

# Глобальное освещение. Трассировка лучей. Излучательность.

---

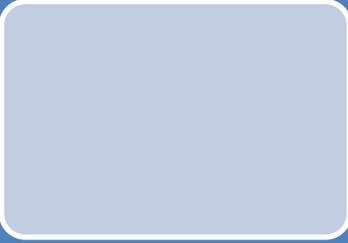
Алексей Викторович Игнатенко  
Лаборатория компьютерной графики и  
мультимедиа  
ВМК МГУ



Лекция посвящена обзору двух подходов к синтезу изображений: излучательности и трассировке лучей



Понятие о глобальном освещении и других сложных оптических эффектах



Трассировка лучей



Излучательность

# Графический процесс: типовая последовательность применения алгоритмов



# При выборе алгоритма надо выбрать между скоростью и качеством

## Скорость



Растеризация

## Качество



Трассировка  
лучей/фотонов  
Излучательность

# Некоторые оптические эффекты крайне сложно визуализировать средствами метода растеризации

---

- Отражения
- Преломления
- Тени
- Вторичное освещение
- Сложные материалы

Лекция посвящена обзору двух подходов к синтезу изображений: излучательности и трассировке лучей



Понятие о глобальном освещении и других сложных оптических эффектах



Трассировка лучей

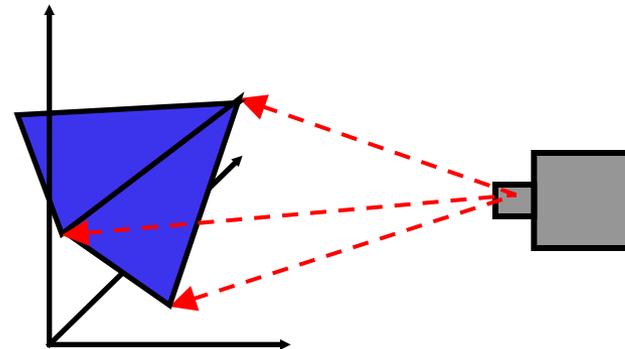
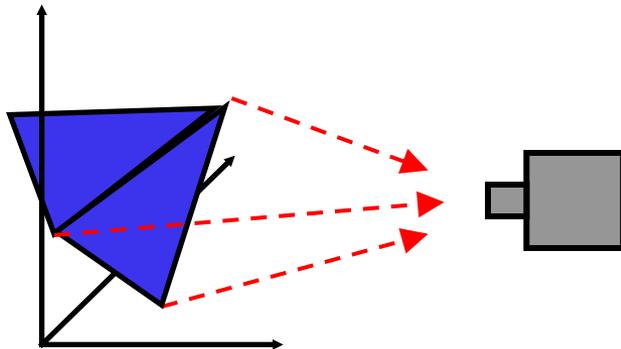


Излучательность

# Растреризация и трассировка лучей – два основных подхода к синтезу изображений

---

Два основных подхода



**Растреризация:**

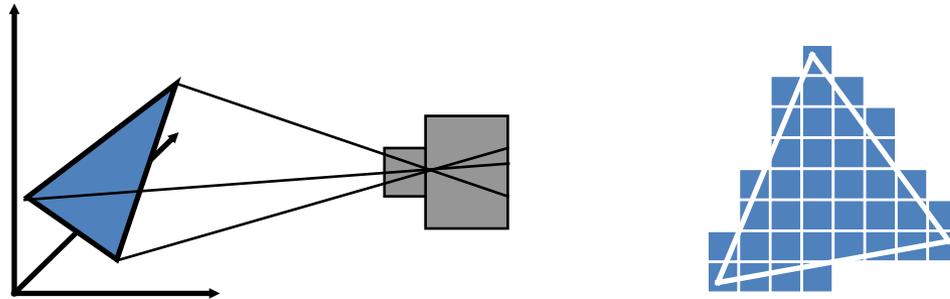
Прямая проекция геометрии

**Трассировка лучей:**

Обратная проекция пикселей изображения

# Растеризация: быстро, но каждый треугольник обрабатывается отдельно

---



Последовательная обработка всех треугольников по одному

- Сложно (неэффективно) работать более, чем с одним треугольником за раз
- Но большинство реалистичных эффектов требуют доступа ко всей сцене: тени, отражения, глобальное освещение!

# Трассировка лучей: моделируем распространение света вдоль лучей

---

**Прямая трассировка:** свет идет из источников света

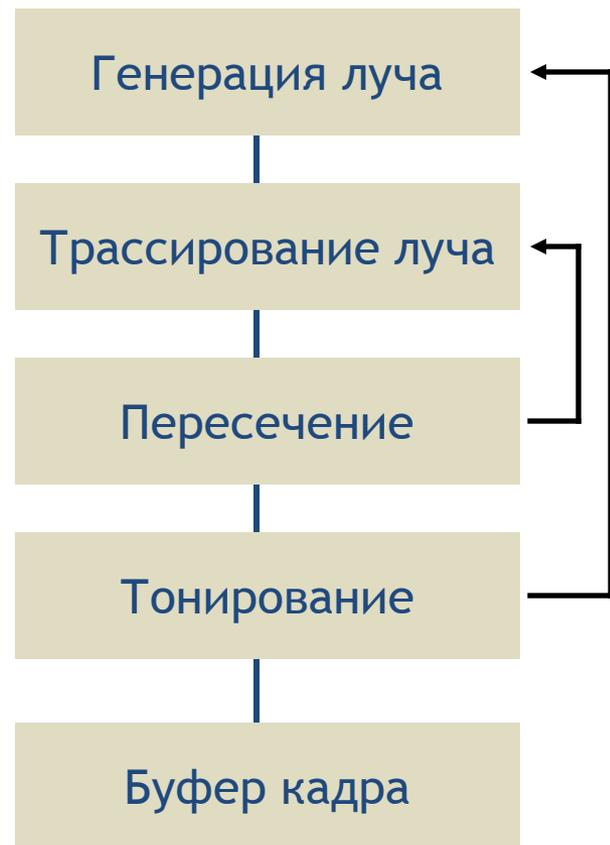
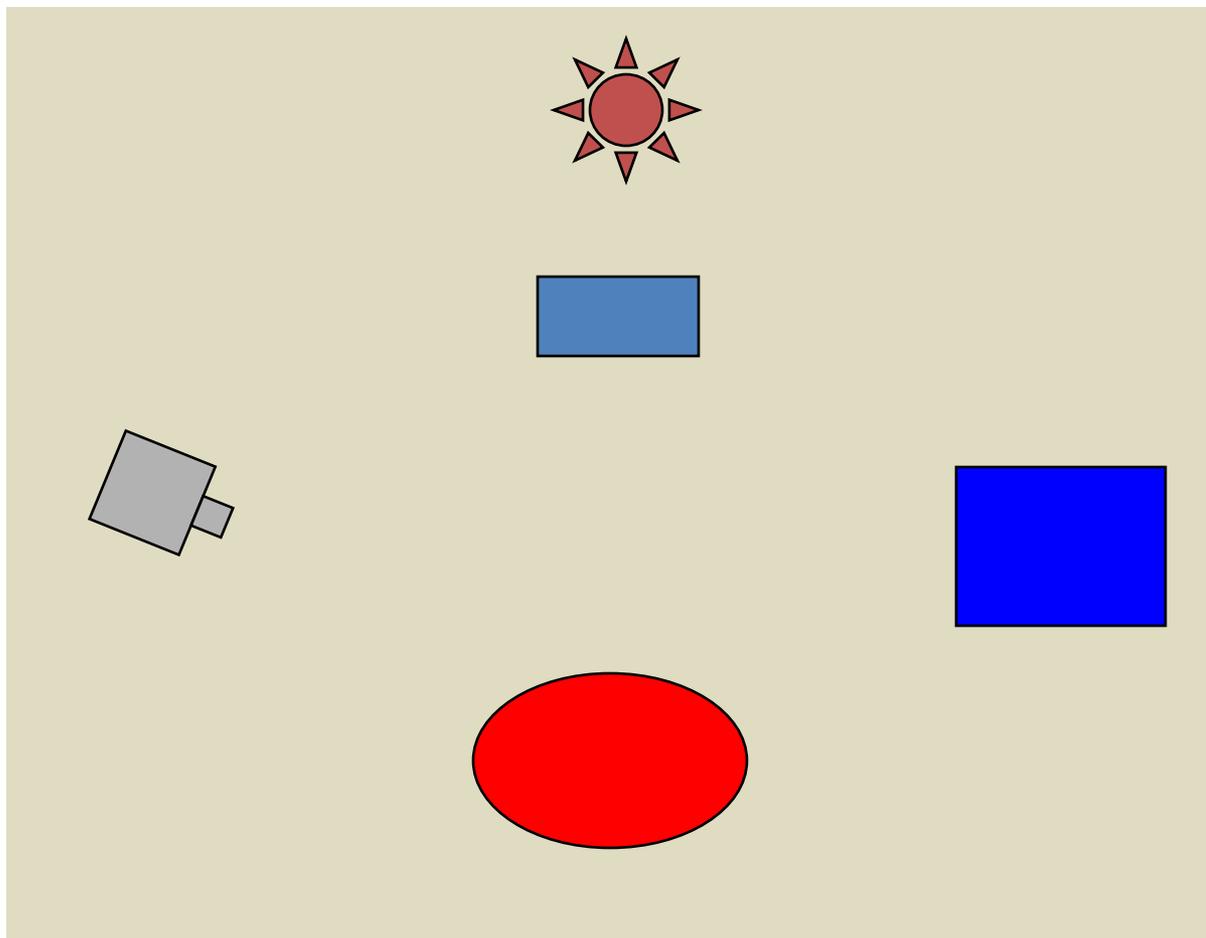
- Плюсы: точная физическая модель
- Минусы: нам нужен только свет, попадающий в камеру  
=> очень неэффективно

Из физики известно, что распространение света обратимо  
=> можно проследить (протрассировать) свет из каждого пикселя.

