

Разработка библиотеки для портирования
приложений с трехмерной графикой, использующих
конвейер фиксированной функциональности,
на метод трассировки путей

Цырендашиев Сулытим Баиржапович, 17.Б11-мм

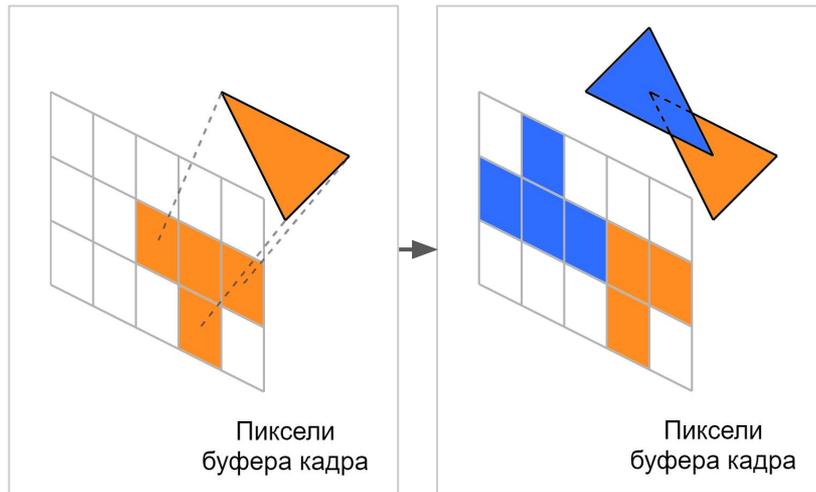
Научный руководитель: к.ф.-м.н., доцент Д.В. Луцив

Консультант: ген.дир. ООО "СКЗ" А.А. Пименов

Рецензент: программист ООО "Леста" Е. М. Щавелев

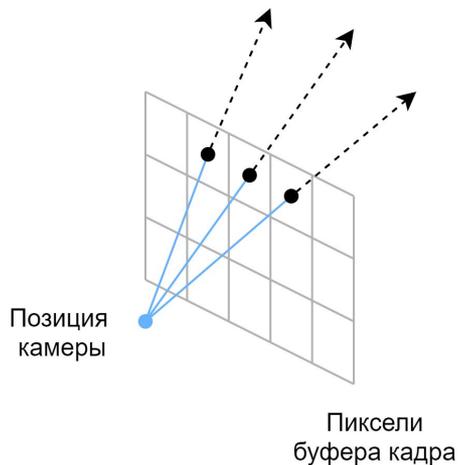
Введение

- Растеризация



```
for object:  
  for pixel:  
    ...
```

- Трассировка лучей



```
for pixel:  
  for object:  
    ...
```

Введение

- Растеризация — высокая производительность. Виды конвейеров:
 - шейдерный конвейер: *с начала 2000-х*
 - гибкость: возможность написания собственных программ (шейдеров) для обработки вершин, пикселей и т.д.
 - фиксированной функциональности: *90-е — начало 2000-х*
- Трассировка лучей
 - Аппаратная поддержка стала доступна в потребительских видеокартах с 2018
 - Подтип: трассировка путей — расчёт уравнения отрисовки в пересечениях, реалистичное освещение

Введение

- Появилась возможность трассировки путей с высокой частотой кадров
- Но всё ещё мало приложений, использующих метод в полной мере:
 - производительность
 - новые концепции — глобальное изменение алгоритмов отрисовки
- Первоначальная идея — портирование видеоигры “SS: TFE”



Кадр из видеоигры “Serious Sam: The First Encounter” (2001)

Цели и задачи

Цель: разработать библиотеку, позволяющую использовать **трассировку путей** в приложениях с конвейером **фиксированной функциональности** для отрисовки трехмерных сцен.

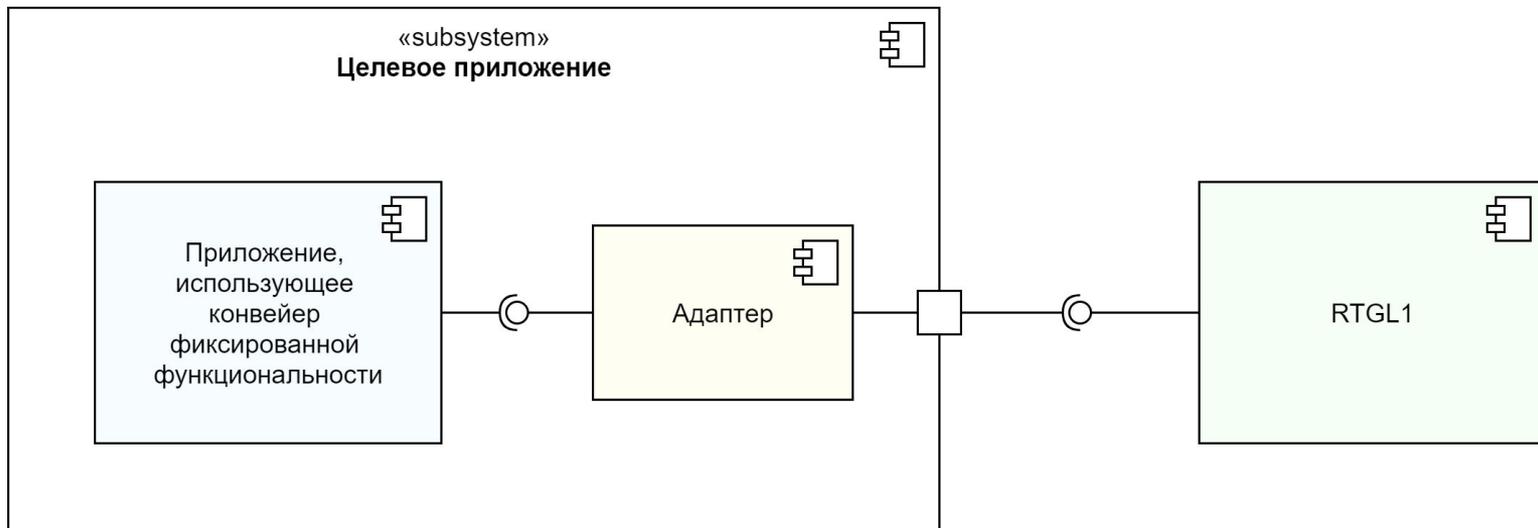
Задачи:

