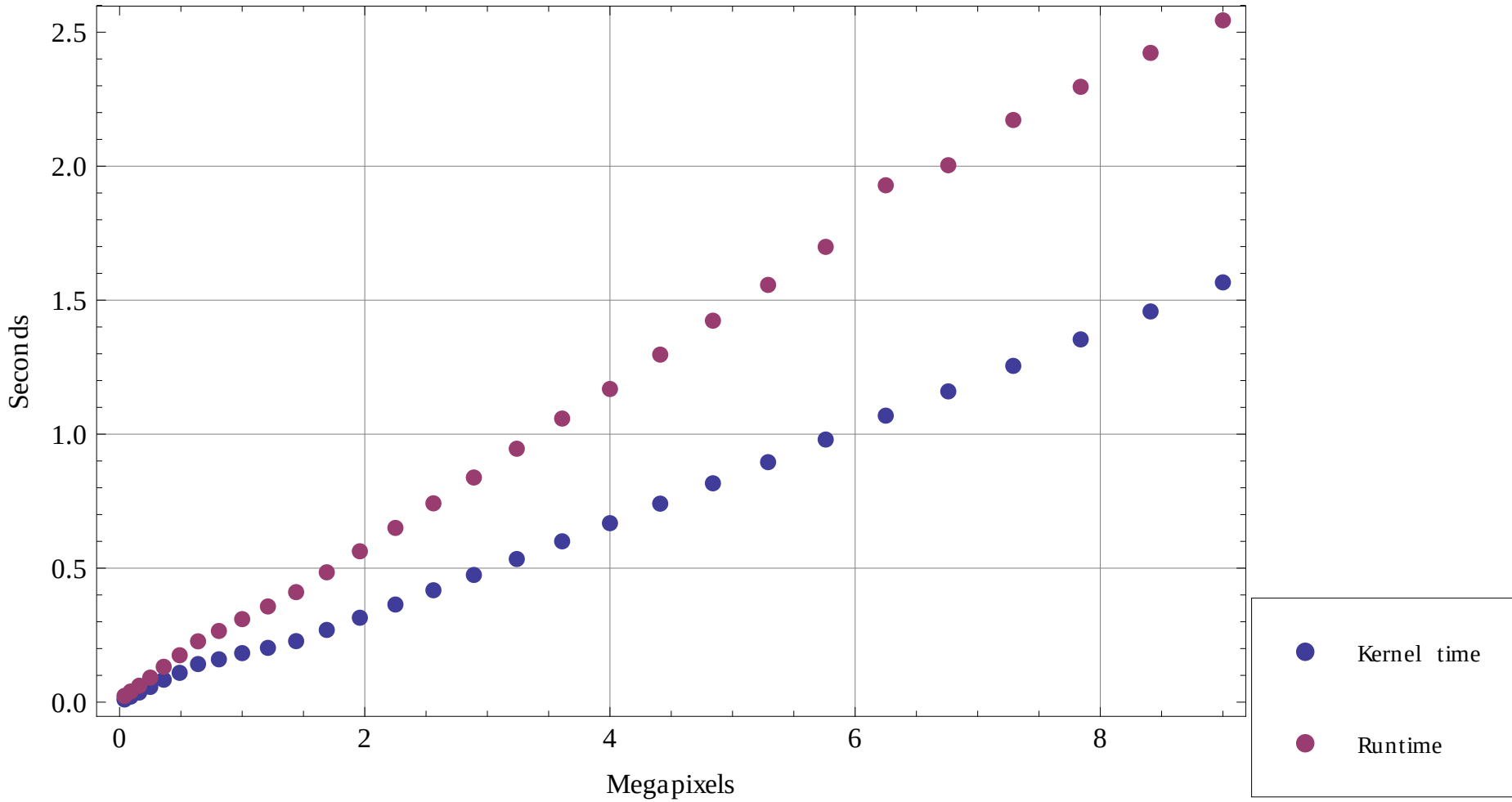
Parallelization with CUDA

- One ray per pixel, one processor per ray
- Each processor computes ray direction from pixel coordinates
- Same computation for $\sim 10^6$ pixels
- All work done on GPU