**Обратная трассировка:** «свет» идет из каждого пикселя изображения

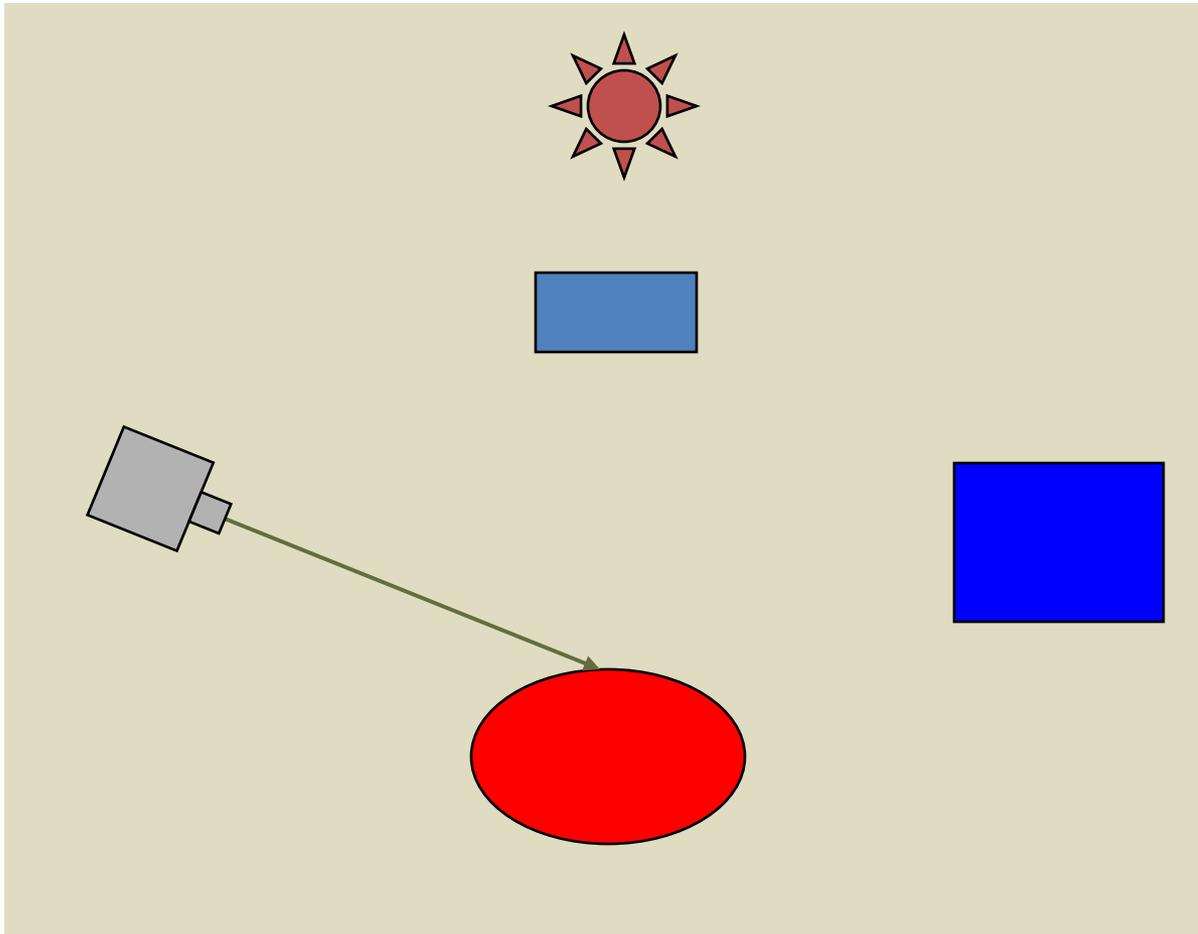
# Трассировка лучей

---



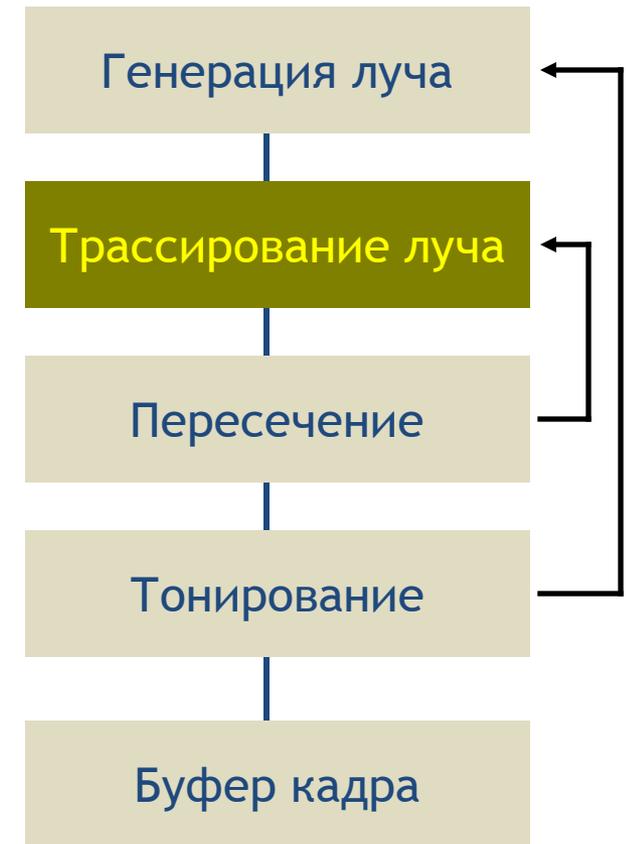
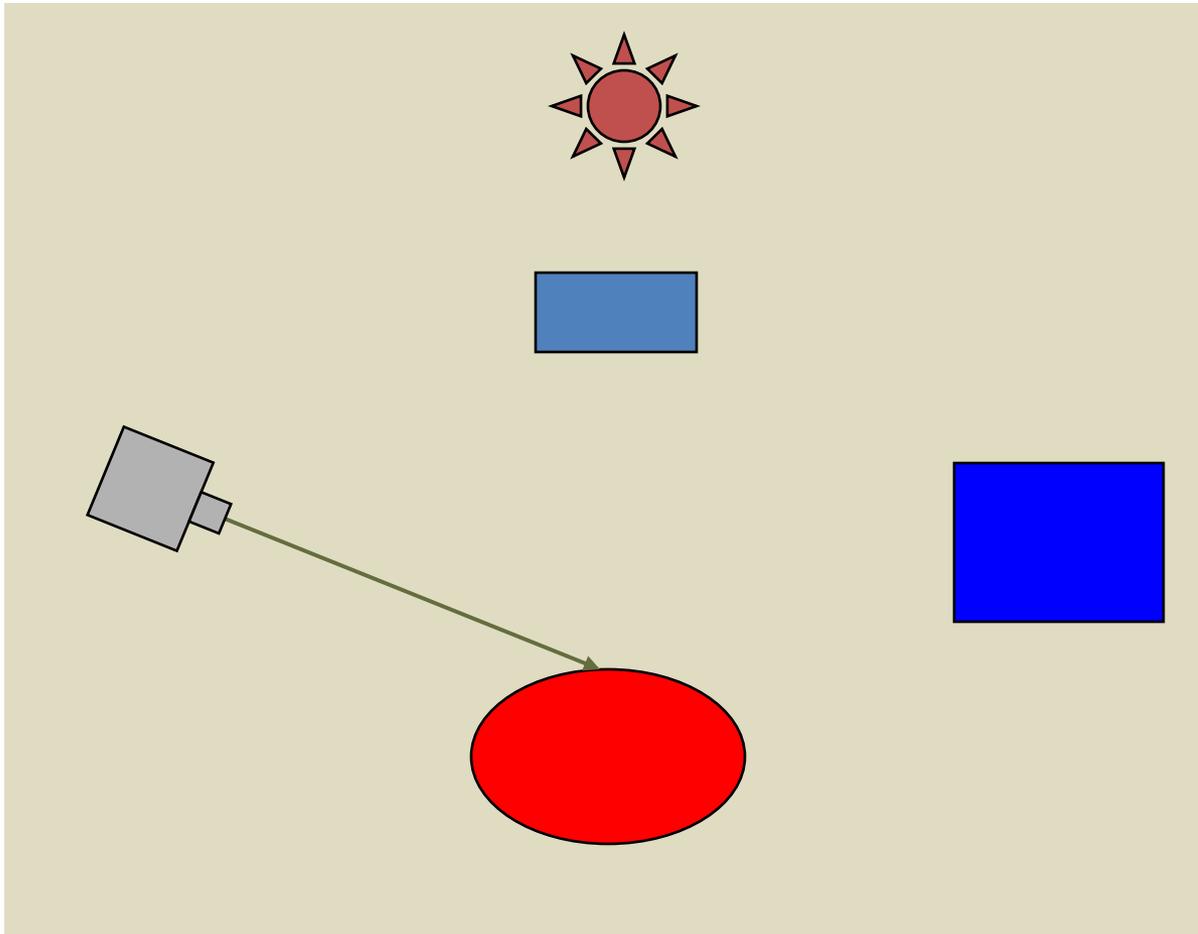
# Трассировка лучей: генерируем луч через каждый пиксель изображения

---



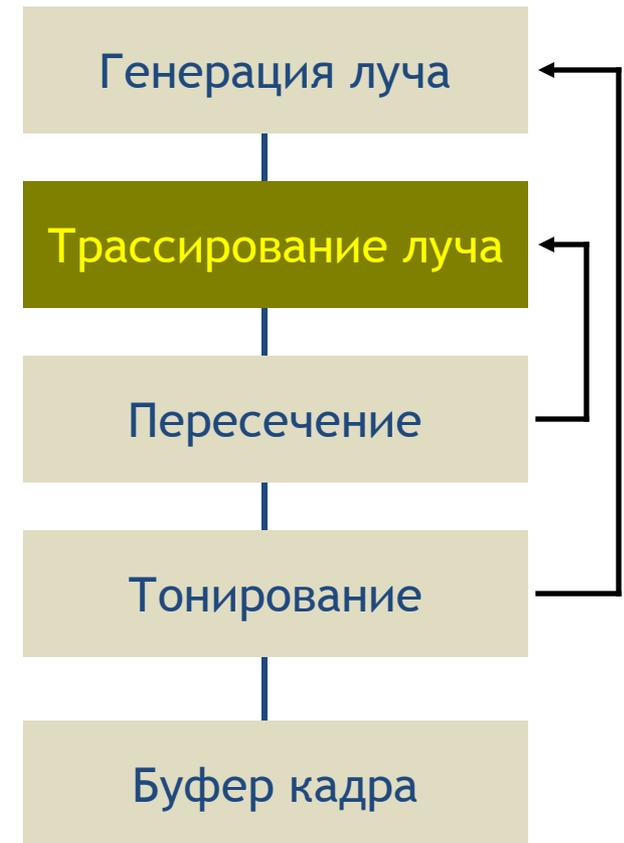
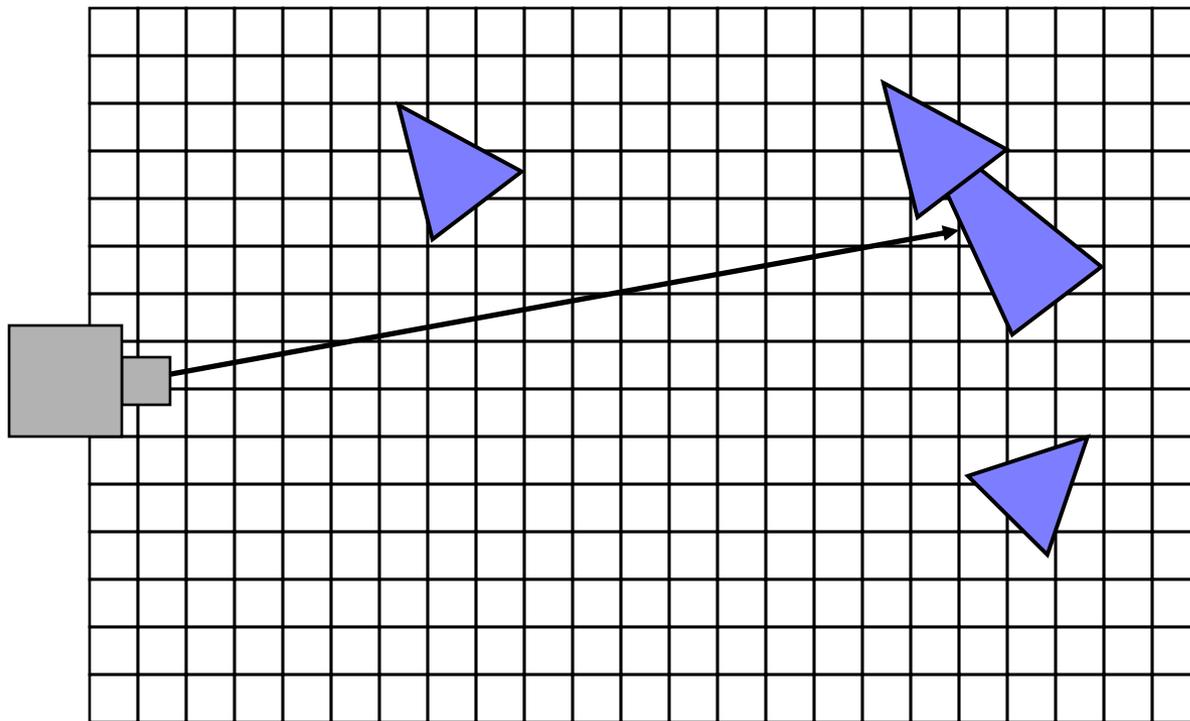
# Трассировка лучей: далее необходимо найти пересечения луча с поверхностью сцены

---

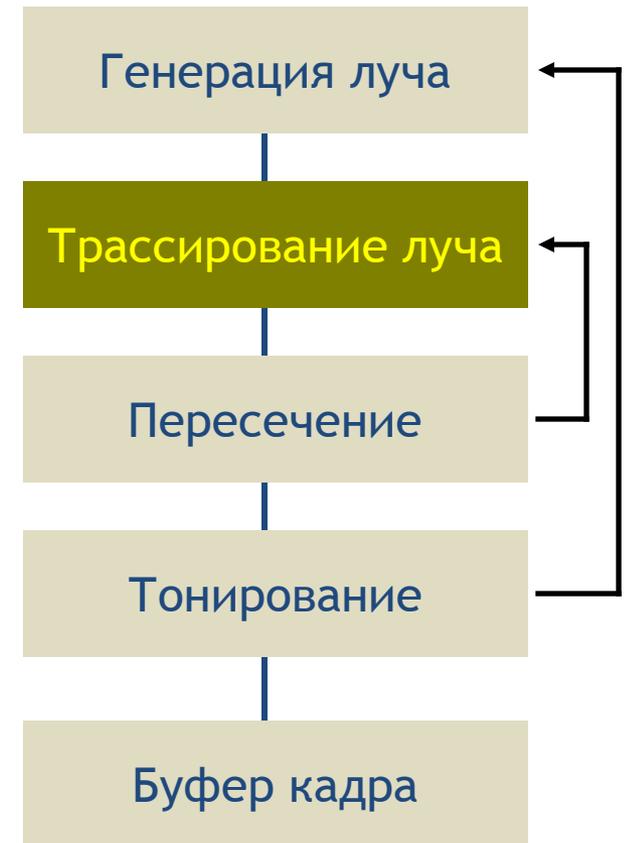
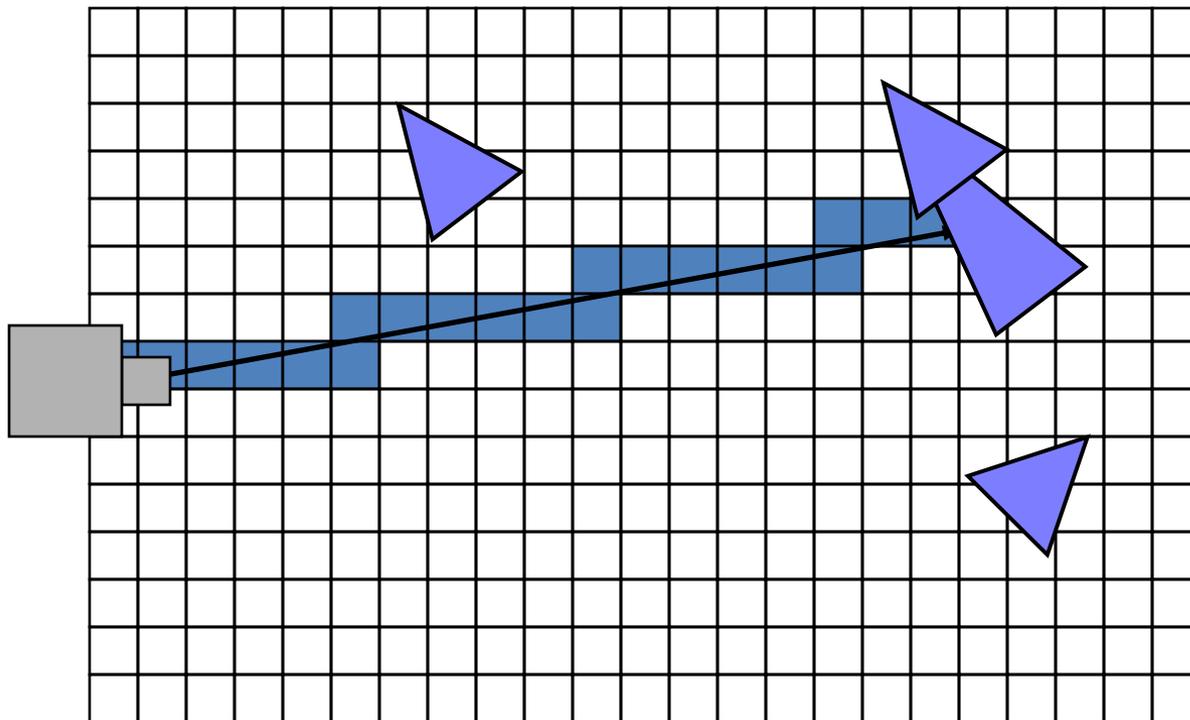