- Проектирование и реализация интерфейса библиотеки
- Реализация системы обработки данных для отрисовки
- Реализация расчета освещения с помощью трассировки путей
- Интеграция библиотеки в целевое приложение

Существующие решения

- Открытый исходный код
 - Исключительно *трассировка* лучей для расчета освещения с *высокой* частотой смены кадров
1. Q2VKPT (2019), Quake 2 RTX (2019)
 - отрисовка разработана специально для Quake 2 (1997)
 2. RTX Direct Illumination (2021), RTX Global Illumination (2021)
 - не нацелены на приложения с конвейером фиксированной функциональности

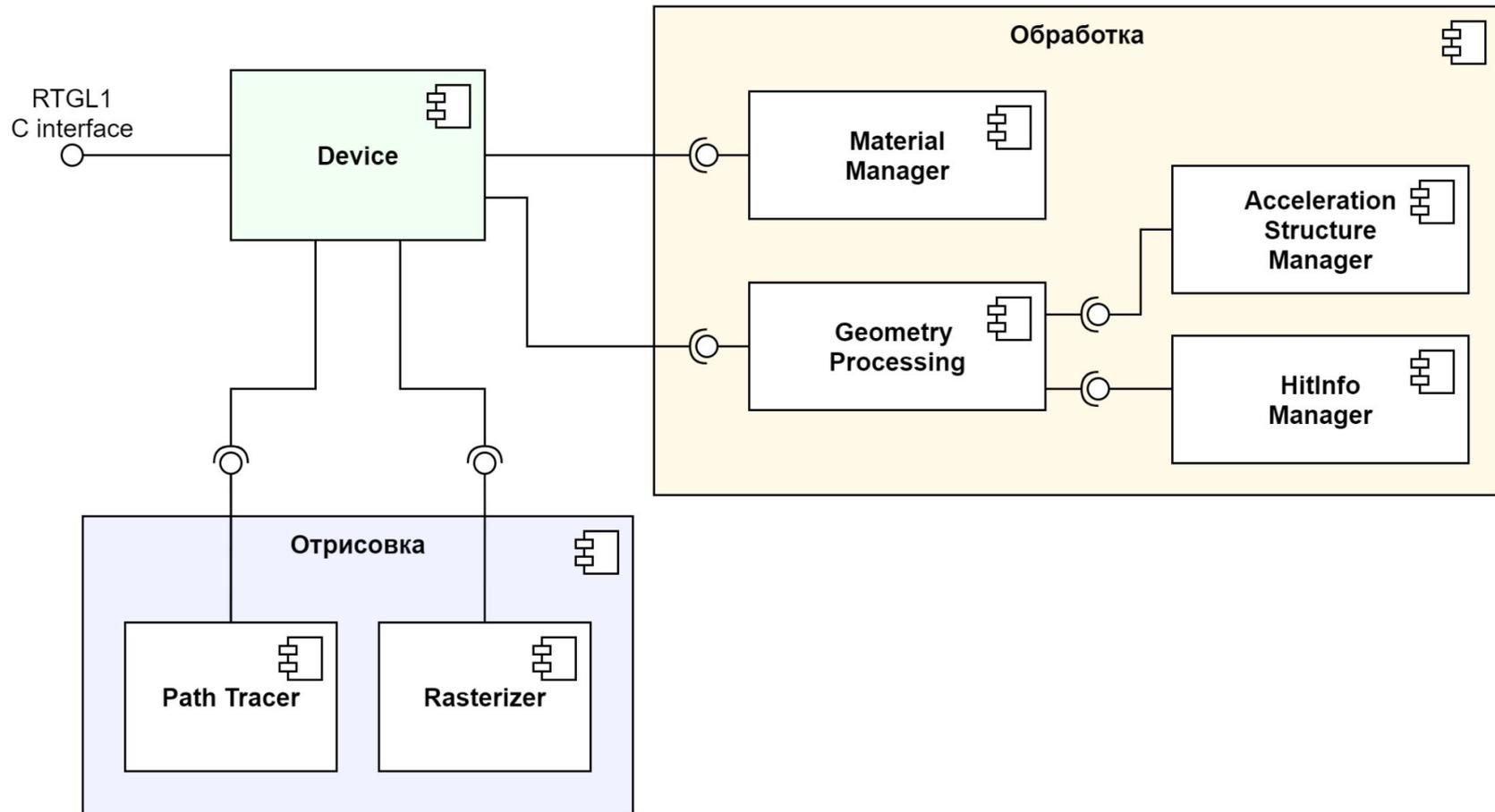
Предлагаемое решение

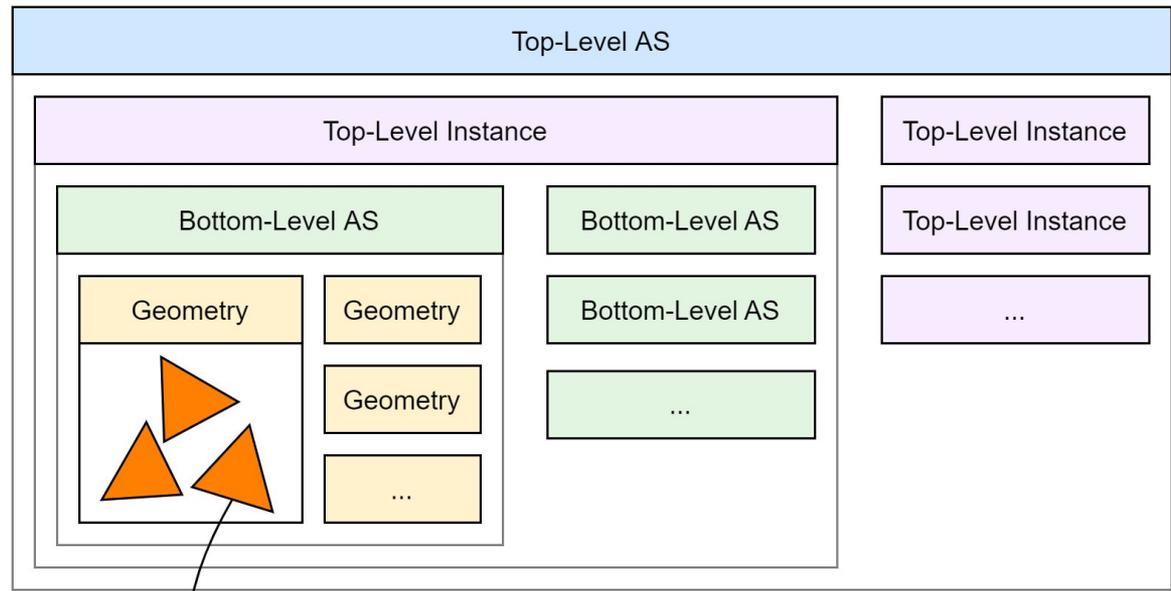


<https://github.com/sultim-t/RayTracedGL1>

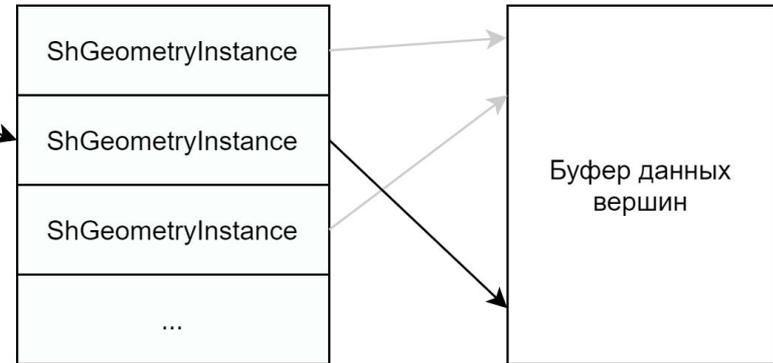
Используемые технологии

- C++
- Vulkan API — взаимодействие с видеокартой на низком уровне
- Язык GLSL — написание шейдеров
- Nvidia Nsight Graphics — профилирование и отладка кадров, сгенерированных на видеокарте



