

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.189 Multicore Programming Primer, January (IAP) 2007

Please use the following citation format:

Natalia Chernenko, Michael D'Ambrosio, Scott Fisher, Brian Sweatt, R. J. Ryan, and Leevar Williams, *6.189 Multicore Programming Primer, January (IAP) 2007*. (Massachusetts Institute of Technology: MIT OpenCourseWare). <http://ocw.mit.edu> (accessed MM DD, YYYY).  
License: Creative Commons Attribution-Noncommercial-Share Alike.

Note: Please use the actual date you accessed this material in your citation.

For more information about citing these materials or our Terms of Use, visit:  
<http://ocw.mit.edu/terms>

# **6.189 IAP 2007**

---

## **Student Project Presentation**

### **Blue-Steel Ray Tracer**

blue-steel

# distributed *real-time* ray tracer

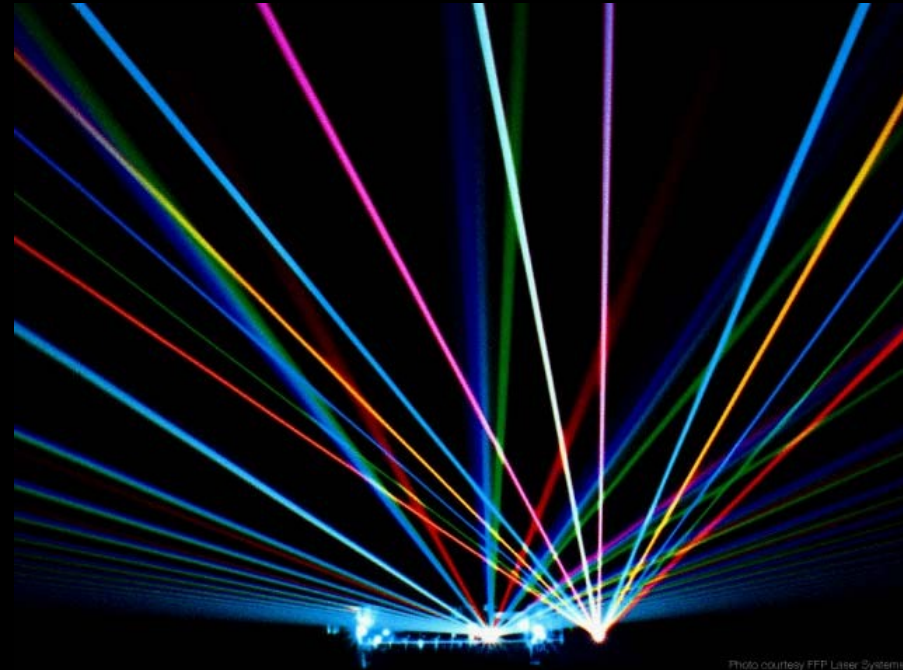


Photo courtesy FFP Laser Systems

# Ray Casting

---

For every pixel

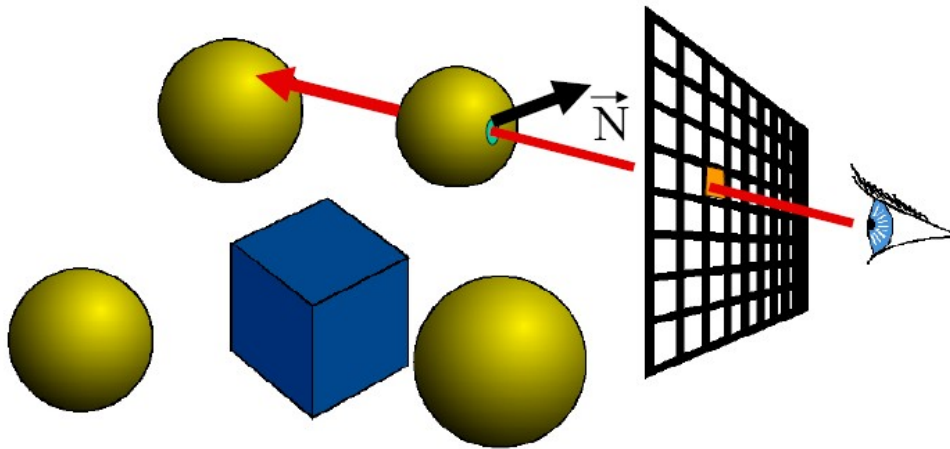
Construct a ray from the eye

For every object in the scene

**Find intersection with the ray**

Keep if closest

Shade depending on light and **normal** vector



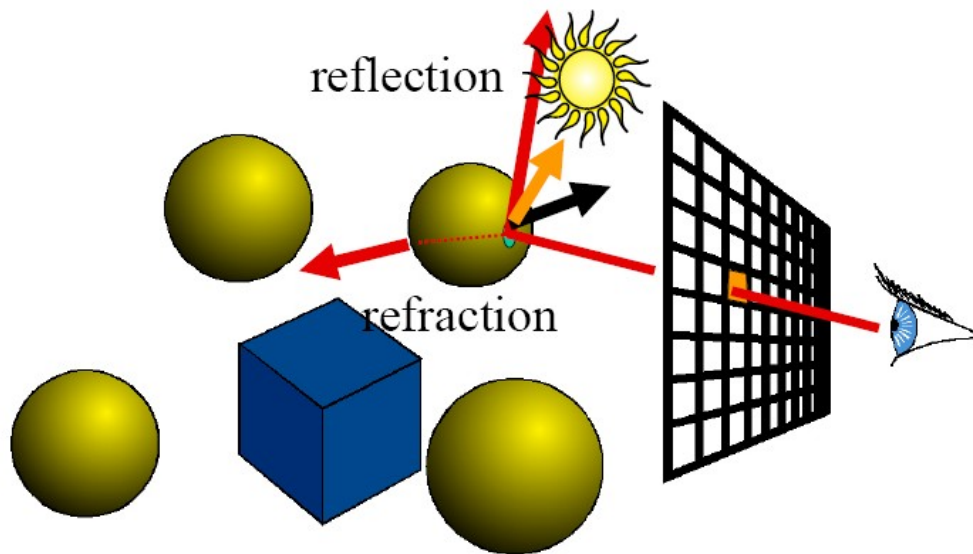
Finding the **intersection** and **normal** is the central part of ray casting

MIT EECS 6.837, Durand

# Ray Tracing

---

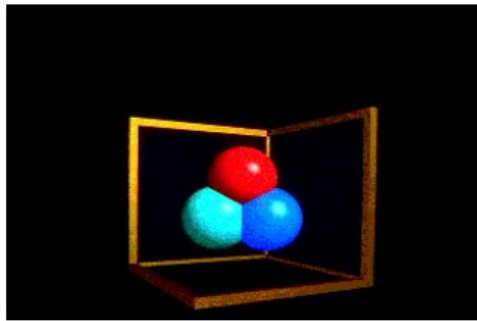
- Secondary rays (shadows, reflection, refraction)