Serial program runtime

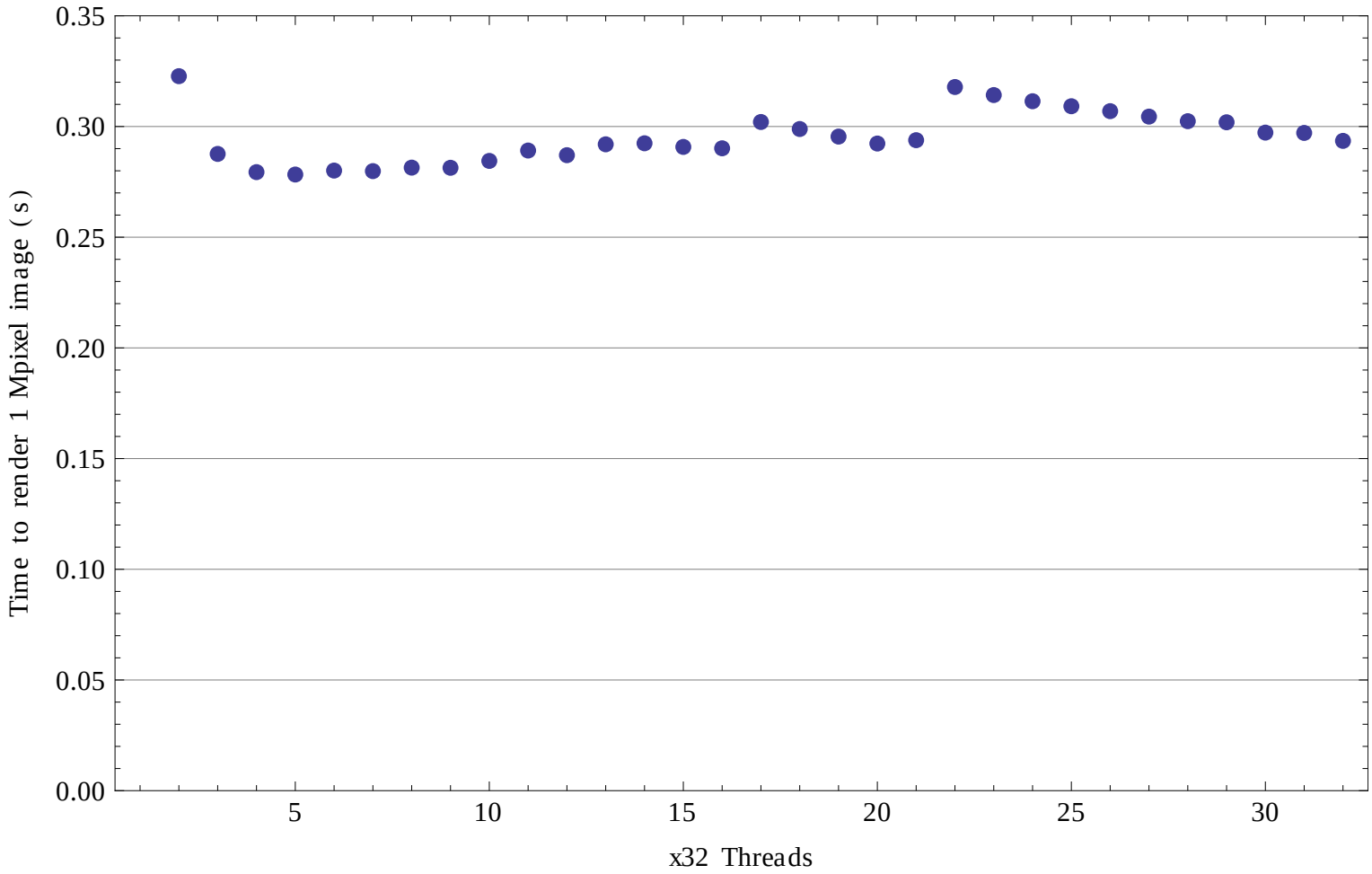


CUDA runtime

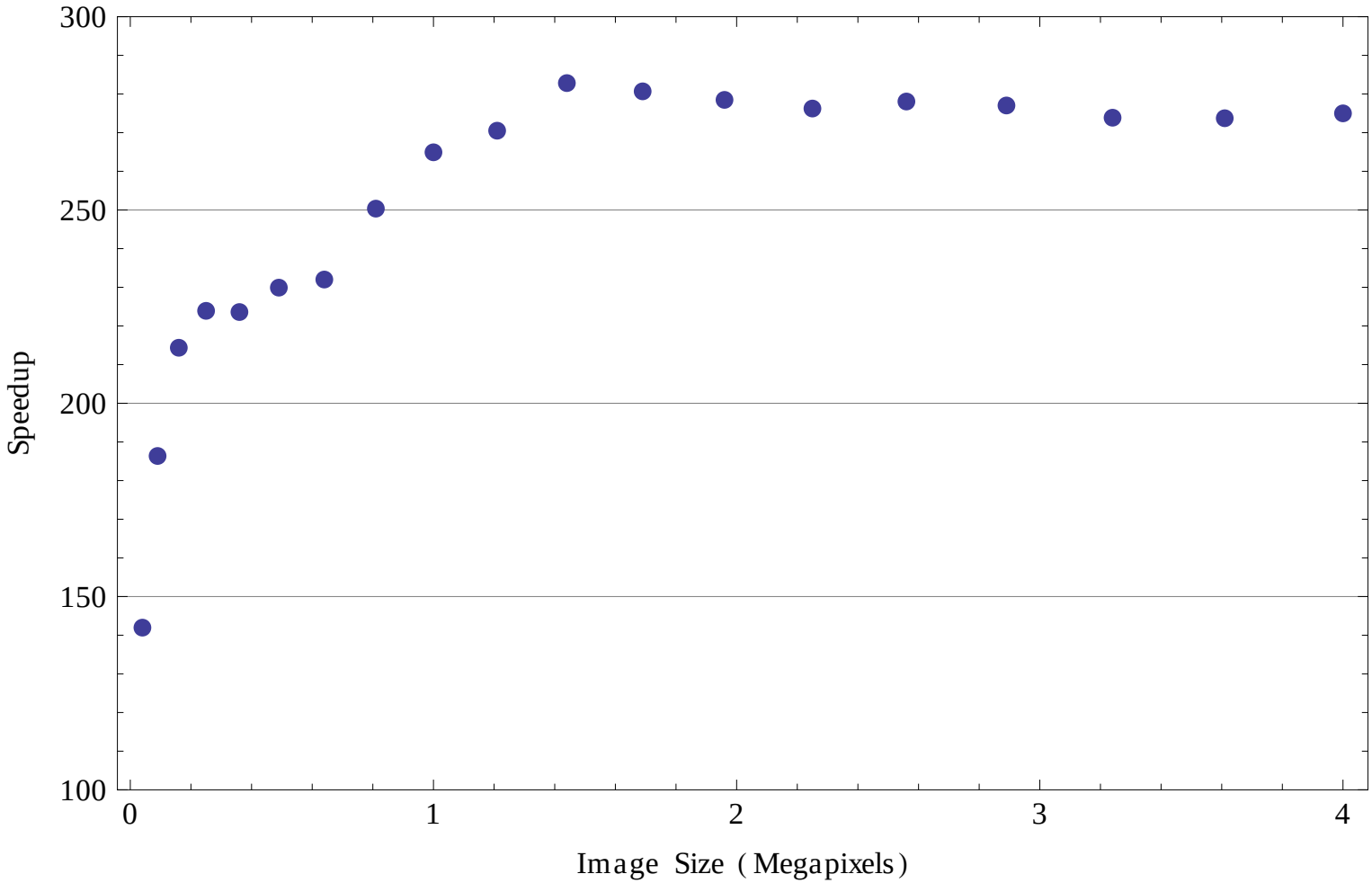


256 threads per block

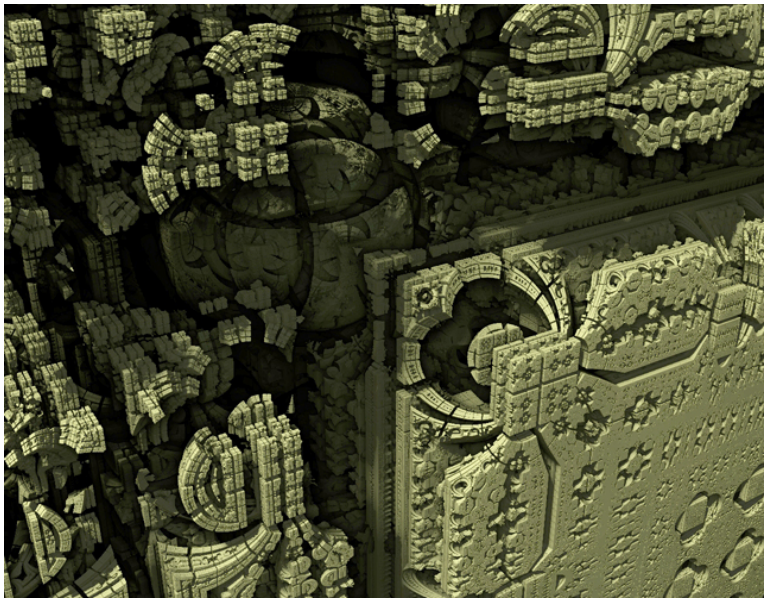
CUDA runtime



Speedup



Future work



- Generalize to other 3D fractals
- Double precision
- Interactive rendering

References

- Mikael Hvidtfeldt Christensen, Syntopia blog
  blog.hvidtfeldts.net/index.php/2011/06/distance-estimated-3d-fractals-part-i/
- Tom Beddard, subblue blog
  2008.sub.blue/blog/2009/12/13/mandelbulb.html