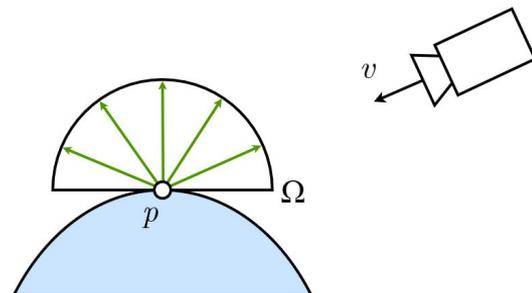


Пересечение луча
и треугольника



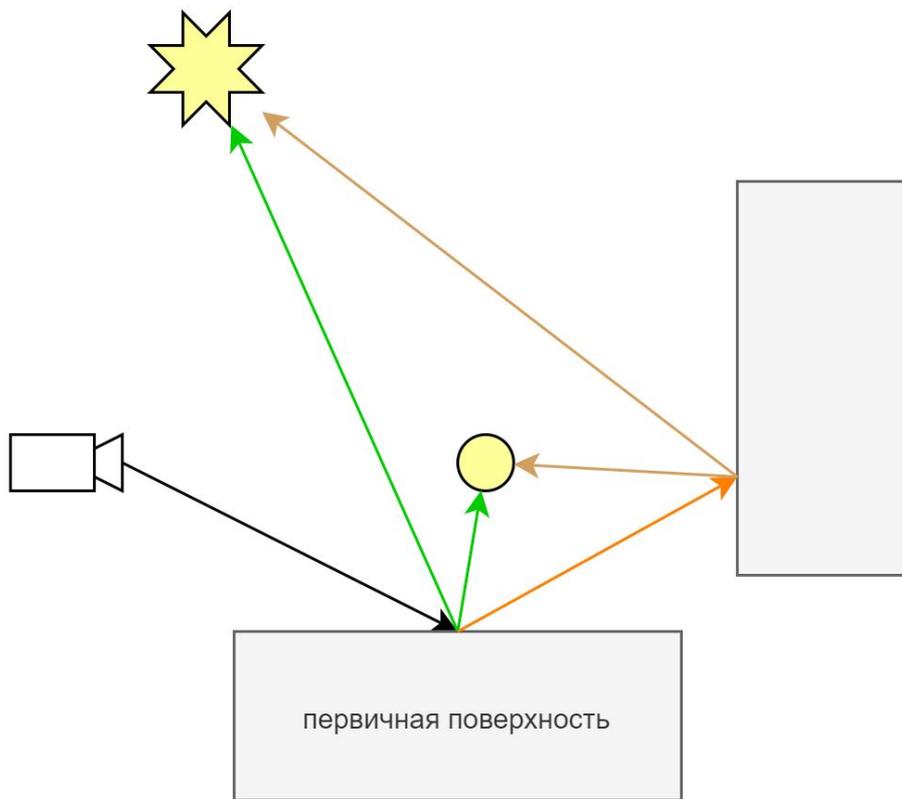
Расчет освещения

$$L_o(p, v) = L_e(p, v) + \int_{l \in \Omega} f(l, v) L_i(p, l) |n \cdot l| dl$$



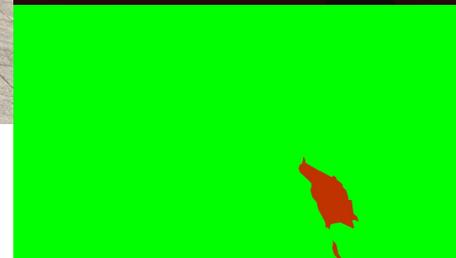
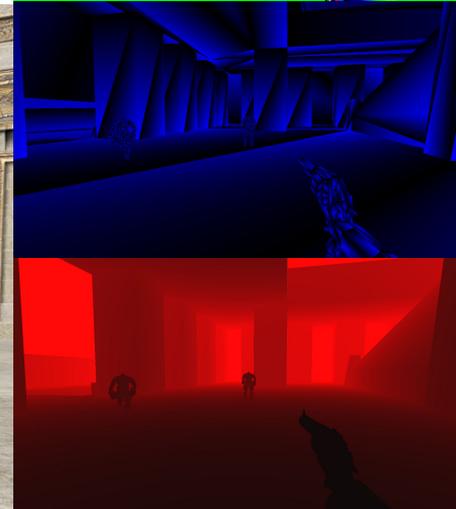
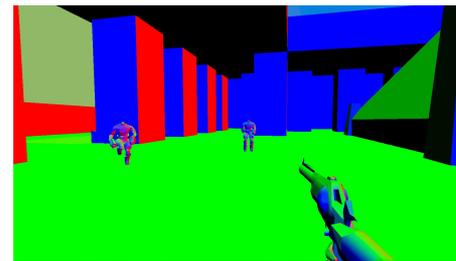
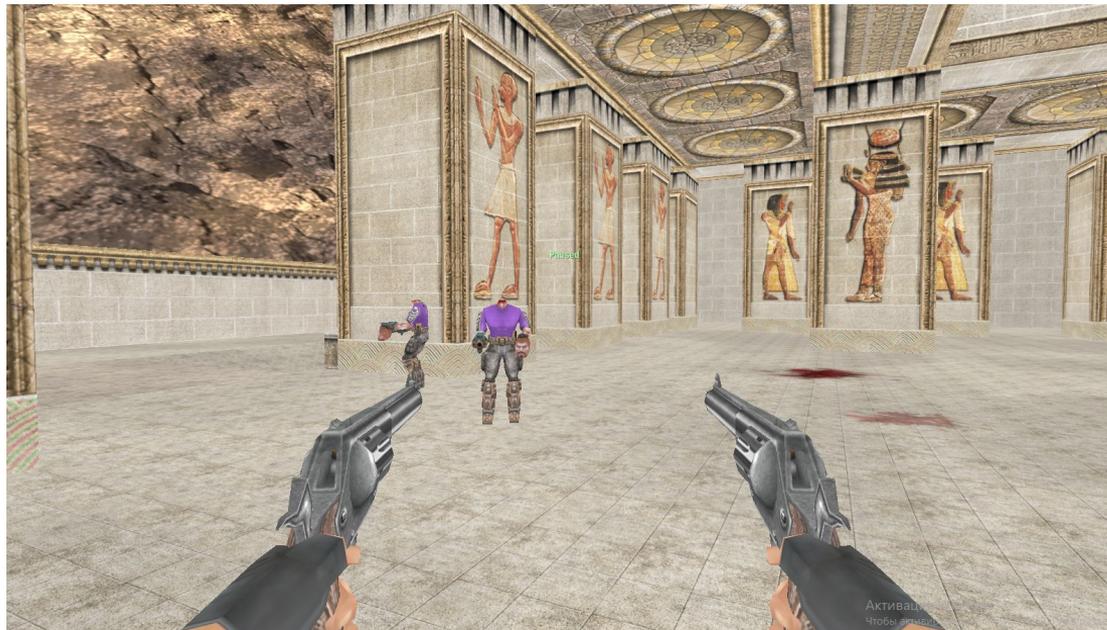
- Малое кол-во лучей \Rightarrow разделение на типы:
 - рассеянное прямое
 - зеркальное
 - рассеянное не прямое
- Т.к. поведение таких типов освещения отличается в одних и тех же условиях сцены
 - Например, поворот камеры