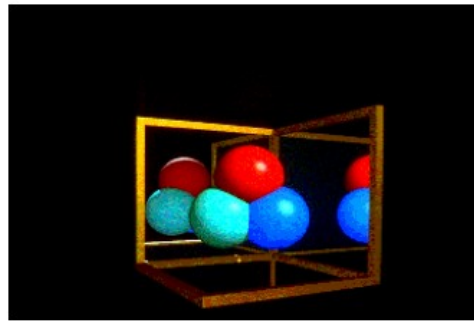
MIT EECS 6.837, Durand

# Recursion For Reflection

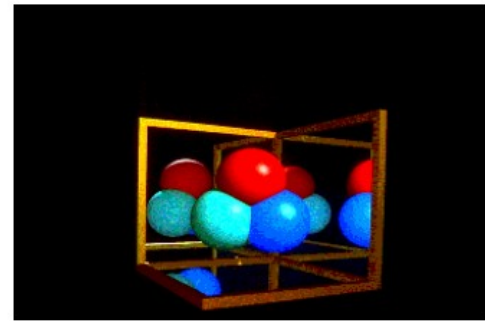
---



0 recursion



1 recursion



2 recursions

MIT EECS 6.837, Durand

Resolution

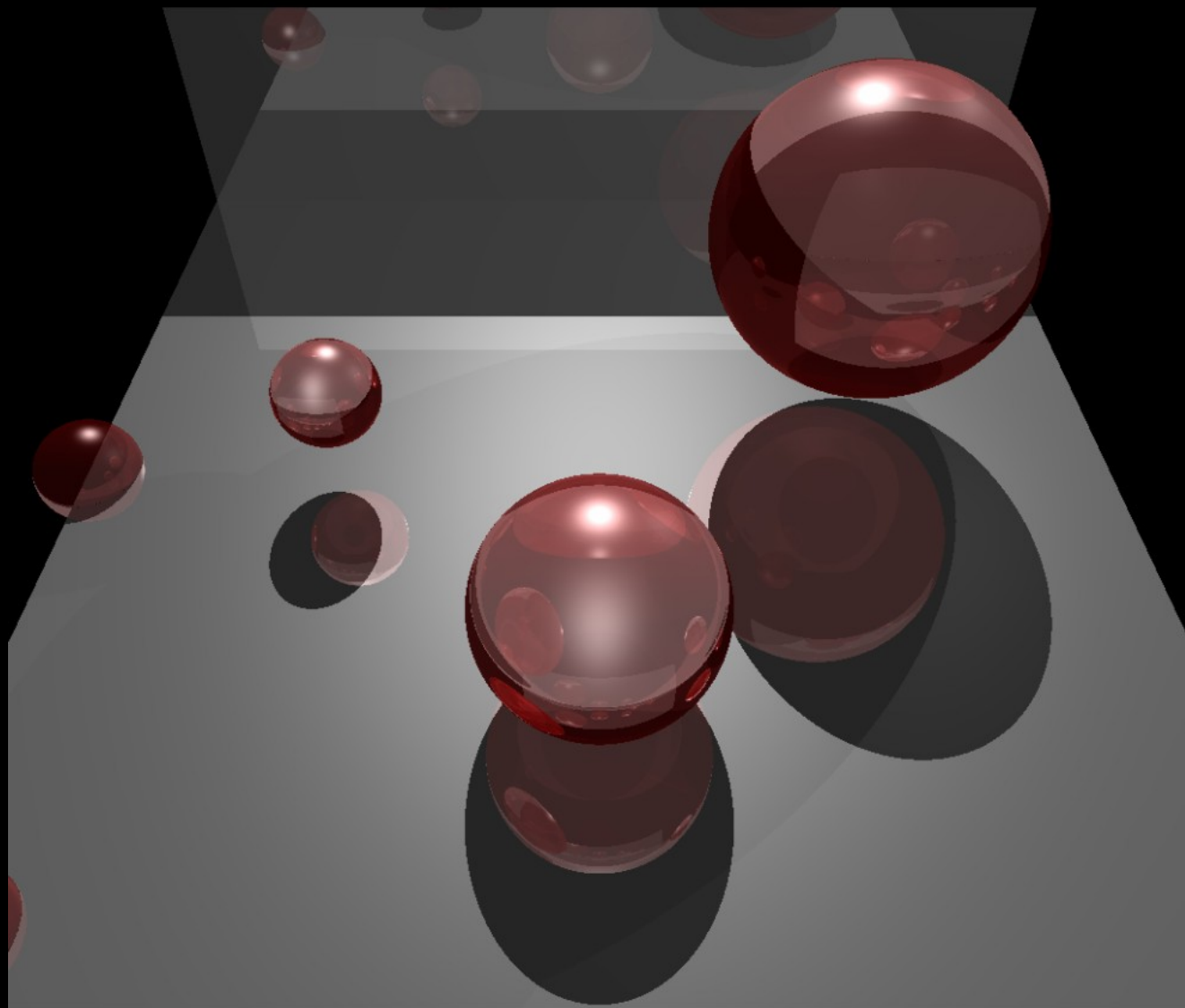
1024x1024

Recursion depth

100

Rendering time

2.6 seconds





# optimization

SIMD SIMD SIMD SIMD

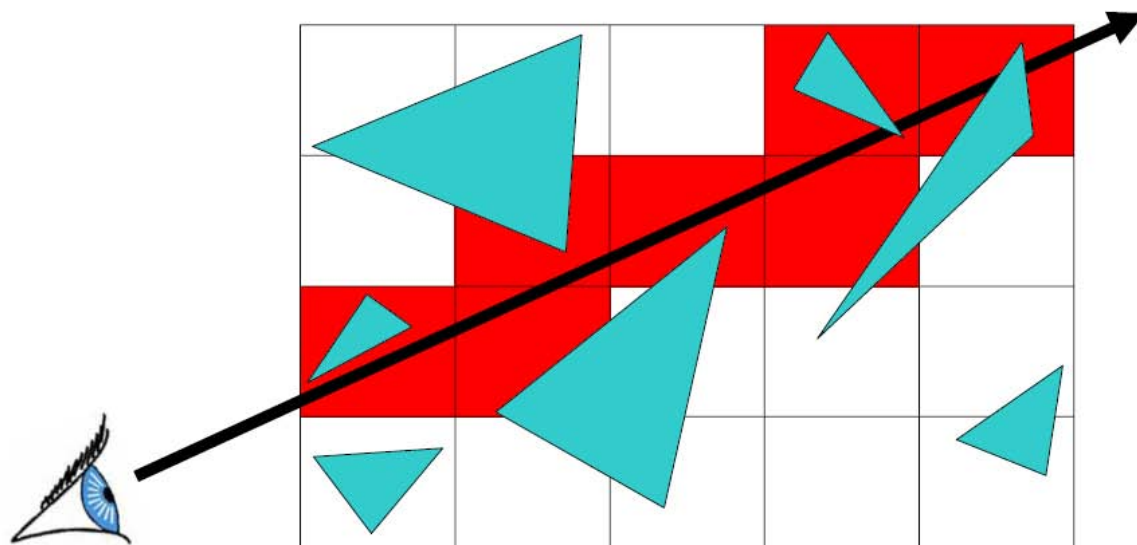
pipelining

no inheritance in rendering

```
void Box::intersectPacket(const RayPacket &r, HitPacket &h, float tmin) {  
    //vector float epsilon = spu_splats(0.0001f);  
    vector float zero = spu_splats(0.0f);  
    vector float max_v = spu_splats(MAX_FLOAT);  
  
    vector float tmin_v = spu_splats(tmin);  
  
    vector unsigned char splat_word_0 =  
        (vector unsigned char){0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3};  
    vector unsigned char splat_word_1 =  
        (vector unsigned char){4, 5, 6, 7, 4, 5, 6, 7, 4, 5, 6, 7, 4, 5, 6, 7};  
    vector unsigned char splat_word_2 =  
        (vector unsigned char){8, 9,10,11, 8, 9,10,11, 8, 9,10,11, 8, 9,10,11};  
  
    vector float minx = spu_shuffle(m_min, m_min, splat_word_0);  
    vector float maxx = spu_shuffle(m_max, m_max, splat_word_0);  
    vector float miny = spu_shuffle(m_min, m_min, splat_word_1);  
