

# Трассировка лучей в реальном времени на x64 архитектуре

**Исполнитель:**

магистрант группы 86М1  
Морозов Андрей Сергеевич

**Научный руководитель:**

д. т. н., профессор кафедры МО ЭВМ  
Турлапов Вадим Евгеньевич

# Постановка задачи

Разработать, реализовать и исследовать высокопроизводительный алгоритм трассировки лучей на центральном процессоре:

- алгоритмическая оптимизация (ускоряющие структуры)
- параллелизм на уровне данных
- параллелизм на уровне команд

- Алгоритм обратной трассировки лучей
  - Модель Фонга

$$\vec{I}_{local} = K_{amb} \cdot \vec{I}_{amb} + K_{diff} \cdot \vec{I}_{diff} \cdot (\vec{L}, \vec{N}) + K_{spec} \cdot \vec{I}_{spec} \cdot (\vec{R}, \vec{V})^n$$

$$\vec{I}_{total} = \vec{I}_{local} + K_{reflection} \cdot \vec{I}_{reflection} + K_{refraction} \cdot \vec{I}_{refraction}$$

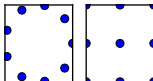
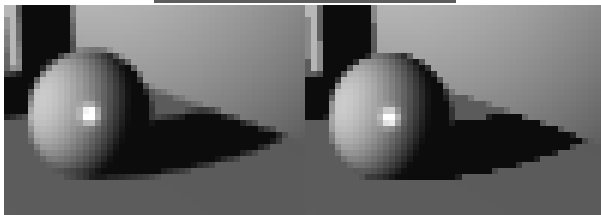
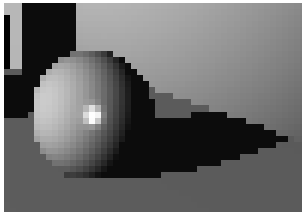
- Теневые лучи
- Расчет отражений

$$\vec{R} = \vec{I} - 2 \cdot \vec{N}(\vec{N}, \vec{I})$$

- Алгоритмическая оптимизация
  - BVH
- Параллелизм на уровне данных
  - OpenMP, TBB
- Параллелизм на уровне команд
  - SSE        957
  - SS2        403
  - SSE3       26
  - SSE4.1    82

# Результаты

- Supersampling



- CPU

- Intel Core i7 980X @ 3.33GHz
- MMX, SSE(1, 2, 3, 3S, 4.1, 4.2), EM64T, VT-x, AES
- Cores/Threads : 6 / 12
- L1/L2/L3 : 6 x 32 KBytes / 6 x 256 KBytes / 12 MBytes

- RAM

- 1066 MHz
- 6 x 2 GB

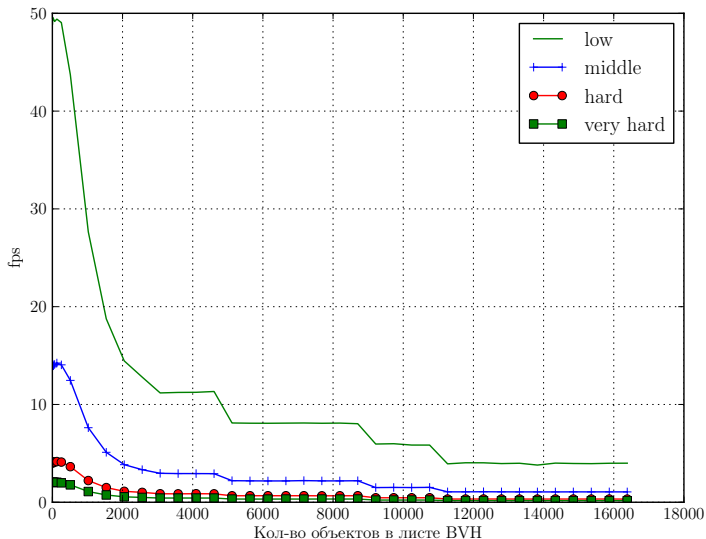
- OC

- Calculate Linux 11.3 x64
- Kernel : 2.6.38.6

- сфера
- меш – набор треугольников
- итого : 10817 объектов

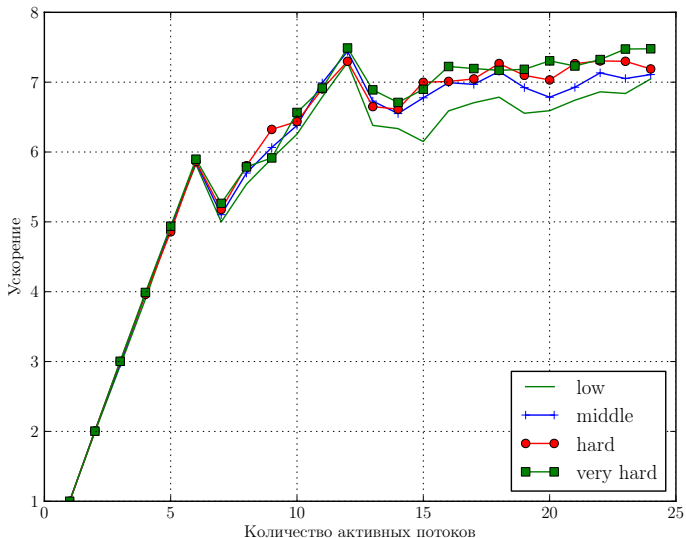
Название режима	Ширина изображения	Высота изображения	Кол-во суб-пикселей	Глубина трассировки	Кол-во источников света
low	512	512	1	1	1
middle	800	600	2	2	1
hard	1024	768	3	4	2
very hard	1920	1080	3	8	2

# Результаты





# Результаты



# Итоговое изображение



# Заключение

- Удалось реализовать высокопроизводительный алгоритм трассировки лучей
  - Разрешение : 640x480
  - Объектов в сцене : 10817
  - Глубина трассировки : 16
  - Итоговая производительность : 36 fps
- Основной вклад в ускорение алгоритма:
  - BVH ( 12 – 13 раз )
  - OpenMP ( 5 – 7.6 раз ) ( Hyper-Threading  $\approx +30\%$  )
  - SSE + ET ( 7 – 8 раз )
  - Итоговое ускорение :  $\approx 640$  раз
- Характеристика проекта:
  - 55 + 10 файлов, 5700 + 650 строк кода.
  - ОС: Calculate Linux 11.3 x64, QtCreator, Scons, GCC, Python,  $\text{\LaTeX}$

Спасибо за внимание !