Лучи в сцене



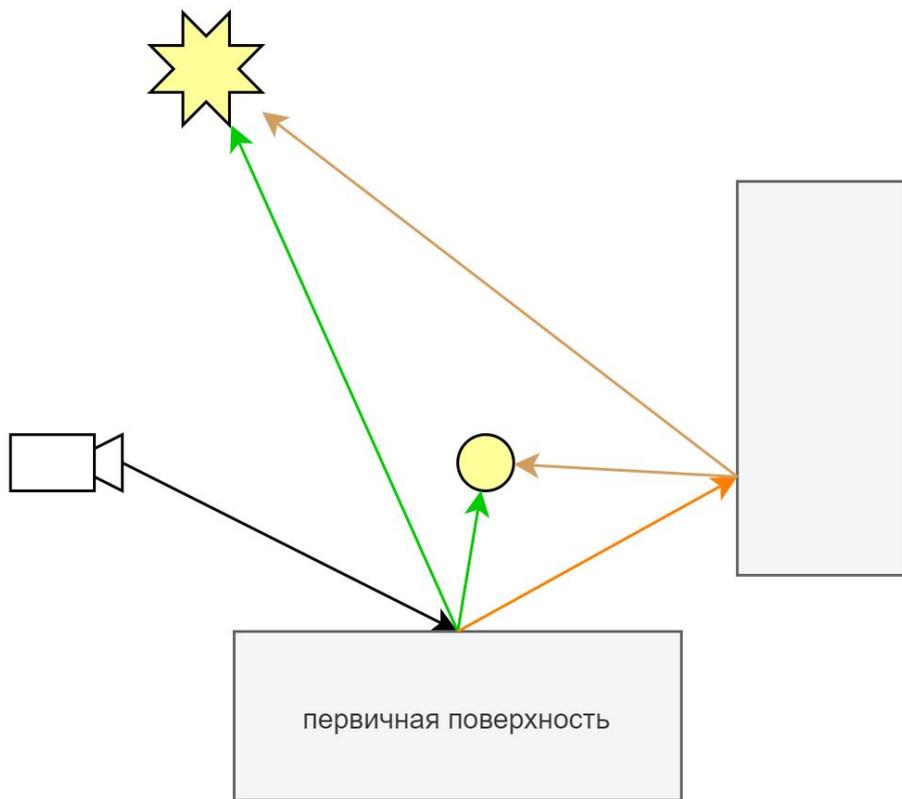
- **1 луч** — подготовка информации для расчета
- **1 луч** — тень *одного* направленного источника
- **1 луч** — тень *одного* сферического источника
- **3 луча** — зеркальное отражение
- **3 луча** — рассеянное отражение

Первичные лучи



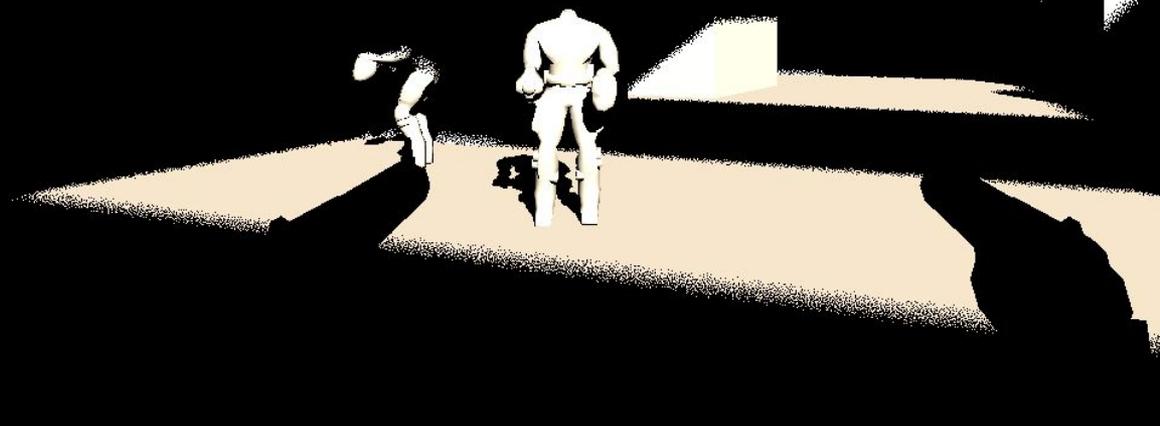
и др.