    vector float maxy = spu_shuffle(m_max, m_max, splat_word_1);  
    vector float minz = spu_shuffle(m_min, m_min, splat_word_2);  
    vector float maxz = spu_shuffle(m_max, m_max, splat_word_2);  
  
    vector unsigned int rdxgt0 = spu_cmpgt(r.dxc, zero);  
    vector unsigned int rdygt0 = spu_cmpgt(r.dyc, zero);  
    vector unsigned int rdzgt0 = spu_cmpgt(r.dzc, zero);  
  
    vector float rdxre = spu_re(r.dxc);  
    vector float rdyre = spu_re(r.dyc);  
    vector float rdzre = spu_re(r.dzc);  
  
    vector float tx1 = spu_sub(minx, r.x0);  
    vector float tx2 = spu_sub(maxx, r.x0);  
    vector float ty1 = spu_sub(miny, r.y0);  
    vector float ty2 = spu_sub(maxy, r.y0);  
    vector float tz1 = spu_sub(minz, r.z0);  
    vector float tz2 = spu_sub(maxz, r.z0);  
  
    tx1 = spu_mul(tx1, rdxre);  
    tx2 = spu_mul(tx2, rdxre);  
    ty1 = spu_mul(ty1, rdyre);  
    ty2 = spu_mul(ty2, rdyre);  
    tz1 = spu_mul(tz1, rdzre);  
  
    vector float tmin = spu_sel(tx2, tx1, rdxgt0);  
    vector float tmax = spu_sel(tx1, tx2, rdxgt0);  
    vector float tmin = spu_sel(ty2, ty1, rdygt0);  
    vector float tmax = spu_sel(ty1, ty2, rdygt0);  
    vector float tmin = spu_sel(tz2, tz1, rdzgt0);  
    vector float tmax = spu_sel(tz1, tz2, rdzgt0);  
}
```

