Ray Tracer Rendering

SHIR AHARON
California Institute of Technology

ACM 114 Project

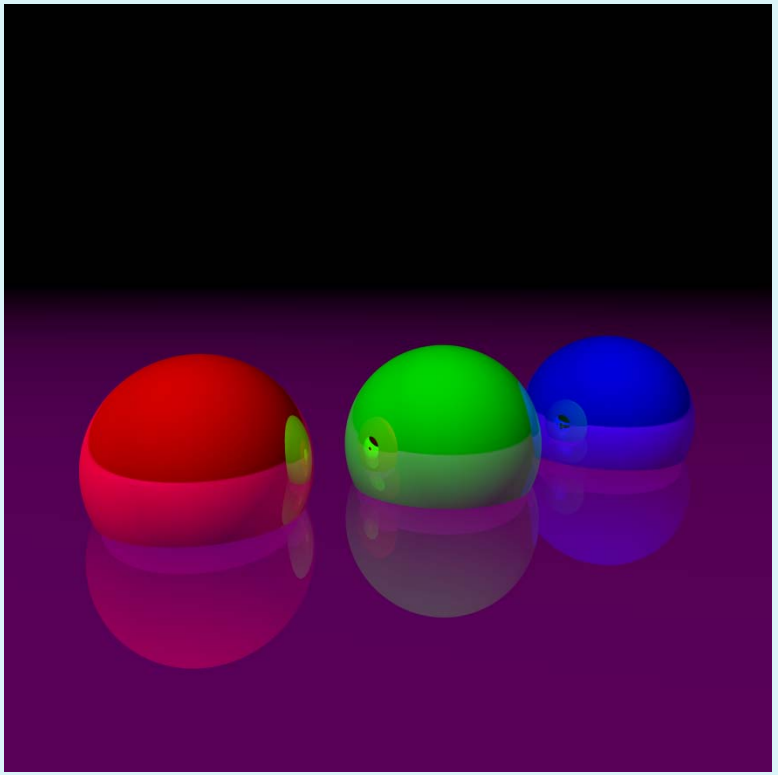
Scene Language

```
// Define the camera for the scene
camera  (-1.5, 1, 3) (-0.3, 0.5, 0) (0, 1, 0)

// Define the lights for the scene
light   (-10, 10, 5) [0.8, 0.8, 0.8]
light   (  5,  3, 5) [0.3, 0.3, 0.3]

// Define the planes for the scene
plane   0   (   0, 1  , 0) [0.5, 0.0, 0.5] .3

// Define the spheres for the scene
sphere  0.5 (-1.2, 0.2,  0.0) [  1, 0, 0  ] .7
sphere  0.5 (   0, 0.2,  0.0) [  0, 1, 0  ] .7
sphere  0.5 ( 1.2, 0.2, -0.2) [  0, 0, 1  ] .7
```



Parallelization Goals

- Decrease overall time of computation
  - Increase scene language capabilities (cylinders, reflections)
  - Increase scene complexity (more objects)
  - Increase scene size
- Render image in real time
  - As it is computed, not just output file

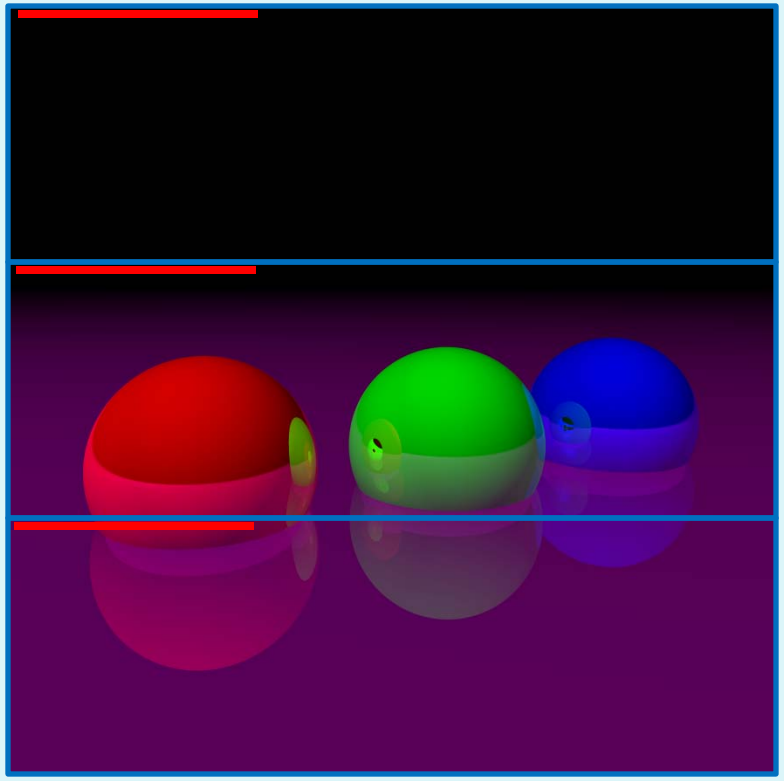
Parallelization Methods

- Distribute rectangular blocks of pixels
- Pass computed color to master
- Master can reassemble and display image

Limitations:

- Image library G2 is not parallelizable
- Color object to G2 ink slowest part cannot be parallelized

Parallelization Methods



Timing Analysis

- t_total
  - total run time for given parameter set
- t_rt
  - amount of time spent in the computation of the image (cuts out file writing time)
- t_wait
  - amount of time spent between the Irecv and Waitall calls.
- t_g2
  - amount of time spent in G2 calls to update the image.
- t_cc
  - amount of time spent in the G2 color conversion function.

Timing Results

- Block size has little effect
  - Tested 250, 500, and 1000 pixels
- Scales as square of size
- Simple image too simple for useful parallelization

No G2:

- Biggest gains from 2 to 3 and 3 to 4
- Smaller gains up to 7 (max used)

Timing Results

Table 4: g2 Analysis (complex image)							
BLOCK	proc	size	t_total	t_rt	t_wait	t_g2	t_cc
500	1	500	13.4	13.0		0.0	6.4
		1000	47.6	46.0		0.0	22.2
500	2	500	7.2	6.8	1.9	0.0	4.9
		1000	32.9	31.3	1.2	0.0	30.1
500	3	500	7.8	7.4	2.0	0.0	5.4
		1000	32.1	30.9	2.0	0.1	28.8
500	4	500	10.0	0.6	3.0	0.0	6.6
		1000	41.3	39.5	3.0	0.1	36.5
500	5	500	9.9	9.5	1.9	0.0	7.5
		1000	38.9	37.3	1.1	0.1	36.1
500	6	500	9.3	8.9	2.0	0.0	6.9
		1000	37.6	35.9	1.1	0.1	34.7
500	7	500	12.3	11.9	2.9	0.3	8.7
		1000	44.3	42.6	4.0	1.9	36.7

Conclusions / Future Work

- Parallelization matters more for larger and more complicated scenes
- Without g2 lots of overhead, but some gain
- With g2 limited by serial portion

- Switch to parallelizable graphics library
- Complete hybrid implementation template
  - Currently only splits rendering work on master (not usable with g2)