Лучи в сцене

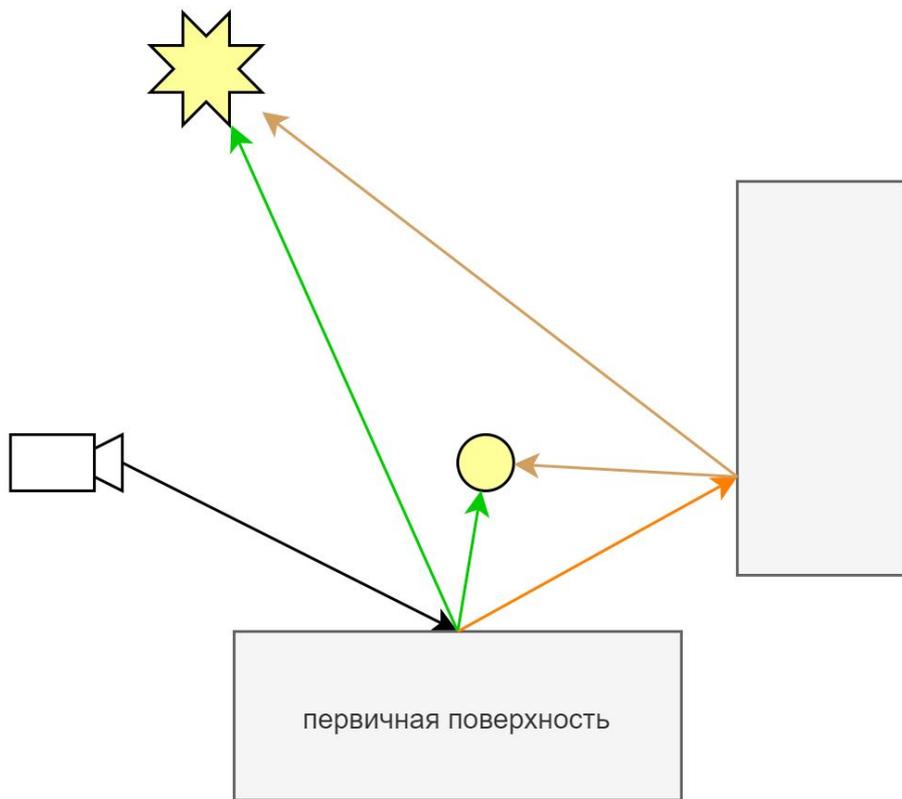


- **1 луч** — подготовка информации для расчета
- **1 луч** — тень *одного* направленного источника
- **1 луч** — тень *одного* сферического источника
- **3 луча** — зеркальное отражение
- **3 луча** — рассеянное отражение

Прямое рассеянное



Лучи в сцене



- **1 луч** — подготовка информации для расчета
- **1 луч** — тень *одного* направленного источника
- **1 луч** — тень *одного* сферического источника
- **3 луча** — зеркальное отражение
- **3 луча** — рассеянное отражение

Зеркальное



Непрямое рассеянное



Освещение
после трассировки лучей

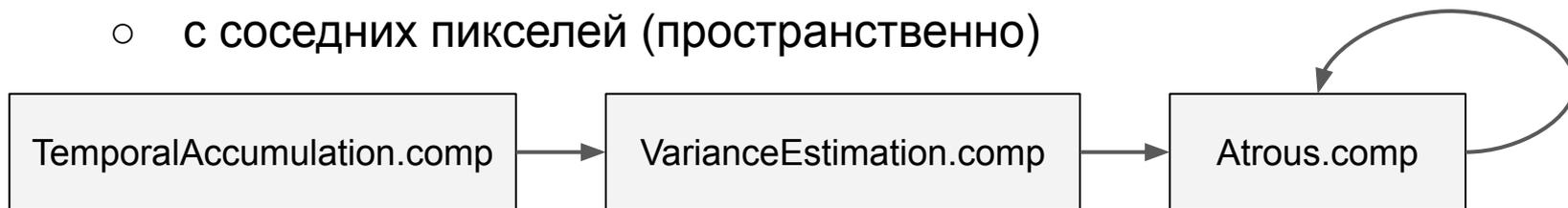


Освещение
после устранения шума

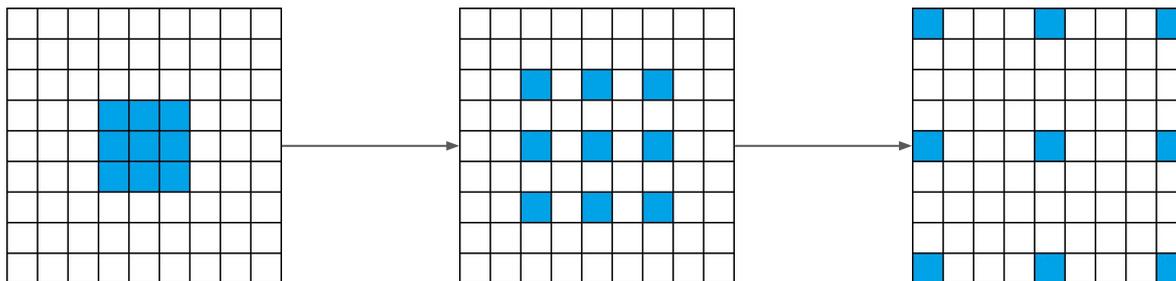


Устранение шума — SVGF

- SVGF — увеличение количества экземпляров через накопление данных:
 - с предыдущих кадров (темпорально)
 - с соседних пикселей (пространственно)