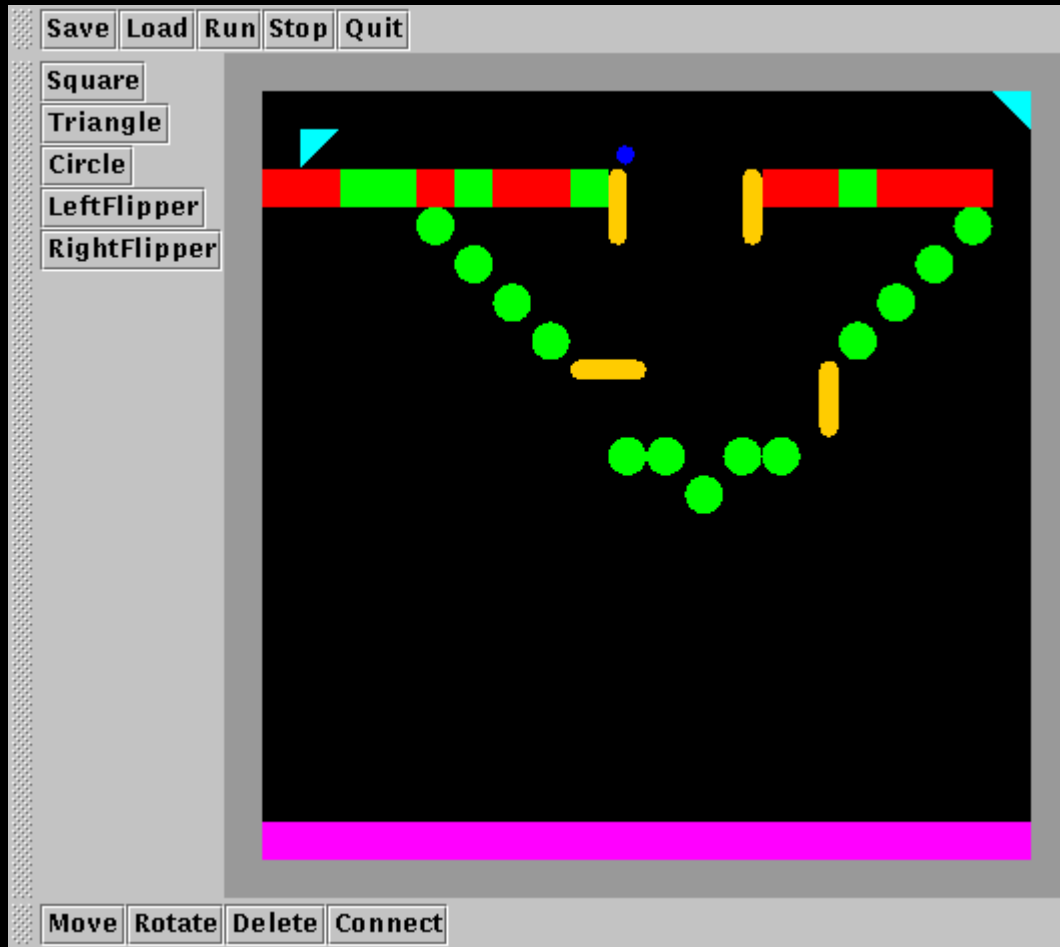
# Regular Grid

---



MIT EECS 6.837, Durand

# Gizmoball





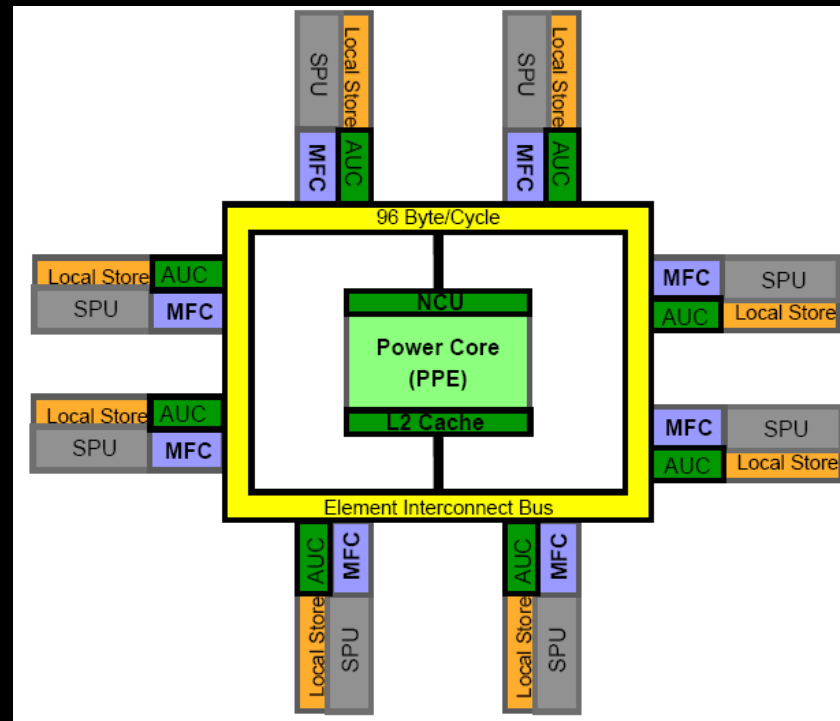
# parallelization

task is intrinsically parallelizable

each spu renders every sixth line of screen

spus dma pixel data directly to frame buffer

ppu controls physics, program initiation, scene parsing