# Пространственные структуры позволяют быстро ограничить множество примитивов

---

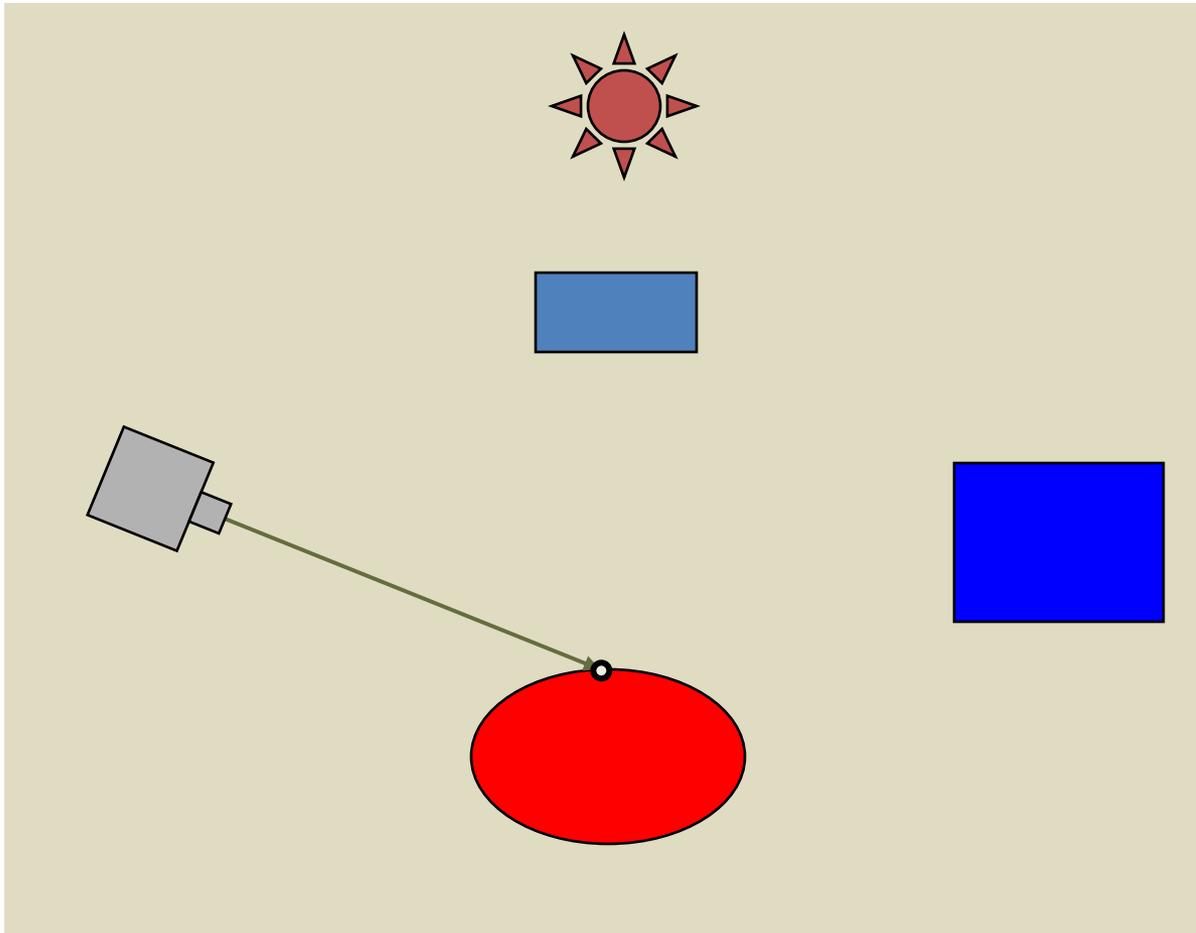


# Пример: трассировка луча по пространственной сетке

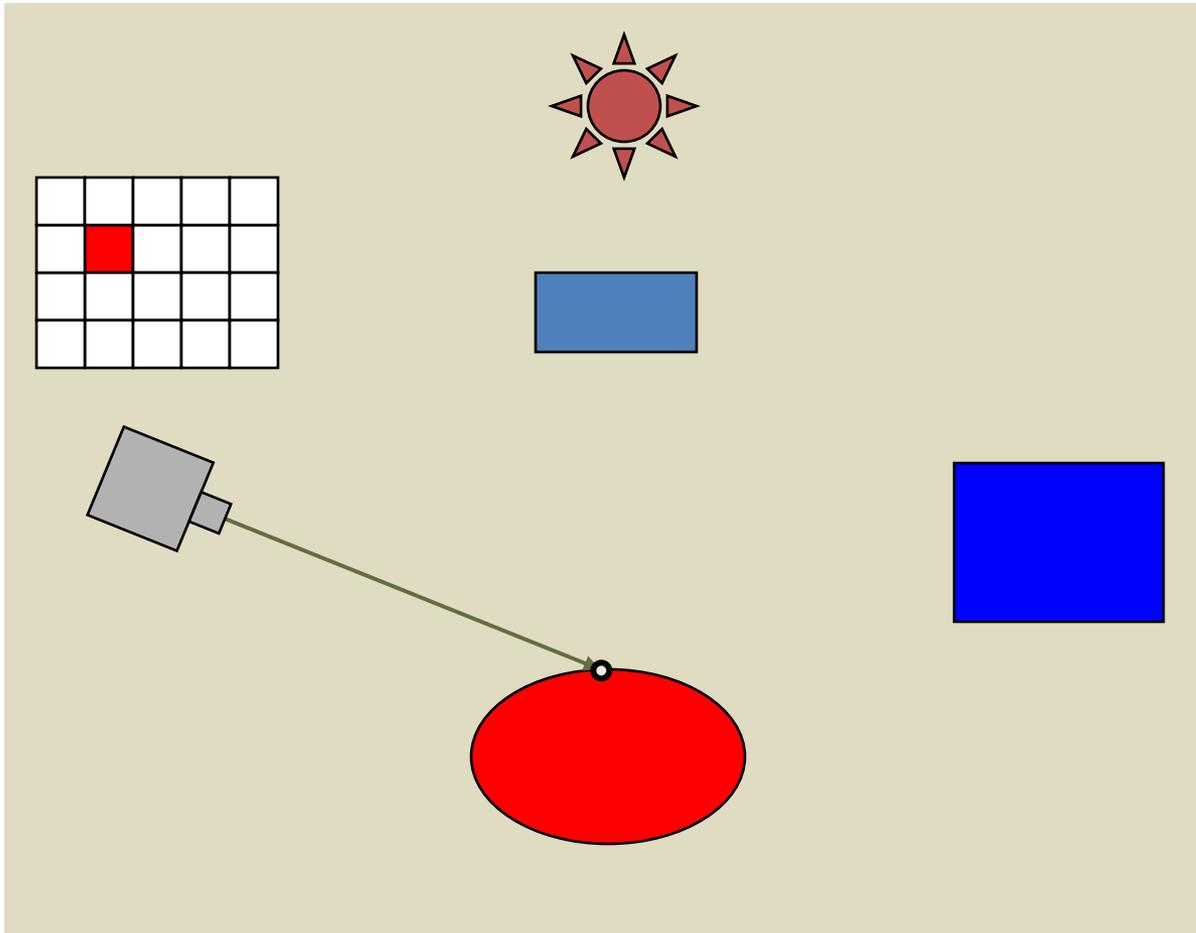


# Трассировка лучей: необходимо найти точное положение пересечения с объектом

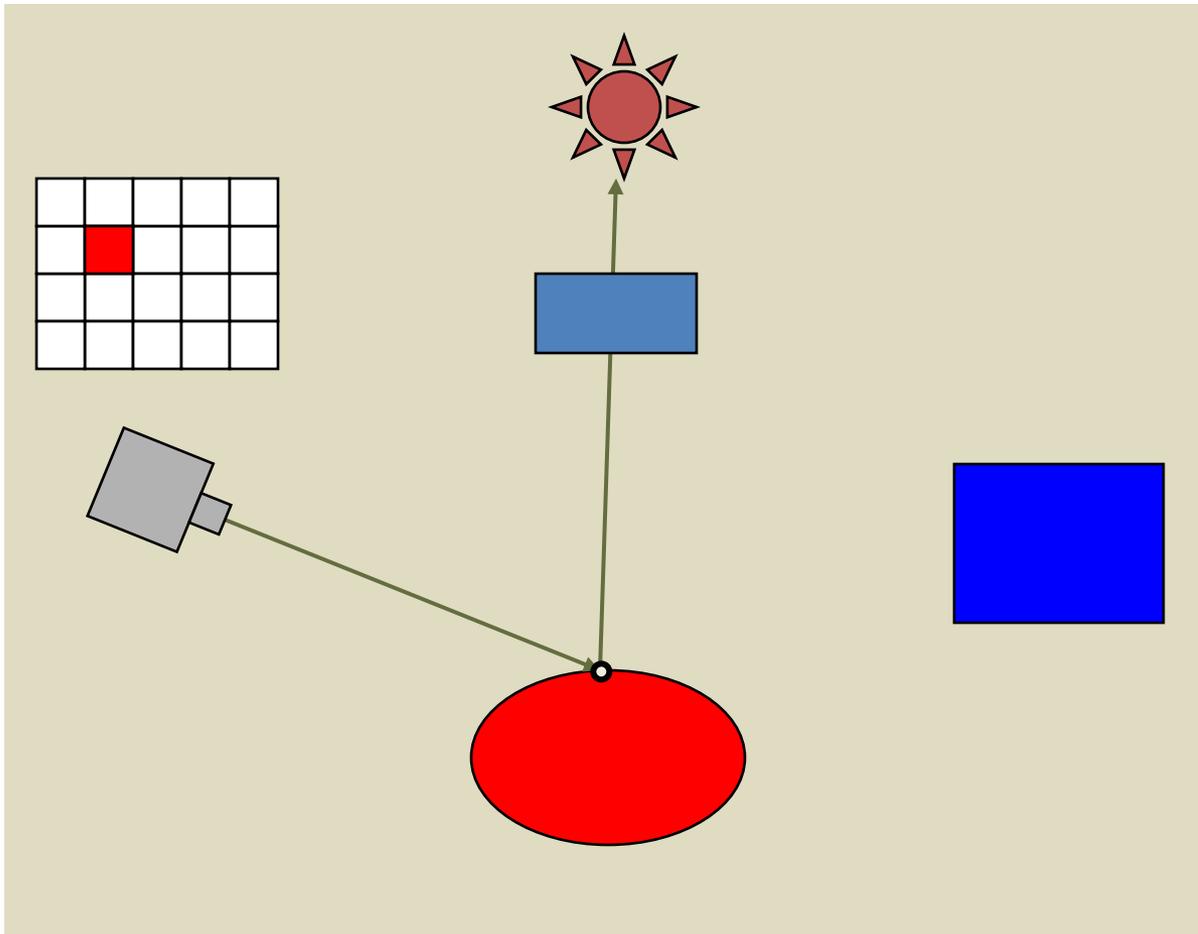
---



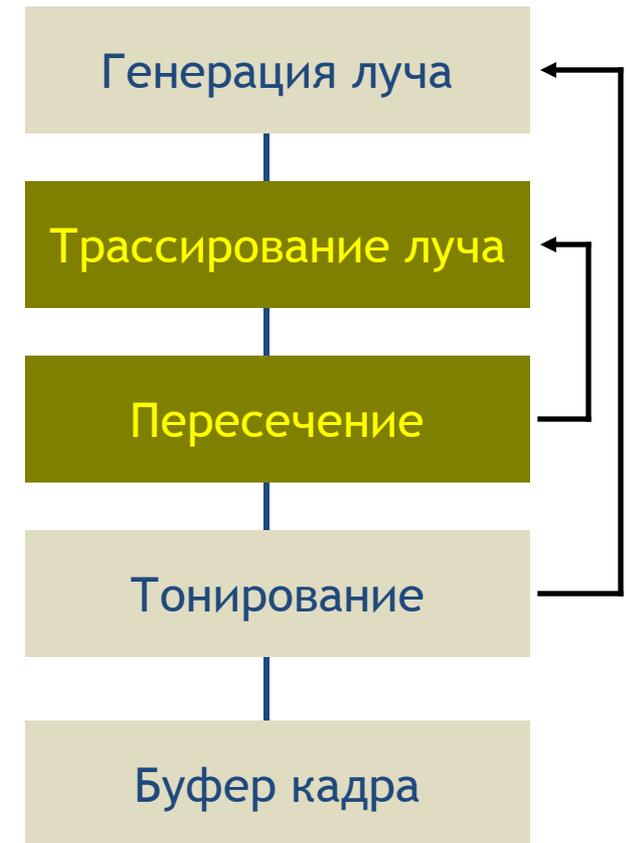
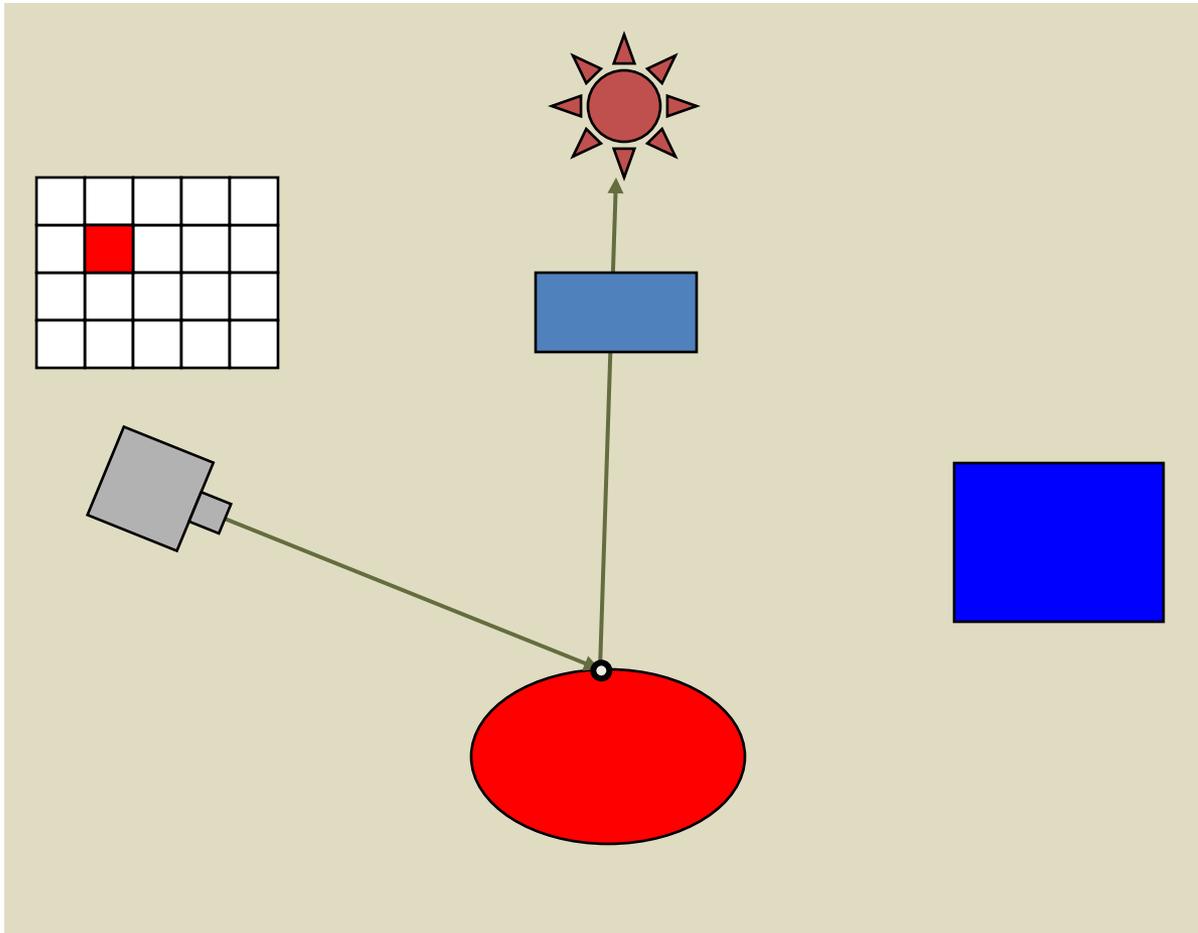
# Трассировка лучей: получение цвета пикселя