- Применение специального фильтра размытия (edge-stopping a-trous)



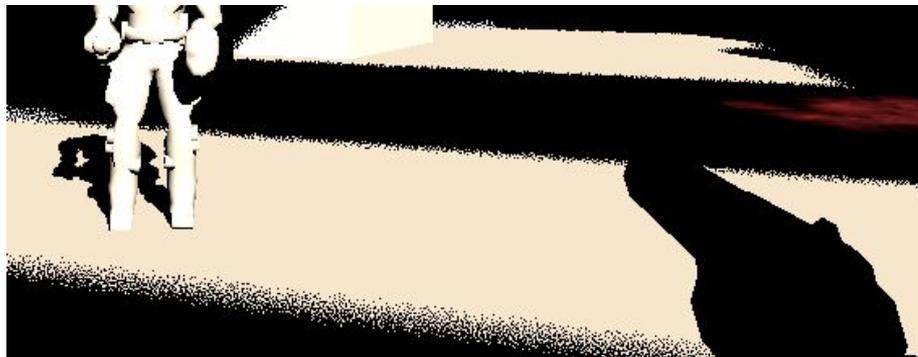
Устранение шума — Gradient Samples

- + Устранение эффекта “запаздывания”:



- Необходимо сохранять состояние сцены с предыдущего кадра

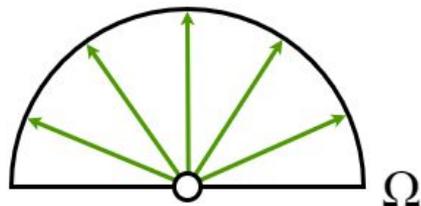
Прямое рассеянное



Зеркальное

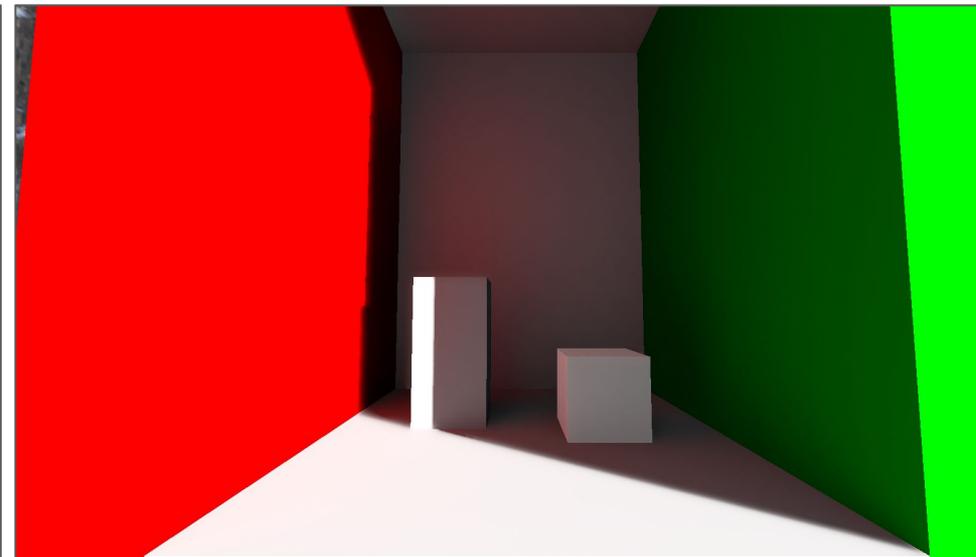
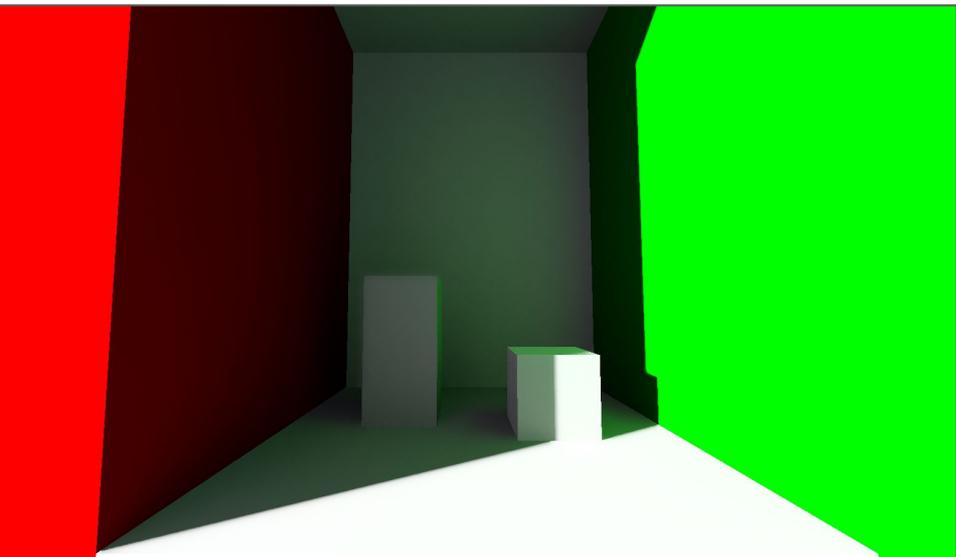
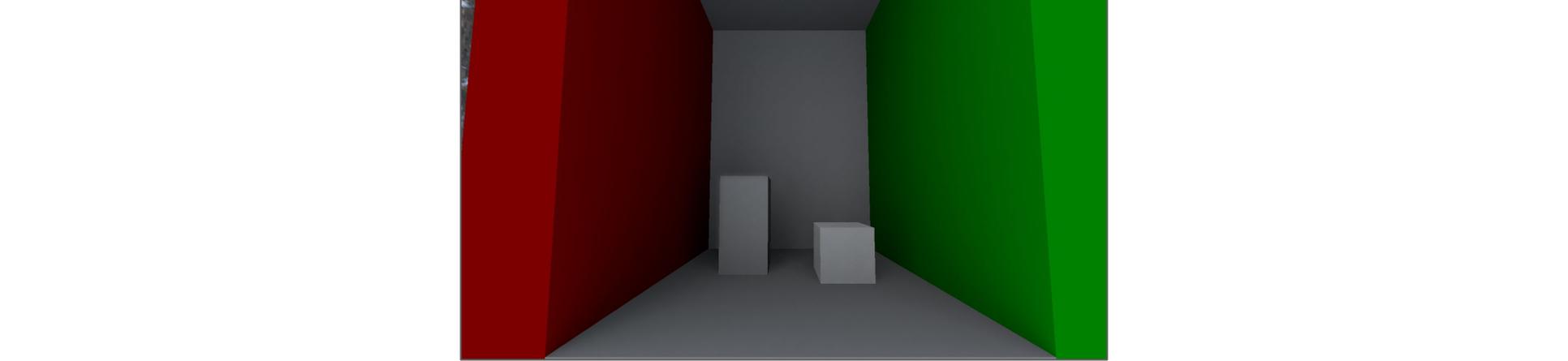


