Parallel Ray Tracing for General Optical Systems

CDR Charles B. Cameron

23 October 2006

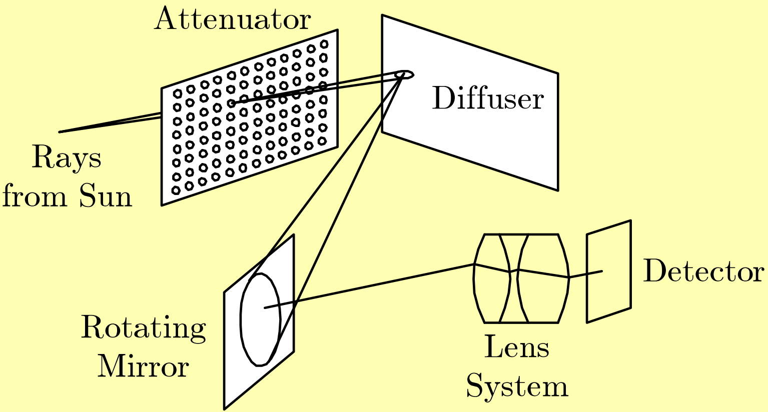
Introduction

- MODIS

  Moderate Resolution Imaging Spectroradiometer

- Multiple DSPs
- Cray XD1
- Pipelined design using FPGAs

MODIS


Attenuator
Diffuser
Rays
from Sun
Detector
Rotating
Mirror
Lens
System

Optical Surfaces in MODIS

Type	**Shape**	**Number**
Transmissive	Planar	13
Reflective	Planar	5
	Spherical	
	Conicoidal	2
	Non-conicoidal	
Refractive	Planar	3
	Spherical	4
	Conicoidal	1
	Non-conicoidal	
Image plane		1
Total number of surfaces		29

MODIS Parameters

- 1 951 pinholes
  (Rays traced from 485 of them.)
- Rays from each pinhole:
  20 rows $\times$ 20 columns $= 400$ rays
- Rays from diffuser:
  241 rows $\times$ 121 columns $= 29\,161$ rays
- Total rays traced:
  5.66×10^9 rays

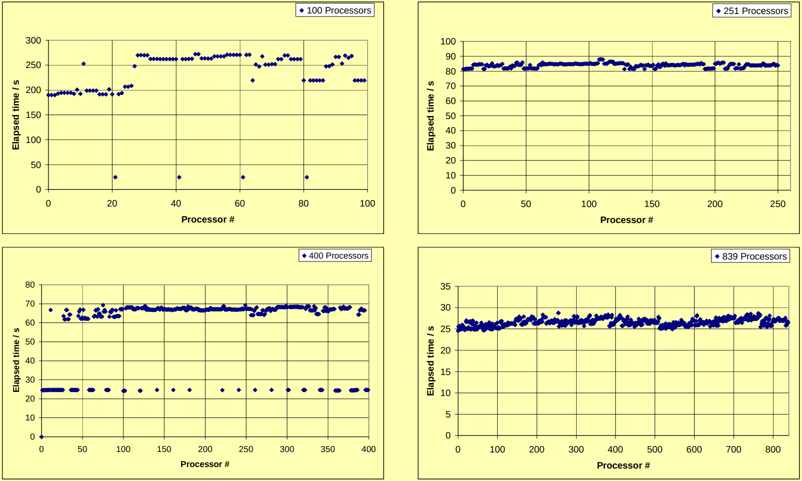
Ray Tracing

- Intersection
- Interaction
- Coordinate Transformation
- Aperture

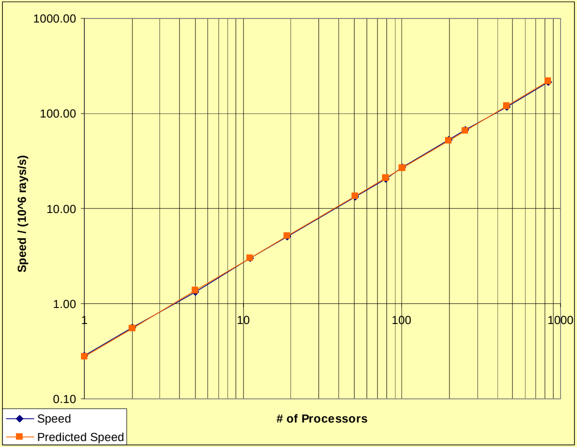
Earlier Simulations

Uniprocessor	DEC Alpha computer	≈ 2 weeks
DSP	One ADSP21160	284.1 hours
	Eight ADSP21160	35.6 hours
Multiprocessor	Cray XD-1, 839 processors	< 27 s