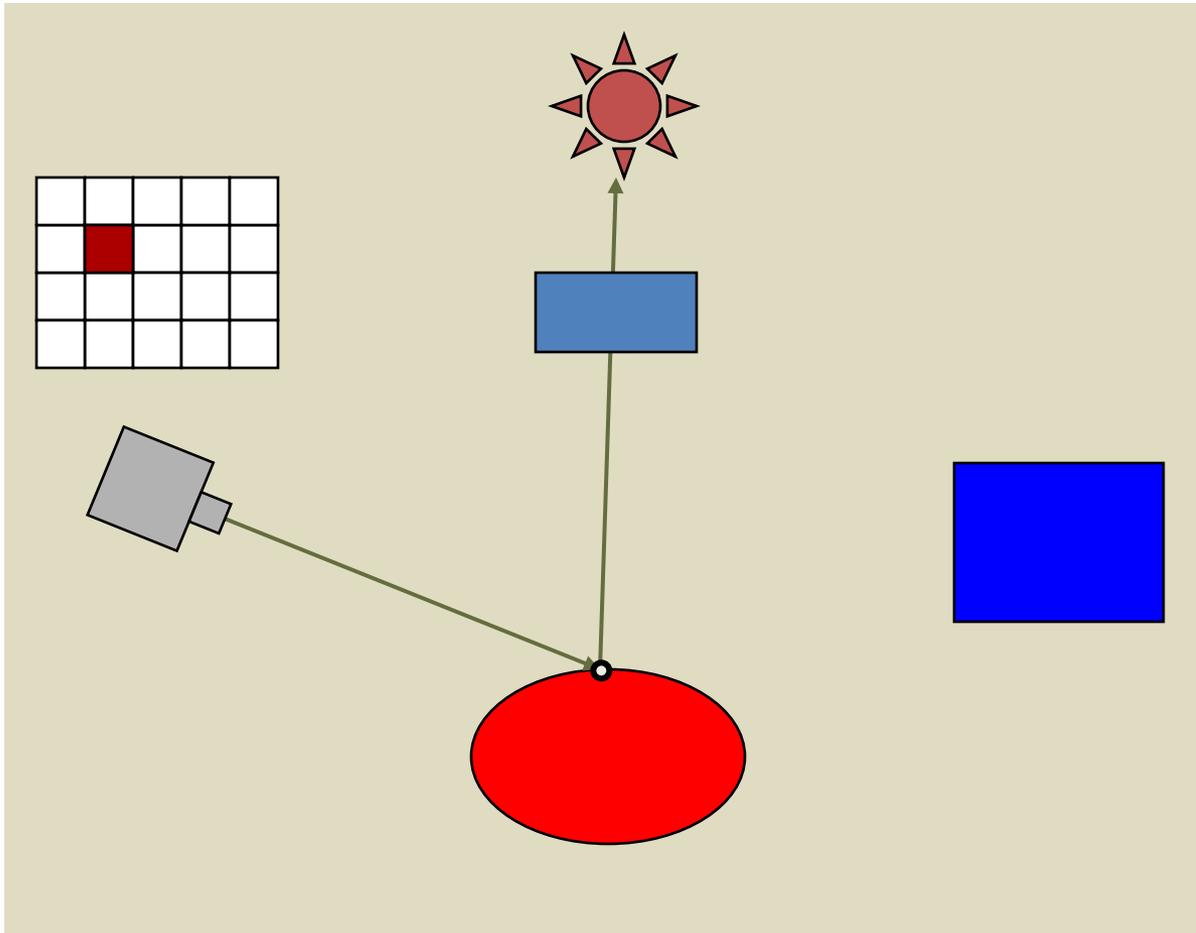
# Трассировка лучей: для получения цвета пикселя бросаем дополнительные лучи



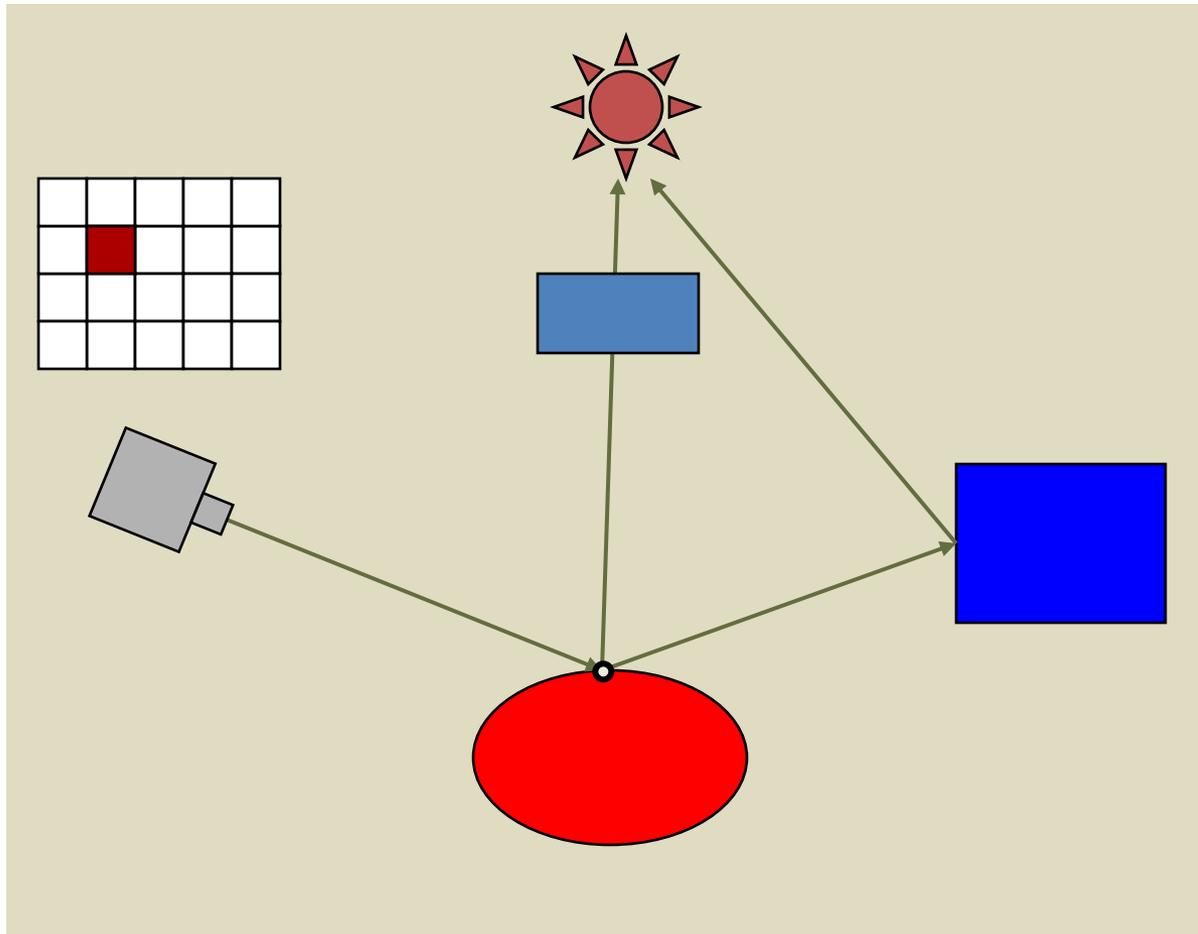
# Трассировка лучей: для получения цвета пикселя бросаем дополнительные лучи



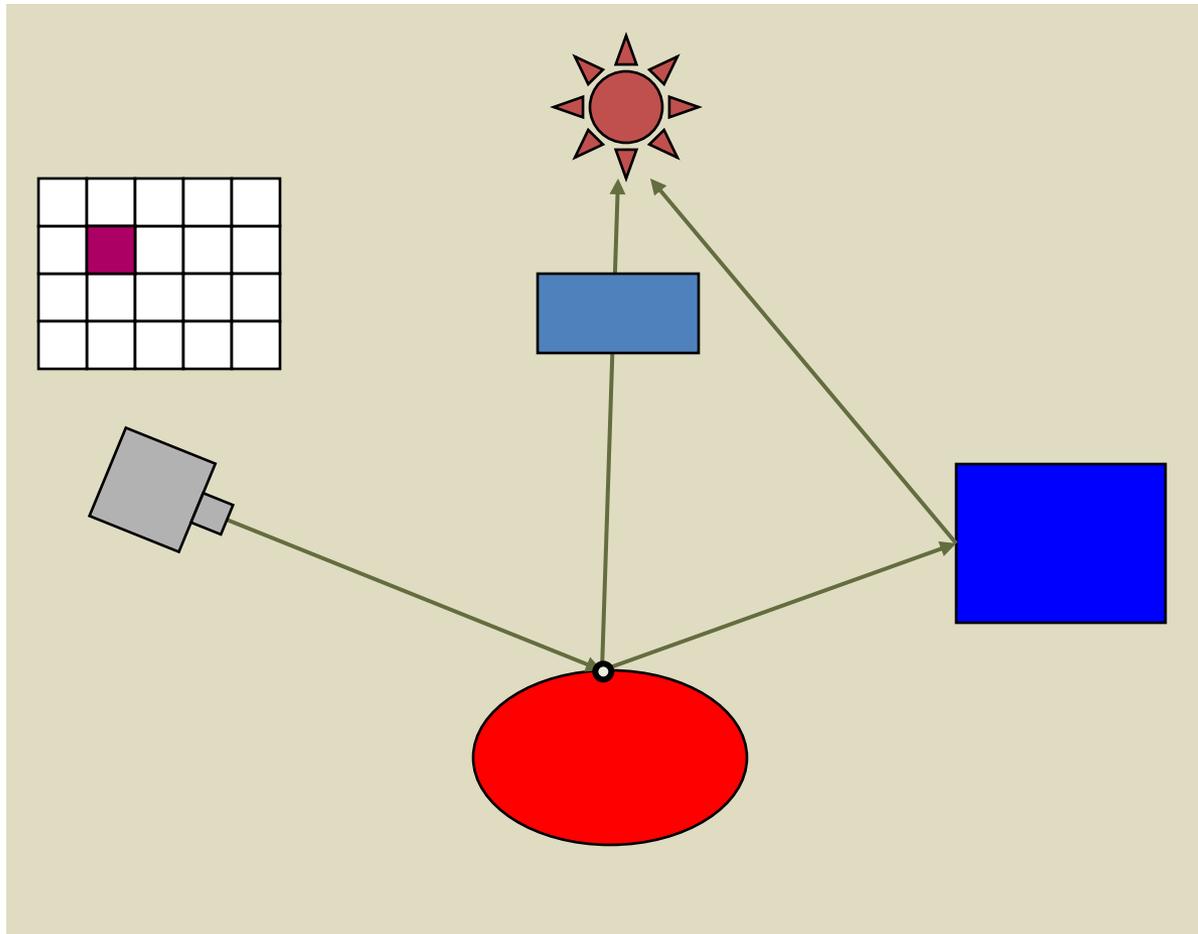
# Трассировка лучей: для получения цвета пикселя бросаем дополнительные лучи