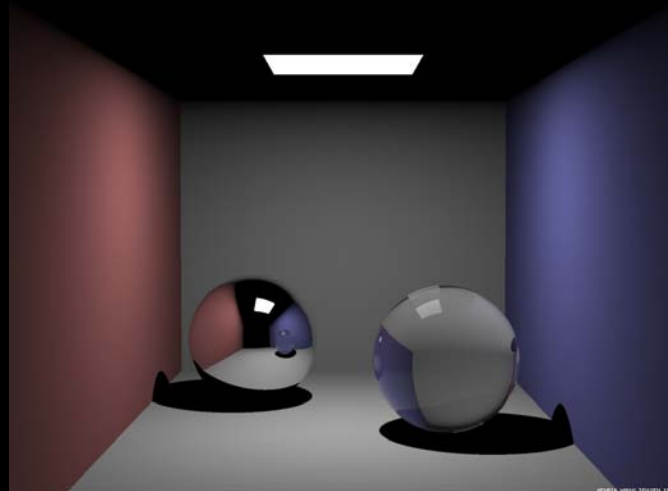


# performance

linear speedup with respect to number of cores used for rendering  
measured in length of time taken to generate a given frame

final framerate with reflection, refraction, procedural shading,  
bump mapping, shadows, and blending: 10 frames per second