Непрямое рассеянное



- Разреженное — м.б. любое направление на полусфере около точки
- ⇒ Кодировем энерг. яркость и направление в виде карты освещенности (irradiance map)





Отрисовка растеризованной геометрии

- “дешевая” полупрозрачность
- необходим буфер глубины
 - “копируется” глубина первичной поверхности через растеризацию



Отрисовка фоновой геометрии



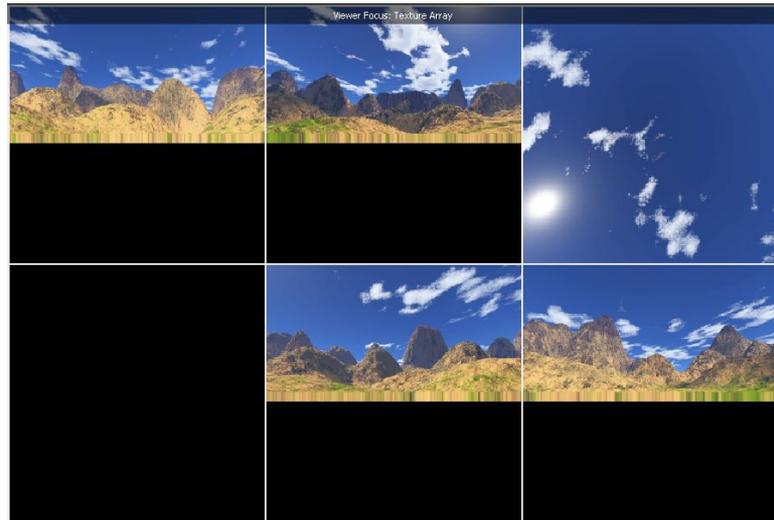
- Пересечение не было найдено. Типы:
 1. цвет
 2. кубическая текстура
 3. дополнительный acceleration structure
 4. **растеризованный + динамическая кубическая текстура**

Отрисовка фоновой геометрии — способ 4

Растреризация в альbedo канал



Альbedo
первичных
пересечений
поверх



Адаптер

- Переработана бóльшая часть системы отрисовки в целевом приложении
 - Оригинальный алгоритм отрисовки не используется
- Адаптер следит за объектами на сцене и посылает данные о них библиотеке
- Сложности:
 - недостаток информации о сцене
 - недостаток документации
 - Vulkan API требует x64 для трассировки лучей

<https://github.com/sultim-t/Serious-Engine-RT>



Paused



Paused



Original

Path traced

Итоги

- Спроектирован и реализован интерфейс библиотеки, нацеленный на приложения с конвейером фиксированной функциональности
- Реализована система для обработки геометрии, материалов и источников света
- Реализована система отрисовки с расчётом прямого и непрямого освещения с минимальным количеством лучей; реализован алгоритм устранения шума для освещения
- Реализован адаптер для интеграции созданной библиотеки в видеоигру “Serious Sam: The First Encounter” для использования метода трассировки путей