# Трассировка лучей: для получения цвета пикселя бросаем дополнительные лучи

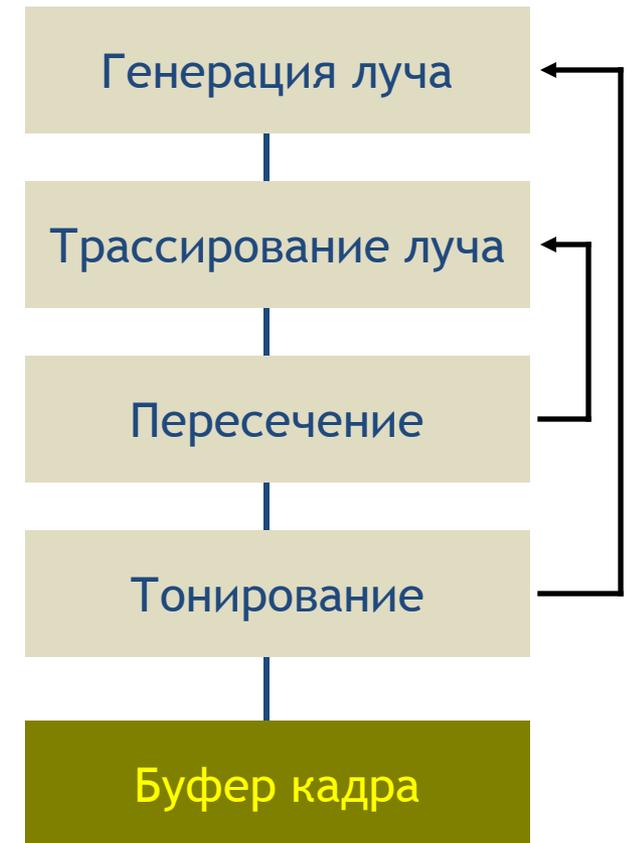
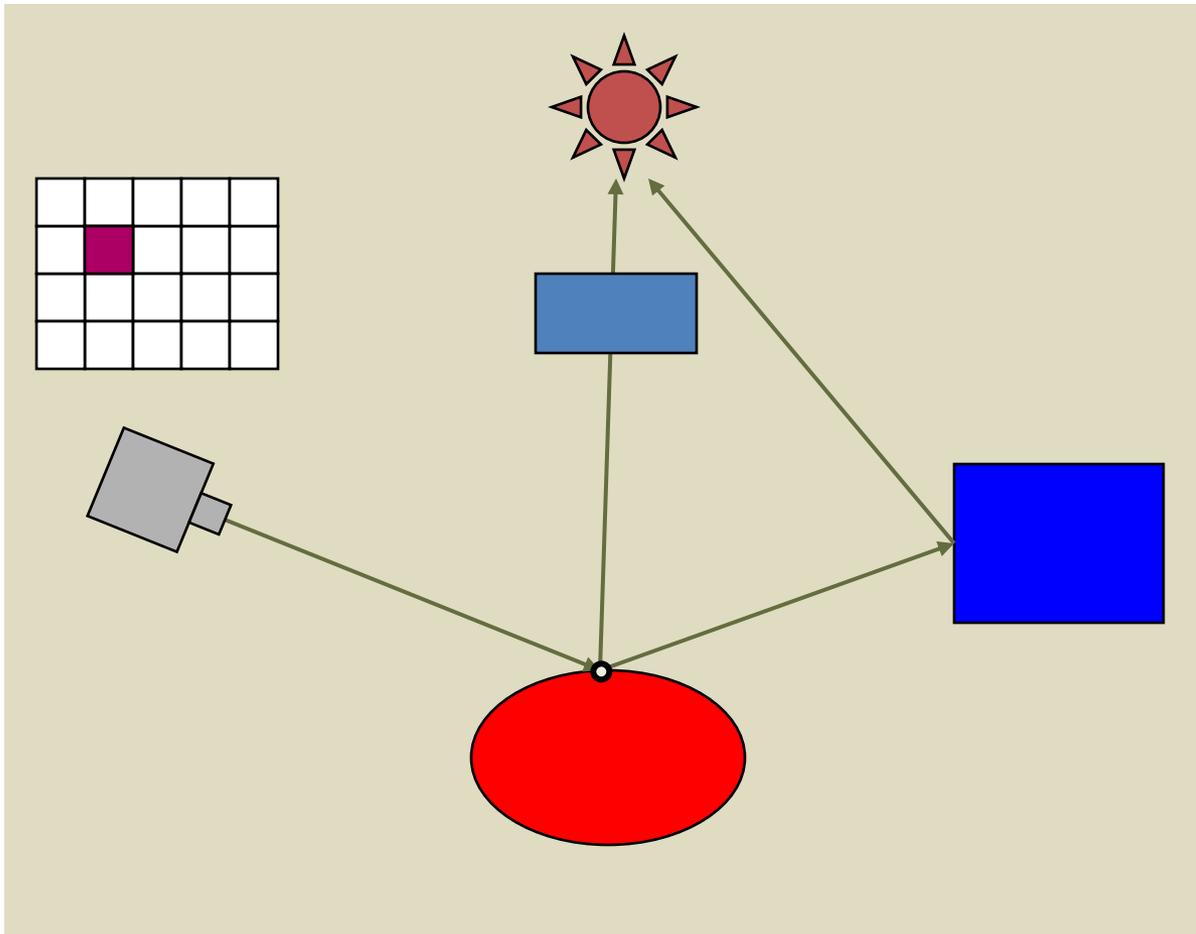


# Трассировка лучей: для получения цвета пикселя бросаем дополнительные лучи



# Трассировка лучей: записываем полученный цвет в пиксель и переходим к следующему

---



# Псевдокод простой трассировки лучей (Whitted)

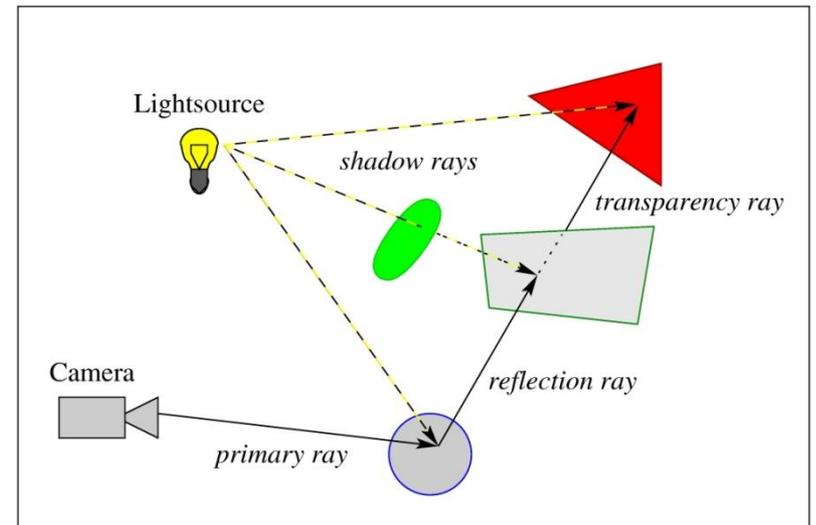
---

```
for each pixel of the screen {
  Final color = 0;
  Ray = { starting point, direction };
  Repeat {
    for each object in the scene {
      determine closest ray object/intersection;
    }
    if intersection exists {
      for each light in the scene {
        if the light is not in shadow of another object {
          add this light contribution to computed color;
        }
      }
    }
    Final color = Final color + computed color * previous reflection factor;
    reflection factor = reflection factor * surface reflection property;
    increment depth;
  } until reflection factor is 0 or maximum depth is reached;
}
```

# Особенности трассировки лучей: глобальное освещение, вычисления по запросу

---

- Глобальное освещение
- Параллелизм
- Расширяемость
- Вычисления только по запросу
- Попиксельные операции



# Ускорение трассировки

---

Пересечение луча со всеми объектами и сортировка для поиска ближайшего пересечения

Очень дорого!

- Ускорение алгоритма пересечения  
Небольшой эффект

- Уменьшение количества пересечений

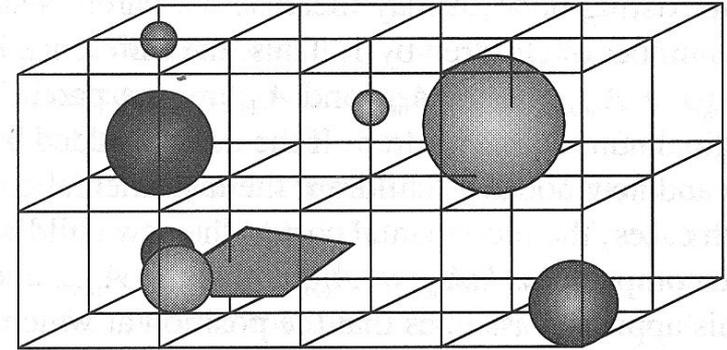
Разбиение пространства (часто иерархическое)

- Сетки, октодеревья, BSD и kd-деревья, деревья ограничивающих объемов
- 5D разбиение (позиция и направление)

# Сетки (Grids): равномерное разбиение пространства

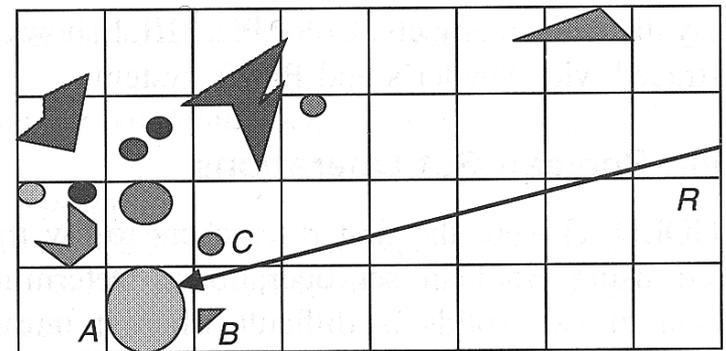
## Построение сетки

- Начинаем с описывающего параллелепипеда
- Треугольники разбиваются по вокселям



## Трассировка

- Алгоритм Брезенхема в 3D
- Останавливаемся, если пересечение найдено в текущем вокселе



# Сетка: проблемы связаны с неадаптивностью разрешения

---

## Обход сетки

- Перечисление вокселей вдоль луча
- Очень простой алгоритм, возможна аппаратная реализация

## Разрешение сетки

- Очень сильно зависит от сцены
- Невозможна адаптация к локальной плотности примитивов
  - Проблема «Чайника на стадионе»
- Возможное решение: иерархические сетки

# Октодерево: «иерархическая сетка»

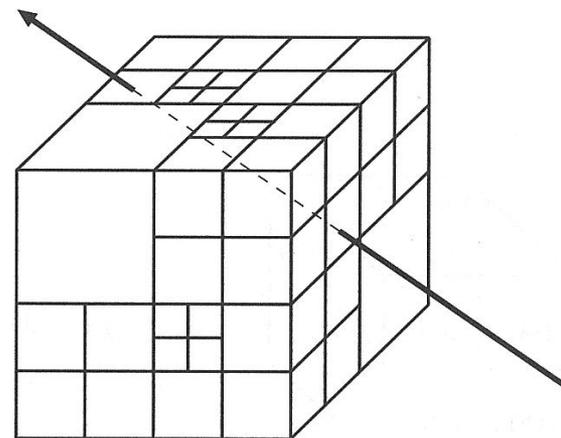
---

## Иерархическое разбиение пространства

- Адаптивное рекурсивное разбиение пространства на 8 равных частей

## Проблемы

- Достаточно сложный алгоритм обхода
- Сложные регионы сходятся медленно

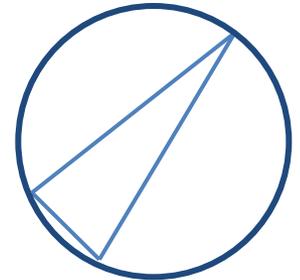


# Описывающие объемы: нерегулярное разбиение пространства

---

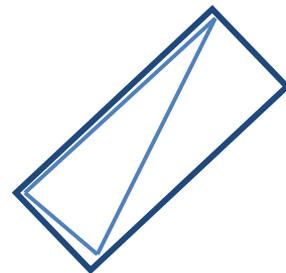
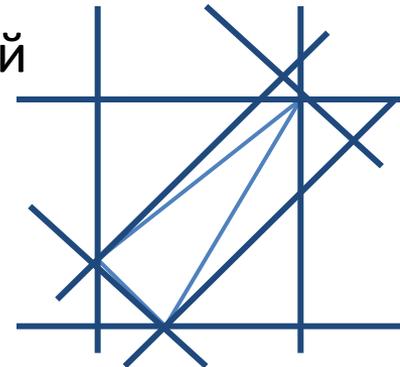
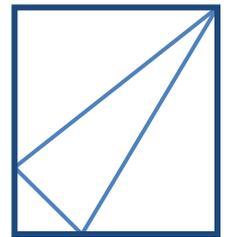
## Идея

- Вычислять пересечение с объектом только если луч пересекает простой описывающий объем



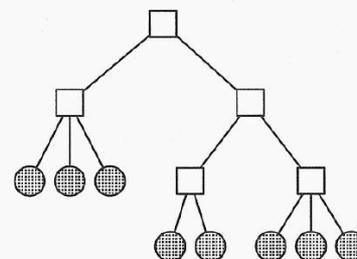
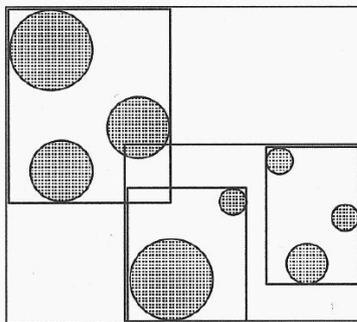
## Возможные описывающие объемы:

- Сфера
- Выровненный по осям описывающий параллелепипед
- Описывающий параллелепипед



# Иерархия описывающих сфер

- Идея:
  - Разбиваем рекурсивно



□ = Bounding Volume

● = Objekt der Szene

## Преимущества:

- Очень хорошая адаптивность
- Эффективный обход  $O(\log N)$

## Проблемы

- Как располагать описывающие объемы?

# BSP- и Kd-деревья: усложнение схемы октодерева для лучшей сходимости

---

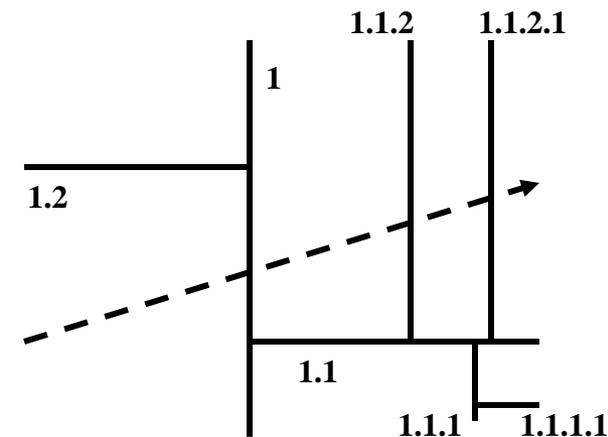
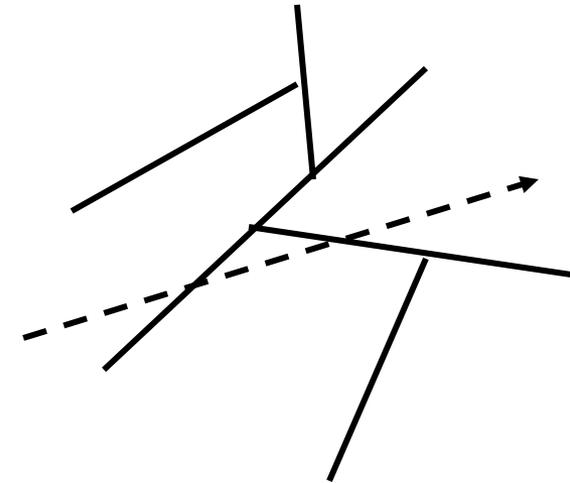
Рекурсивное разбиение пространства  
на полупространства

Двоичное разбиение пространства  
(BSP):