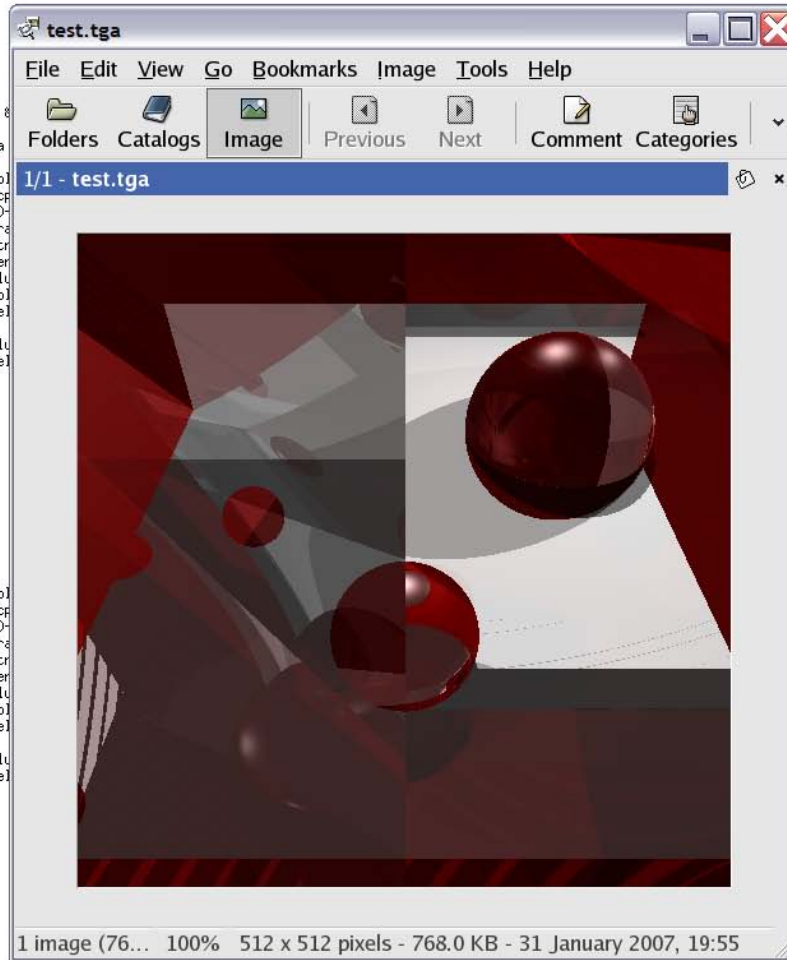
total rays traced: over 5,000,000 per second





