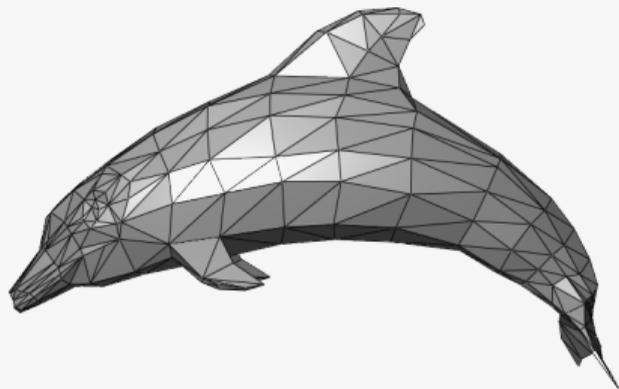




Трассировка лучей и генерация
псевдослучайных чисел

Раздел 1

Трассировка лучей



Растреризация

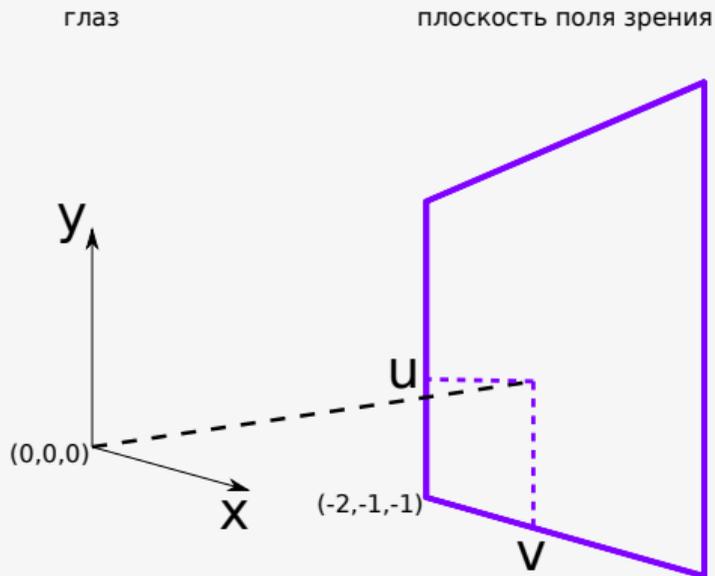
- ▶ Для каждого треугольника ищутся пиксели, которые он закрывает.
- ▶ Реализуется в OpenGL с помощью шейдеров.

Трассировка лучей

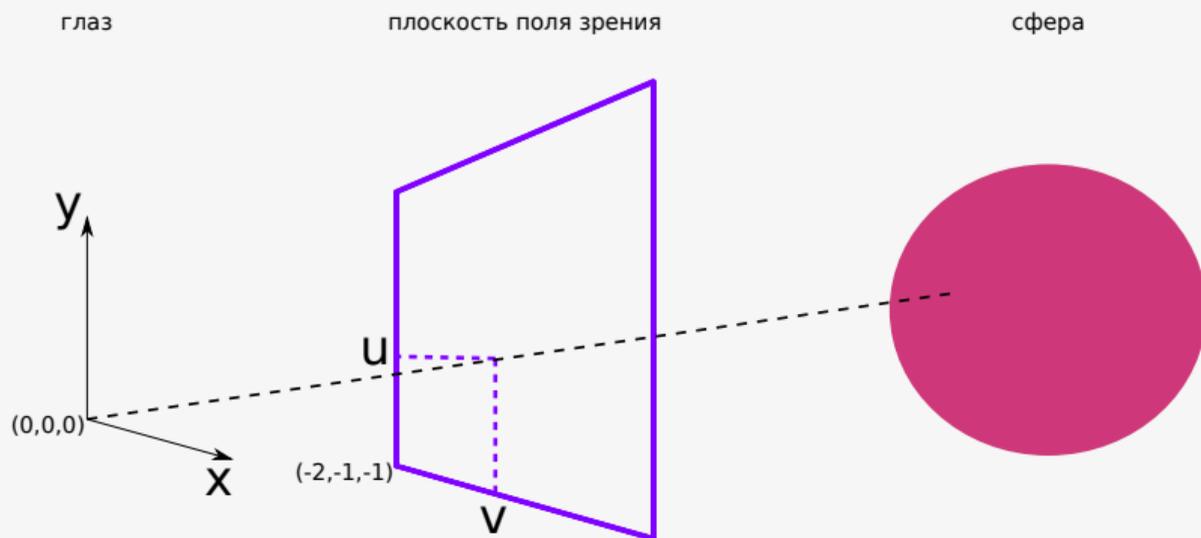
- ▶ Для каждого пикселя ищется ближайший треугольник.
- ▶ Реализуется в OpenCL с помощью ядер.