- Разбиение плоскостями в произвольных положениях

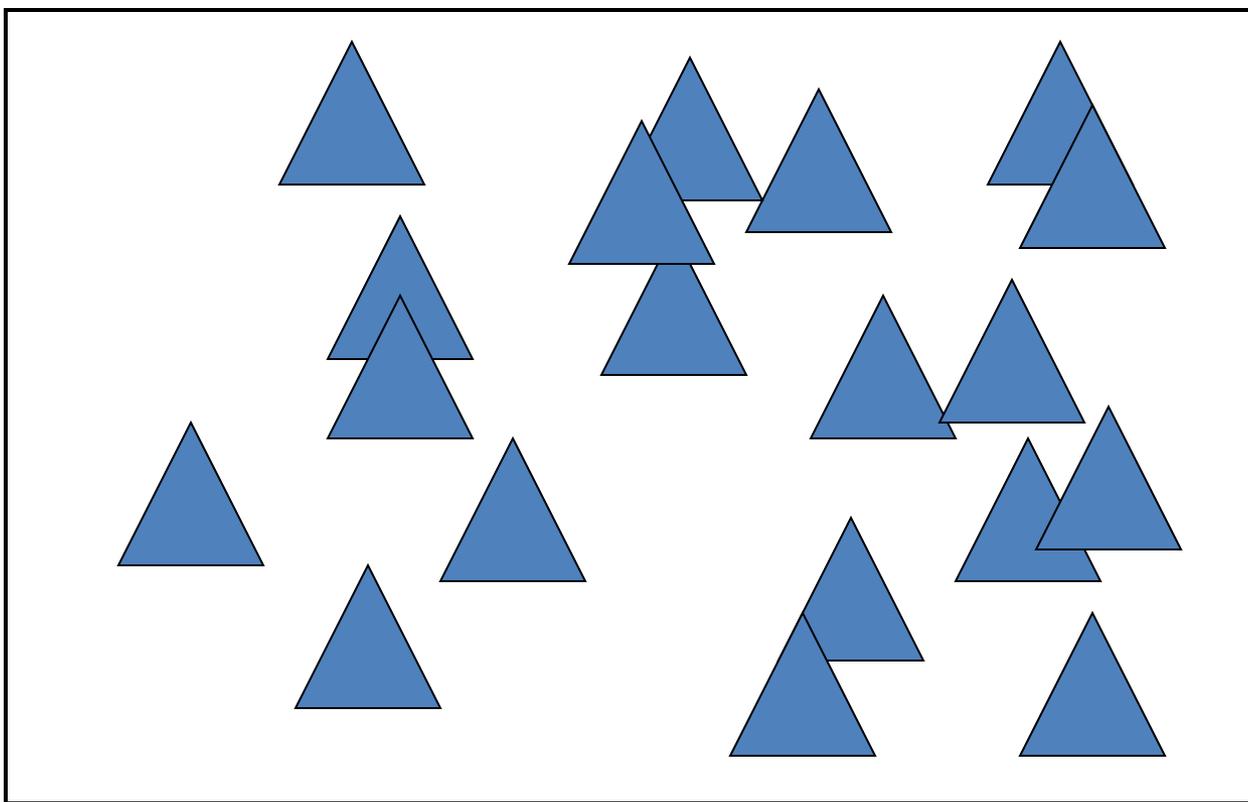
Kd-деревья

- Разбиение выровненными относительно осей плоскостями



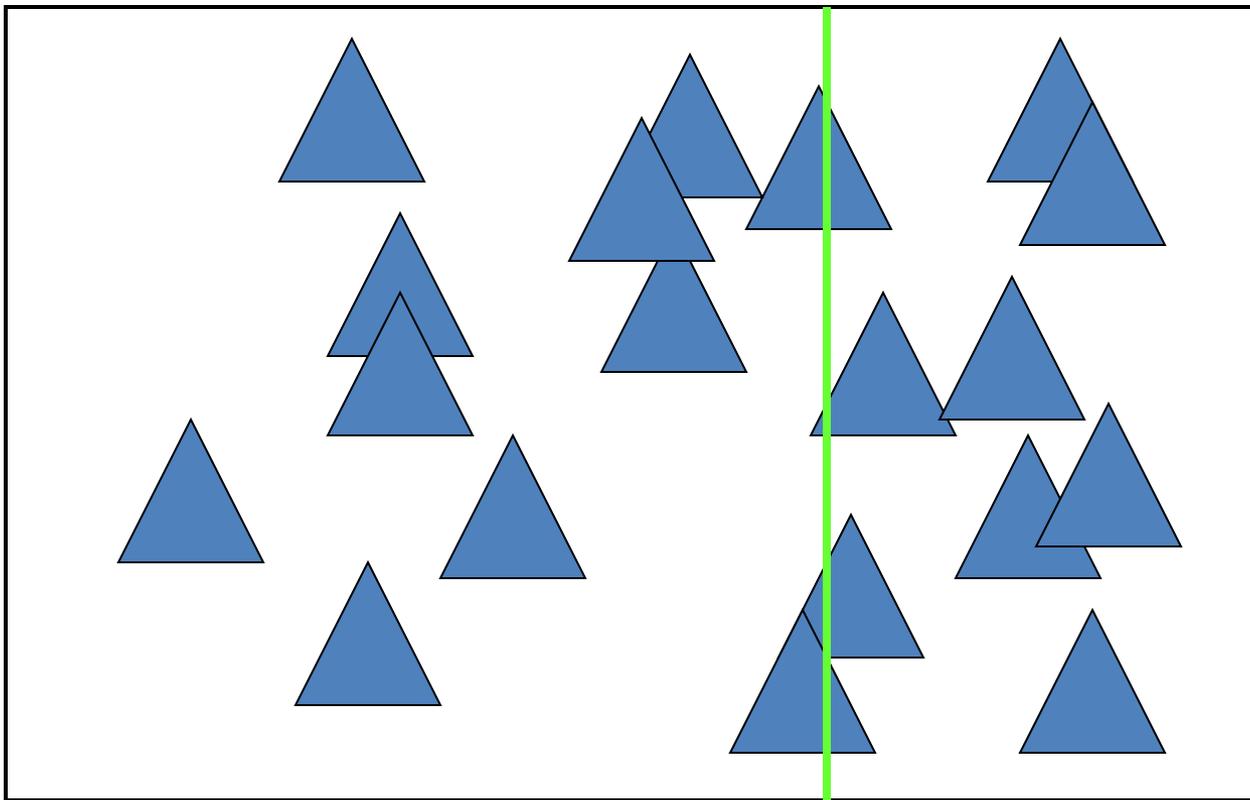
# Построение kD-дерева

---



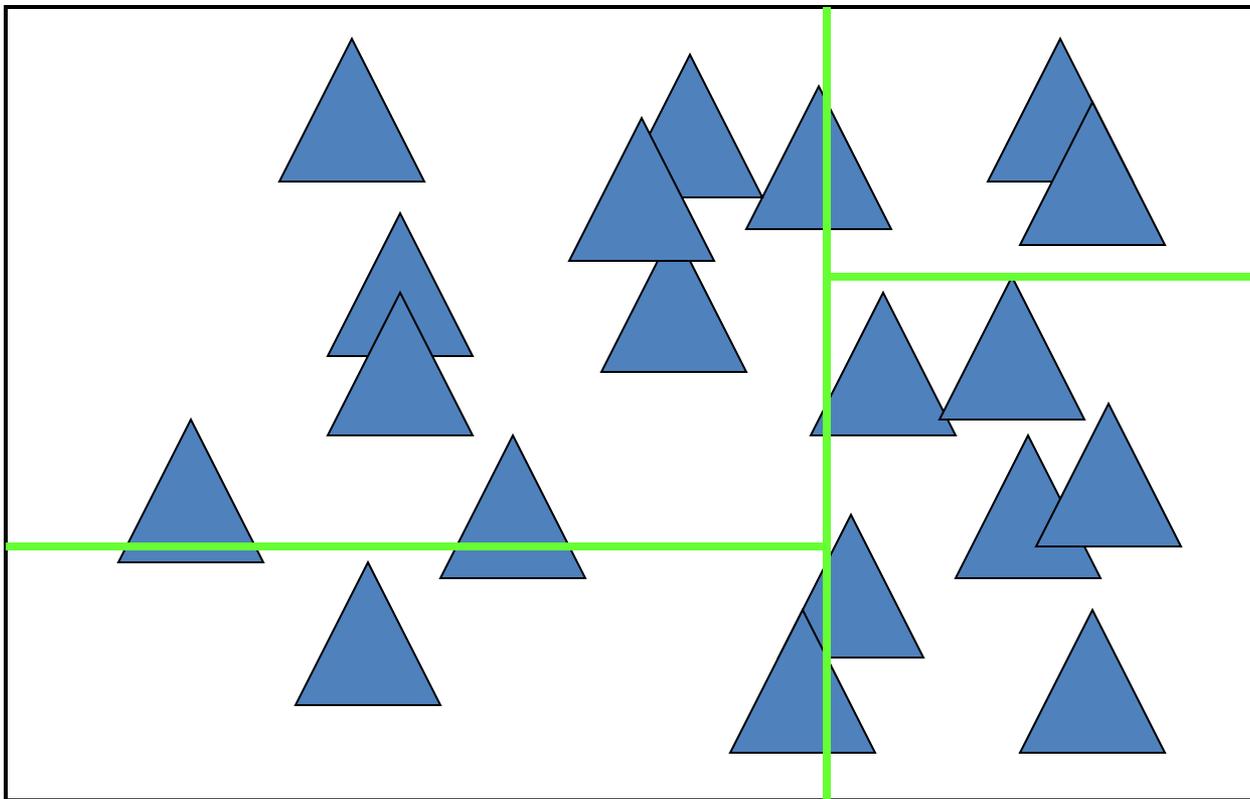
# Построение kD-дерева: уровень 1

---



# Построение kD-дерева : уровень 2

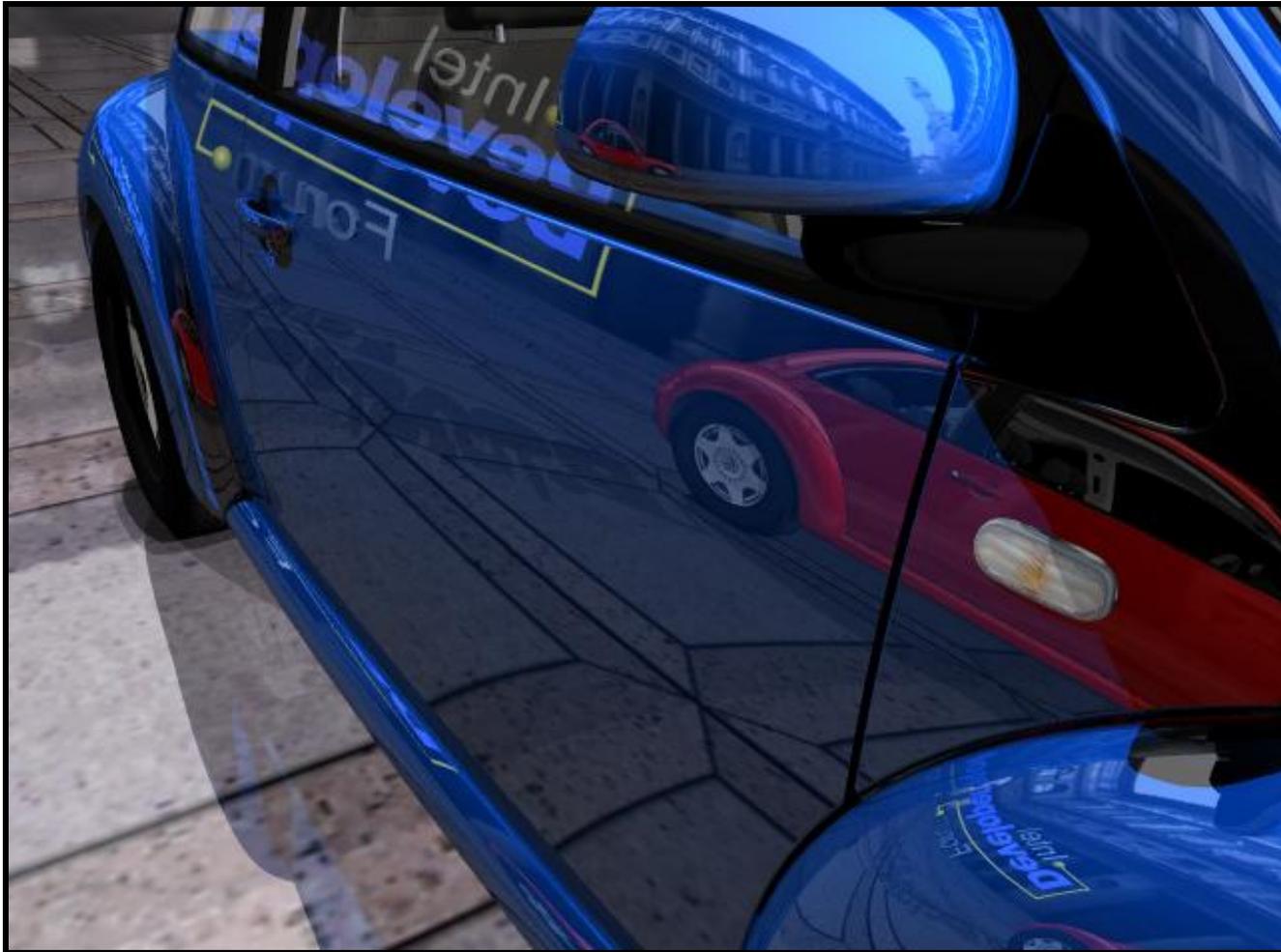
---





# Реалистичные изображения по умолчанию

---



Volkswagen Beetle with correct shadows and (multi-)reflections on curved surfaces

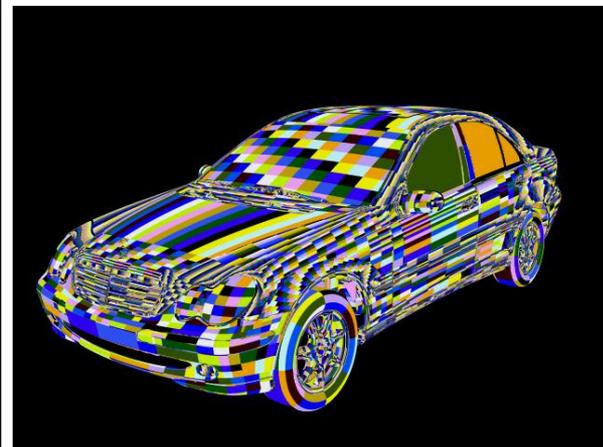
# Физическая корректность



Fully ray traced car head lamp, faithful visualization requires up to 50 rays per pixel

# Физическая корректность

---



Rendered directly from trimmed NURBS surfaces, with smooth environment lighting

# Физическая корректность

---



Rendered with accurately measured BTF data that accounts for micro lighting effects