# problems

```
Camera center: 1,000000 -0,500000 1,500000
(2)Done rendering scene
(4)Done rendering scene
(5)Done rendering scene
(1)Done rendering scene
(0)Done rendering scene
(3)Done rendering scene
[natalia3@sloth blue-steel]$ gthumb test.tga &
[8] 21112
[7] Done gthumb test.tga
[natalia3@sloth blue-steel]$ make
make[1]: Entering directory `/home/natalia3/bl
/opt/ibmcomp/xlc/8.1/bin/spuclc++ -qcc
"/src/./src/hitpacket.h", line 13,12: 1540-
/opt/ibmcomp/xlc/8.1/bin/spuclc++ -o spu_raytr
/usr/bin/embedspu -m32 spu_raytracer spu_raytr
/usr/bin/ar -qcs spu_raytracer.a spu_raytracer
make[1]: Leaving directory `/home/natalia3/bl
make[1]: Entering directory `/home/natalia3/bl
/opt/ibmcomp/xlc/8.1/bin/ppuclc++ -o blue-steel
cp -pf blue-steel ../blue-steel
make[1]: Leaving directory `/home/natalia3/bl
[natalia3@sloth blue-steel]$ sudo ./blue-steel
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
(3)Done rendering scene
(2)Done rendering scene
(4)Done rendering scene
(5)Done rendering scene
(1)Done rendering scene
(0)Done rendering scene
[natalia3@sloth blue-steel]$ make
make[1]: Entering directory `/home/natalia3/bl
/opt/ibmcomp/xlc/8.1/bin/spuclc++ -qcc
"/src/./src/hitpacket.h", line 13,12: 1540-
/opt/ibmcomp/xlc/8.1/bin/spuclc++ -o spu_raytr
/usr/bin/embedspu -m32 spu_raytracer spu_raytr
/usr/bin/ar -qcs spu_raytracer.a spu_raytracer
make[1]: Leaving directory `/home/natalia3/bl
make[1]: Entering directory `/home/natalia3/bl
/opt/ibmcomp/xlc/8.1/bin/ppuclc++ -o blue-steel
cp -pf blue-steel ../blue-steel
make[1]: Leaving directory `/home/natalia3/bl
[natalia3@sloth blue-steel]$ sudo ./blue-steel
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
Camera center: 1,000000 -0,500000 1,500000
(4)Done rendering scene
(5)Done rendering scene
(3)Done rendering scene
(2)Done rendering scene
(0)Done rendering scene
(1)Done rendering scene
[natalia3@sloth blue-steel]$ make
make[1]: Entering directory `/home/natalia3/blue-steel/spu'
/opt/ibmcomp/xlc/8.1/bin/spuclc++ -qplusplus -M -ma -I. -I. /src/ -I. -I /opt/ibm/cell-sdk/prototype//sysroot/usr/spu/include -qinclude=spu_intrinsics.h -03 -g -c ./src/box.cpp
"/src/./src/hitpacket.h", line 13,12: 1540-1608 (W) An anonymous union should only define non-static data members.
/opt/ibmcomp/xlc/8.1/bin/spuclc++ -o spu_raytracer triangle.o spu_raytracer.o sphere.o plane.o camera.o raytracer.o light.o material.o box.o -L/opt/ibm/cell-sdk/prototype//sysroot/usr/spu/lib -Wl,-N -lvector -lmisc -l
/usr/bin/embedspu -m32 spu_raytracer spu_raytracer spu_raytracer-embed.o
/usr/bin/ar -qcs spu_raytracer.a spu_raytracer-embed.o
make[1]: Leaving directory `/home/natalia3/blue-steel/spu'
make[1]: Entering directory `/home/natalia3/blue-steel/ppu'
/opt/ibmcomp/xlc/8.1/bin/ppuclc++ -o blue-steel main.o image.o fb.o -L../lib/ -L/opt/ibm/cell-sdk/prototype//sysroot/usr/lib -Wl,-m,elf32ppc -R/opt/ibm/cell-sdk/prototype//sysroot/usr/lib ../spu/spu_raytracer.a -lspe -lm
cp -pf blue-steel ../blue-steel
make[1]: Leaving directory `/home/natalia3/blue-steel/ppu'
[natalia3@sloth blue-steel]$ sudo ./blue-steel
```



```
_intrinsics.h -03 -g -c ./src/box.cpp
opt/ibm/cell-sdk/prototype//sysroot/usr/spu/lib -Wl,-N -lvector -lmisc -l
```

```
opt/ibm/cell-sdk/prototype/sysroot/usr/lib ../spu/spu_raytracer.a -lspe -lm
```

```
_intrinsics.h -03 -g -c ./src/box.cpp
opt/ibm/cell-sdk/prototype//sysroot/usr/spu/lib -Wl,-N -lvector -lmisc -l
```

```
opt/ibm/cell-sdk/prototype/sysroot/usr/lib ../spu/spu_raytracer.a -lspe -lm
```

```
-L/opt/ibm/cell-sdk/prototype//sysroot/usr/spu/lib -Wl,-N -lvector -lmisc -l
```

```
-L/opt/ibm/cell-sdk/prototype//sysroot/usr/lib -Wl,-m,elf32ppc -R/opt/ibm/cell-sdk/prototype//sysroot/usr/lib ../spu/spu_raytracer.a -lspe -lm
```

team

scott fisher

michael d'ambrosio

brian sweatt

natalia chernenko

rj ryan

leevar williams