Задний план

```
vec3 bottom_left(-2,-1,-1);
vec3 horizontal(4,0,0);
vec3 vertical(0,2,0);
vec3 origin(0,0,0);
for (int i=0; i<nx; ++i) {
    for (int j=0; j<ny; ++j) {
        float u = float(i)/nx;
        float v = float(j)/ny;
        ray r(origin, bottom_left +
            u*horizontal + v*vertical);
        pixels[i][j] = trace(r);
    }
}
```



Сфера



```
vec3 trace(const ray& r) {  
    if (hit_sphere(...)) { return ...; } else { return ...; }  
}
```

Сглаживание

```
for (int i=0; i<nx; ++i) {  
    for (int j=0; j<ny; ++j) {  
        vec3 col(0,0,0);  
        for (int k=0; k<nrays; ++k) {  
            float u = (i+random(0,1))/nx;  
            float v = (j+random(0,1))/ny;  
            ray r(origin, bottom_left +  
                u*horizontal + v*vertical);  
            col += trace(r);  
        }  
        pixels[i][j] = col/nrays;  
    }  
}
```

И многое другое:

- ▶ отражение (зеркала),
- ▶ рефракция (полупрозрачные объекты),
- ▶ поглощение (матовые поверхности)
- ▶ размытие изображения,
- ▶ и т.д.



Трассировка лучей на видеокарте

Параллелизм по пикселям:

```
int i = get_global_id(0);
int j = get_global_id(1);
float u = ((float)i)/nx;
float v = ((float)j)/ny;
ray r(origin, bottom_left + u*horizontal + v*vertical);
pixels[i*nx + j] = trace(r);
```

Параллелизм по объектам?

```
vec3 color(const ray& r) {
    ...
    for (int i=0; i<nobjects; ++i) {
        if (hit(r, objects[i])) { ... } else { ... }
    }
    ...
}
```

Раздел 2

Генерация псевдослучайных чисел

Генерация псевдослучайных чисел

Два этапа

- ▶ генерация последовательности чисел с равномерными распределением,
- ▶ преобразование к нужному распределению (функция квантилей).

Преобразование к нормальному распределению

Функция распределения:

$$F(x) = \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left[\frac{x}{\sqrt{2}} \right]$$

Обратная функция распределения:

$$F^{-1}(p) = \sqrt{2} + \operatorname{erf}^{-1} [2p - 1]$$

Преобразование Бокса—Мюллера:

$$x_0 = \sin(2\pi p_0) \sqrt{-2 \log p_1}$$

$$x_1 = \cos(2\pi p_0) \sqrt{-2 \log p_1}$$

Вихрь Мерсенна

Последовательная версия

```
unsigned long seed = ...;  
sgenrand(seed);          // инициализация генератора  
double d = genrand();    // генерация псевдо-случайного числа
```

Параллельная версия

Программа для динамического создания генераторов (DCMT):

```
init_dc();                // инициализация генератора  
mtconf = genmt(period);  // генерация Вихря Мерсенна
```

Основная программа:

```
unsigned long seed = ...;  
sgenrand(seed, mtconf);  // из программы 1  
double d = genrand();    // генерация псевдо-случайного числа
```

Особенности

- ▶ Программа DCMT последовательная, основная — параллельная.
- ▶ У параллельных генераторов меньший период.
- ▶ Последовательности чисел некоррелированы друг с другом.
- ▶ Параллелизм по данным.

© 2019–2021 Ivan Gankevich i.gankevich@spbu.ru

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. The copy of the license is available at <https://creativecommons.org/licenses/by-sa/4.0/>.

Статьи

- ▶ Peter Shirley [Ray Tracing in One Weekend](#).
- ▶ Roger Allen [Accelerated Ray Tracing in One Weekend in CUDA](#).