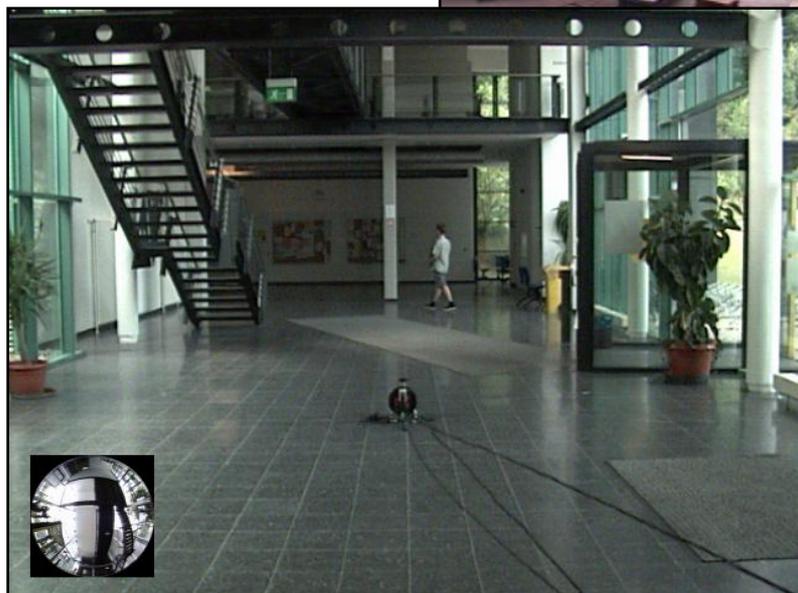
Textured Phong for comparison



BTF Data Courtesy R. Klein, Uni Bonn

# Физическая корректность

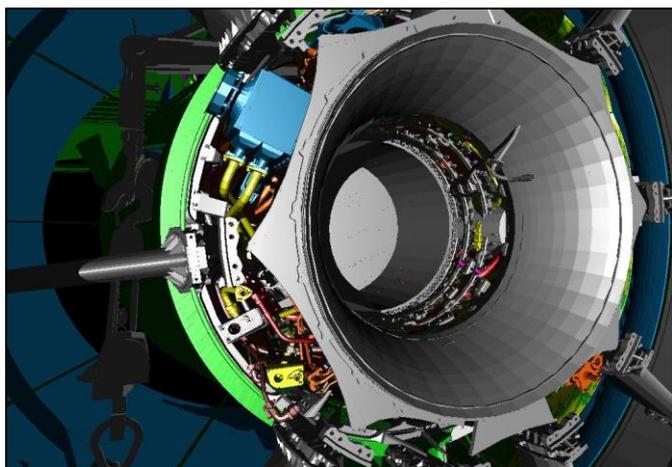
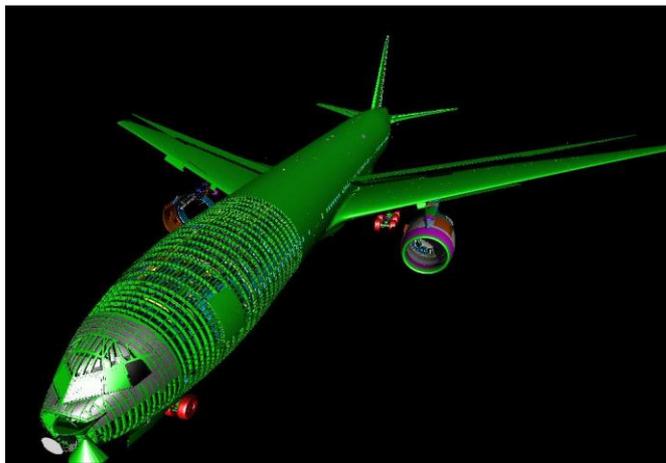
---



VR scene illuminated from realtime video feed, AR with realtime environment lighting

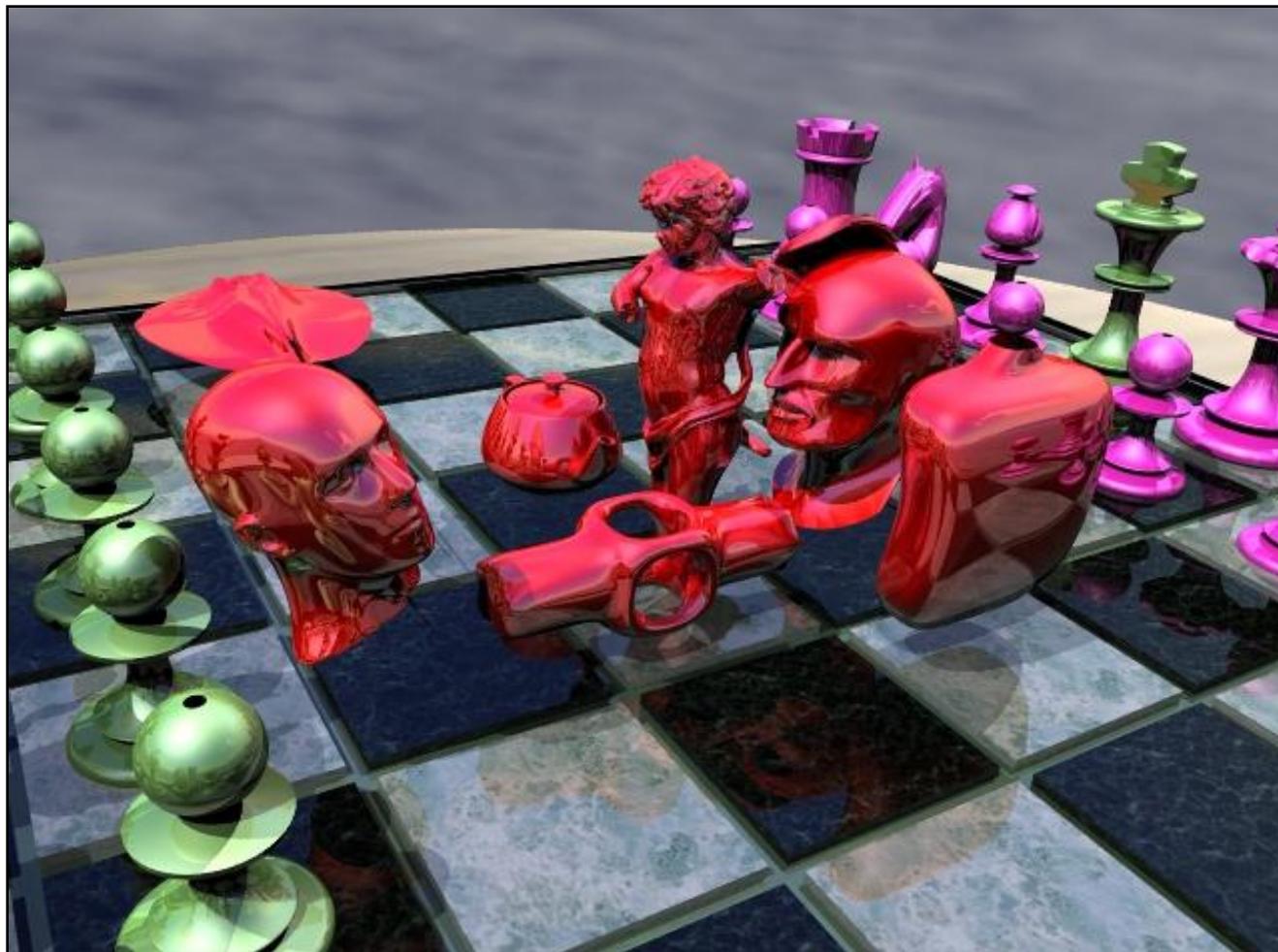
# Поддержка массивных сцен

---



# Интеграция различных типов примитивов

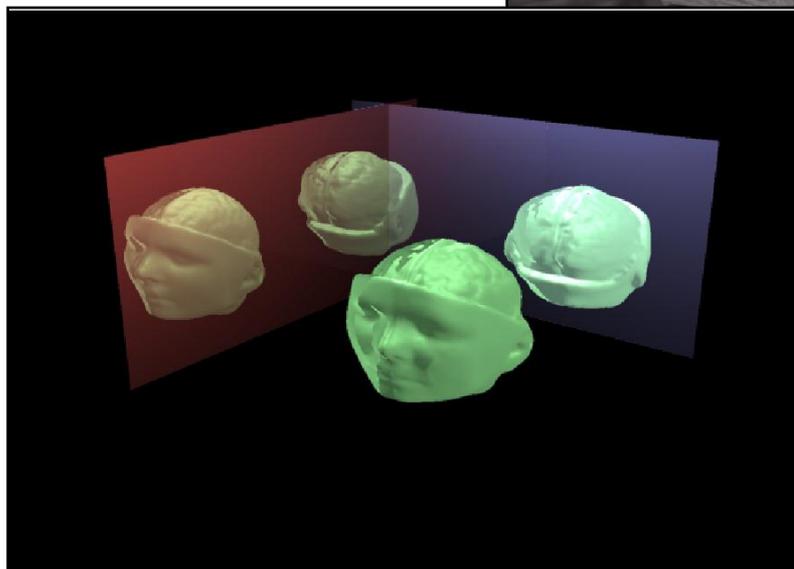
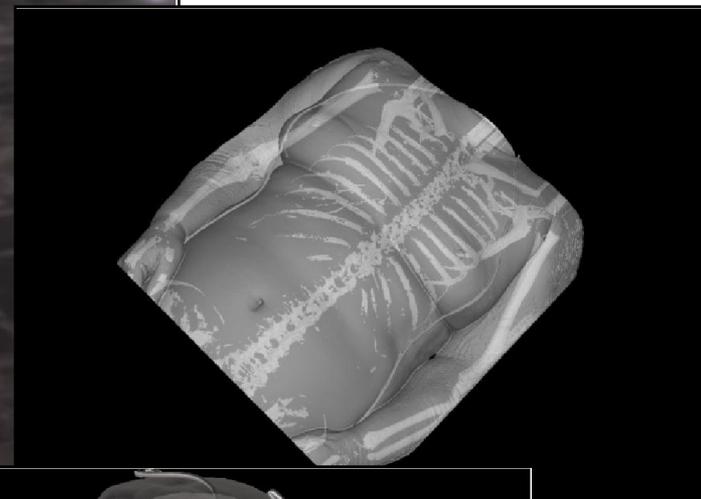
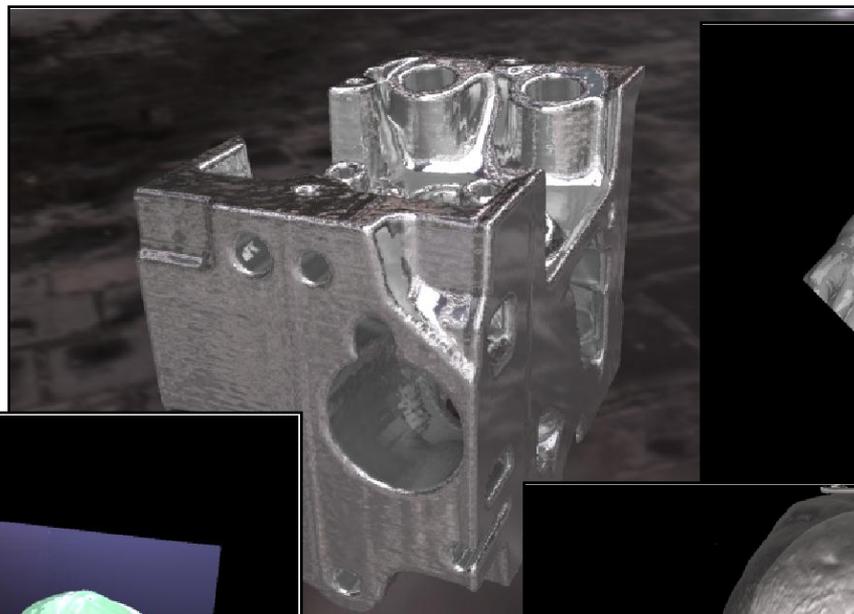
---



Triangles, Bezier splines, and subdivision surfaces fully integrated

# Интеграция различных типов примитивов

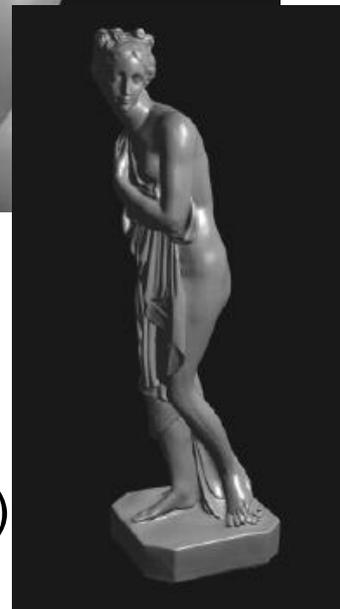
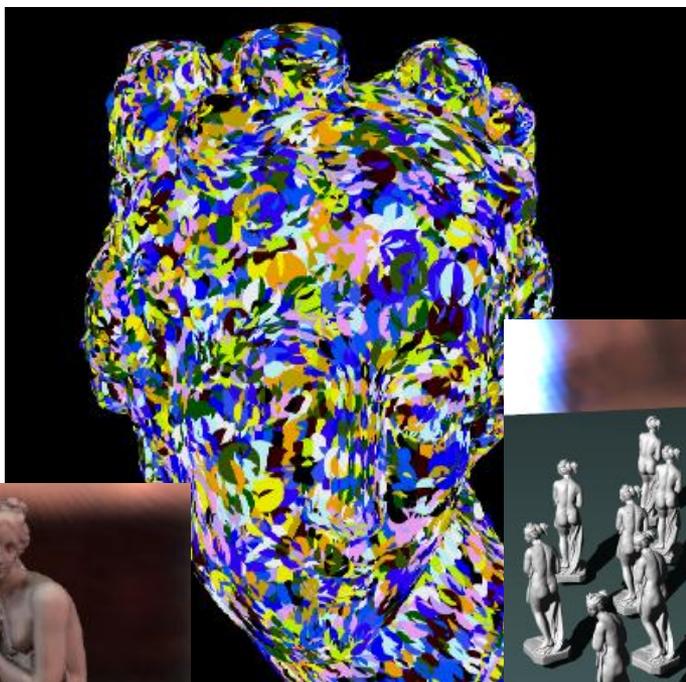
---



Volume visualization using multiple iso-surfaces in combination with surface rendering

# Интеграция различных типов примитивов

---



24 MPoints, 2.1 fps with shadow @ 640x480

Realtime ray tracing of point clouds (1 Mpoints each)  
On *one* dual-Opteron 2.4 GHz: 4-9 fps