Ray Tracing Using the Cray XD1



Results for the Cray XD1



$$s = (278000 \pm 3500)n^{0.9891} \text{ rays/s}$$

FPGA

- Xilinx Virtex 4 xc4vfx20-ff672
  - Pipelined design
  - 12 intersections at once
  - 29 distinct states
  - 348 steps per calculation
- but …Cray XD1 uses Virtex II Pro (mostly)

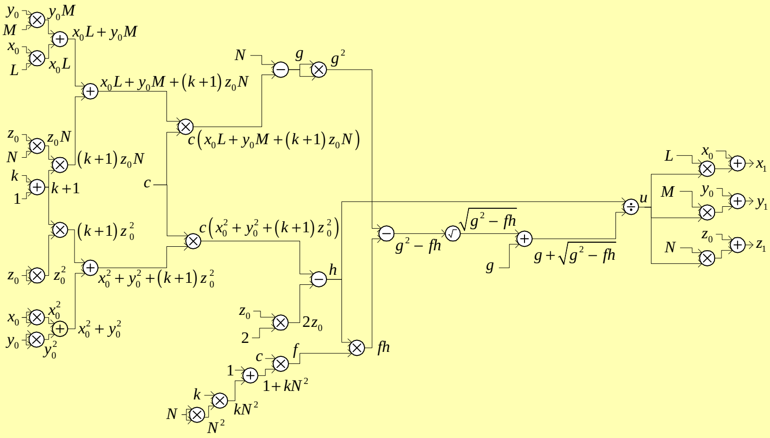
Intersection Calculations

Surface	$\pm$	$\times$	$\div$	$\sqrt{\ }$
Plane	3	3	1	0
Sphere	11	13	1	1
Conicoid	13	19	1	1
Aspheric	?	?	?	?

Conicoid Intersection Equations

$$
\begin{aligned}
f &= c\left(1 + kN^2\right), \\
g &= N - c\left(x_0 L + y_0 M + (1+k) z_0 N\right) \\
h &= c\left(x_0^2 + y_0^2 + (1+k) z_0^2\right) - 2z_0 \\
u &= \frac{h}{g + \sqrt{g^2 - fh}} \\
\boldsymbol{x}_1 &= \boldsymbol{x}_0 + u\hat{\boldsymbol{\omega}}_0
\end{aligned}
$$

Flow of the Calculations



Pipelining

- Pipelined floating point modules for $\pm$, $\times$, $\div$, $\sqrt{}$
- Chip too small for, say, 19 multipliers
- Choose desired level of pipelining, n
- Begin operations only at times jn
- Find total delay $N = kn$
- k should be prime to simplify logic
- Start next pipelined calculation at times mk

Results to Date

- Designed and simulated a pipelined version with $n = 12$, $N = 348$, $k = 29$
- Simulation suggests one intersection calculated every 290 ns
- Conservative estimate: exceeds time for other kinds of intersection, as well as reflection, refraction, coordinate-transformation, and aperture-checking calculations
- Would need $4 \times 290\,\text{ns} = 1.16\,\mu\text{s}$ per ray at each surface
- Opteron 275 time: 129 ns per ray at each surface

Areas for Further Investigation

- Replace reciprocation with iteration to find $y = 1/d$:

$$y_{i+1} = y_i(2 - y_i d).$$

- Use a following multiplication to get division:

$$n/d = n \times (1/d)$$

- Replace square root with iteration to find $y = 1/\sqrt{c}$:

$$x_{i+1} = \frac{x_i \left(3 - x_i^2 c\right)}{2}$$

- Use a following multiplication to get square root:

$$\sqrt{c} = c \times \left(1/\sqrt{c}\right)$$

Other Issues

- Multiplication takes
  - Nine cycles on the Virtex 4
  - Six cycles on the Virtex II Pro
- Algorithm needed to generate hardware schedule
- Cray's Virtex II Pro bigger than board's Virtex 4
- Division of work between Opteron and FPGA
- Sharing of FPGA within one Cray node