# Лекция посвящена обзору двух подходов к синтезу изображений: излучательности и трассировке лучей

---



Понятие о глобальном освещении и других сложных оптических эффектах



Трассировка лучей



Излучательность

# Трассировка и Излучательность

---

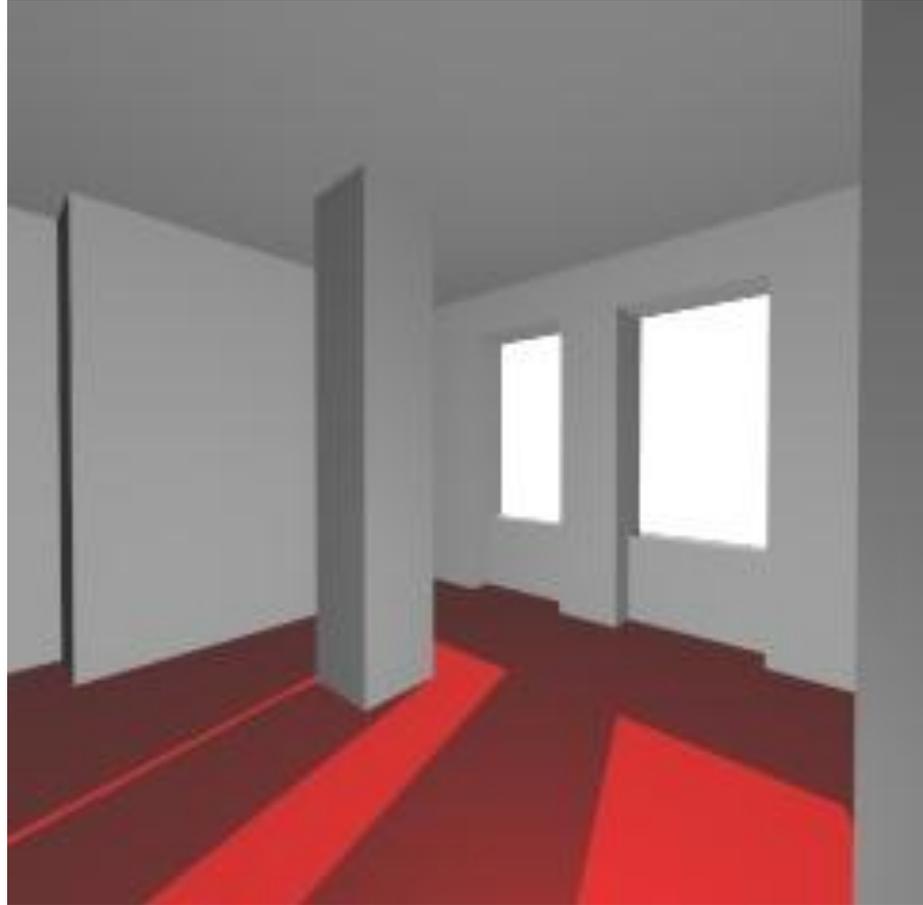
- **Проблема трассировки лучей** – для вычисления глобального освещения нужно трассировать большое количество лучей
  - Вычислительно сложная задача
- **Основная идея излучательности** – сохранение светимости поверхностей по мере продвижения света от источников



Трассировка



Излучательность



# Излучательность = Энергетическая СВЕТИМОСТЬ

---

- Полная энергия, покидающая поверхность единичной площади
- Обозначение:  $V$  ( $=M$ )
- Единицы измерения:  $Вт/м^2$

$$M = \frac{d\Phi}{dS}$$

# Предположения для решения задачи излучательности. Ламбертовы поверхности

---

- Алгоритмы излучательности решают уравнение глобального освещения при наборе ограничивающих предположений:
  - Все поверхности ламбертовы (идеально диффузные)
  - Поверхности могут быть поделены на участки (патчи) константной излучательности
  - Излучательность рассчитывается только на поверхностях
    - Требуются дополнительные действия по построению изображения
  - Расчет производится для замкнутой системы
    - Энергия не пропадает
- Эти предположения позволяют сделать уравнение глобального освещения линейным!

# Излучательность диффузных поверхностей: простая форма, простая ДФО

---

- Излучательность для диффузных поверхностей

$$B(\mathbf{x}) = \int_{\Omega} L_o(\mathbf{x}) \cos \theta d\omega_o = \pi L_o(\mathbf{x})$$

- Диффузная ДФО в терминах коэффициента отражения поверхности:

$$f(\mathbf{x}) = \frac{\rho}{\pi}$$

# Можно упростить интеграл освещенности

$$f(\mathbf{x}) = \frac{\rho}{\pi}$$

Упрощение формулы глобальной освещенности:

$$L(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} f(\mathbf{x}, \omega_o, \omega) L(\mathbf{x}, \omega) \cos \theta d\omega$$

$$\pi L(\mathbf{x}) = \pi L_e(\mathbf{x}) + \pi \int_{\Omega} \frac{\rho(\mathbf{x})}{\pi} L(\mathbf{x}, \omega) \cos \theta d\omega$$

$$B(\mathbf{x}) = E(\mathbf{x}) + \rho(\mathbf{x}) \int_{\Omega} L(\mathbf{x}, \omega) \cos \theta d\omega$$

По-прежнему есть интеграл для вычисления освещенности...

# Будем рассчитывать интеграл по точкам поверхностей вместо направлений

---

Преобразуем интеграл по телесному углу в интеграл по всем точкам поверхностей сцены

$$d\omega = \frac{\cos \theta' dA}{r^2}$$

$\theta'$  – угол между нормалью площадки

$A$  - дифференциальная площадь

$$V(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{если } \mathbf{x} \text{ и } \mathbf{y} \text{ взаимно видны} \\ 0 & \text{в противном случае} \end{cases}$$

$$B(\mathbf{x}) = E(\mathbf{x}) + \rho(\mathbf{x}) \int_{\mathbf{y} \in S} B(\mathbf{y}) \frac{\cos \theta \cos \theta'}{\pi r^2} V(\mathbf{x}, \mathbf{y}) dA$$

# Чтобы преобразовать интеграл в сумму, необходимо представить сцену как набор поверхностей

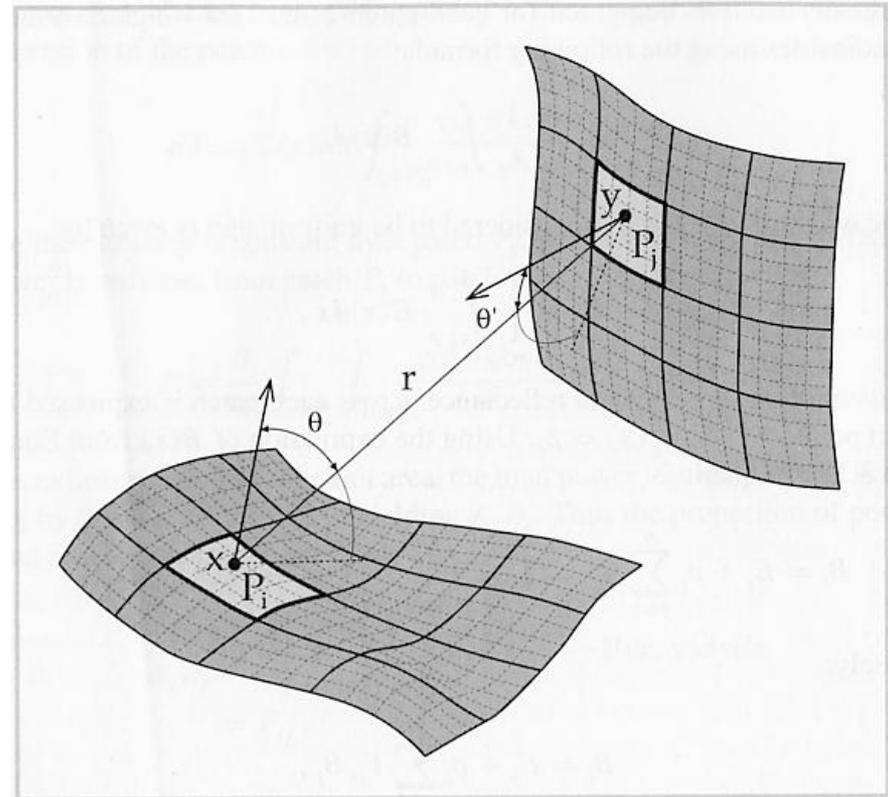
---

Предположим, что геометрия разбита на  $N$  непересекающихся поверхностей (patches)  $P_i$ ,  $i=1..N$

Площадь каждой  $A_i$

Предположим, что излучательность константна на каждом

$$B_i = \frac{1}{A_i} \int_{\mathbf{x} \in P_i} B(\mathbf{x}) dx$$



# Уравнение освещенности для дискретного диффузного случая – продолжаем упрощение

---

Заменяем интеграл по точкам геометрии на сумму по поверхностям

$$B(\mathbf{x}) = E(\mathbf{x}) + \rho_{hd}(\mathbf{x}) \sum_{j=1}^N \int_{y \in P_j} B(\mathbf{y}) \frac{\cos \theta \cos \theta'}{\pi r^2} V(\mathbf{x}, \mathbf{y}) dy$$

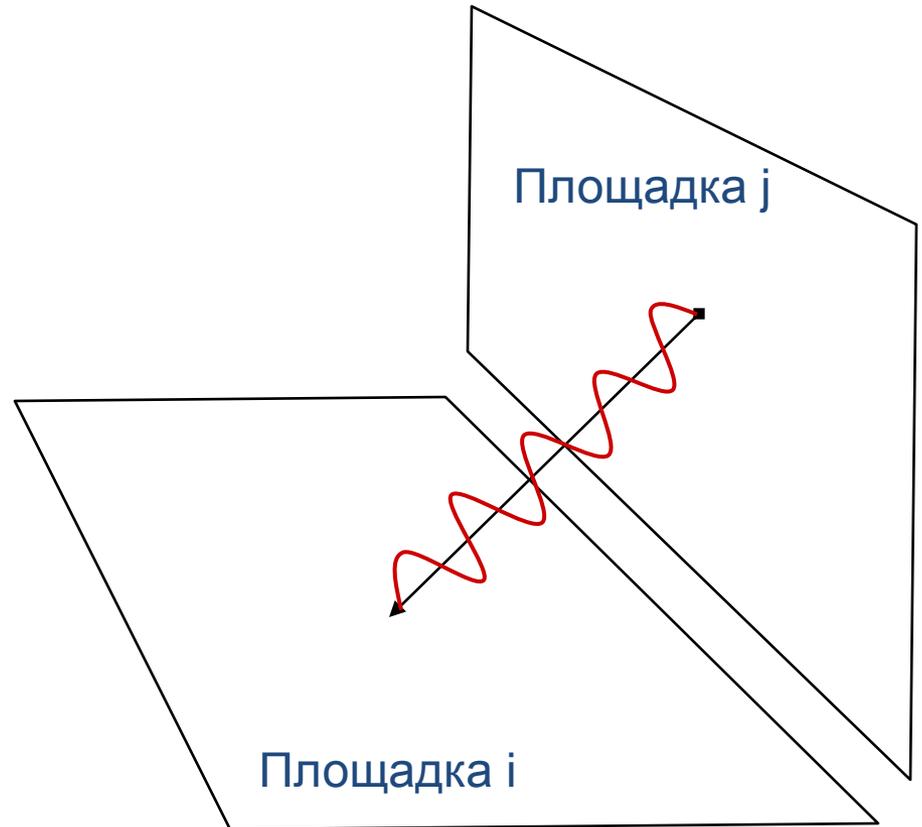
$$\frac{1}{A_i} \int_{\mathbf{x} \in P_i} B(\mathbf{x}) d\mathbf{x} = \frac{1}{A_i} \int_{\mathbf{x} \in P_i} \left[ E(\mathbf{x}) + \rho_{hd}(\mathbf{x}) \sum_{j=1}^N \int_{y \in P_j} B(\mathbf{y}) \frac{\cos \theta \cos \theta'}{\pi r^2} V(\mathbf{x}, \mathbf{y}) dy \right] d\mathbf{x}$$

$$B_i = E_i + \rho_i \sum_{j=1}^N B_j \frac{1}{A_i} \int_{\mathbf{x} \in P_i} \int_{y \in P_j} \frac{\cos \theta \cos \theta'}{\pi r^2} V(\mathbf{x}, \mathbf{y}) dy d\mathbf{x}$$

# Форм-фактор: зависит только от размера и расположения поверхностей

$$F_{ij} = \frac{1}{A_i} \int_{x \in P_i} \int_{y \in P_j} \frac{\cos \theta \cos \theta'}{\pi r^2} V(x, y) dy dx$$

$F_{ij}$  - часть полной энергии, покинувшей площадку  $P_i$  и полученной площадкой  $P_j$



Уравнение для дискретной излучательности  
может быть записано в виде СЛАУ

---

$$B_i = E_i + \rho_i \sum_{j=1}^N B_j F_{ij}$$

Это СЛАУ!

$$\mathbf{B} = \mathbf{E} + \rho \mathbf{B} \mathbf{F}$$

$$\mathbf{E} = \mathbf{M} \mathbf{B} \quad \text{где} \quad \mathbf{M} = (\mathbf{I} - \rho \mathbf{F})$$

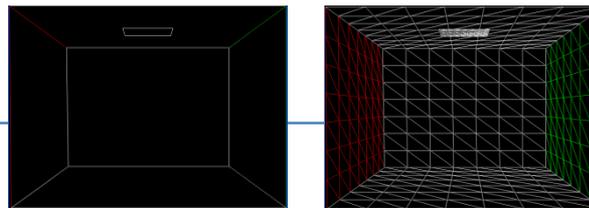
Размеры  $M$  :  $N \times N$

- Большая система 😊
- Но матрица  $M$  имеет некоторые особенности,  
упрощающие вычисление

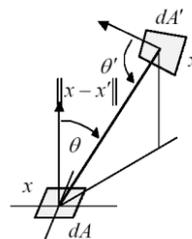
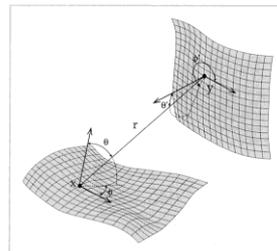


# Излучательность

Разбиваем геометрию на площадки



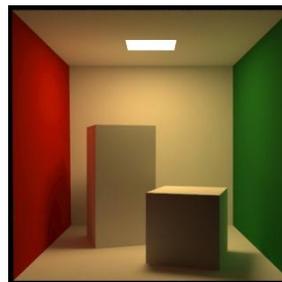
Вычисляем форм-факторы



Решаем СЛАУ

$$\begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix} \mathbf{x} = \begin{bmatrix} \times \\ \times \end{bmatrix}$$

Реконструкция и показ решения

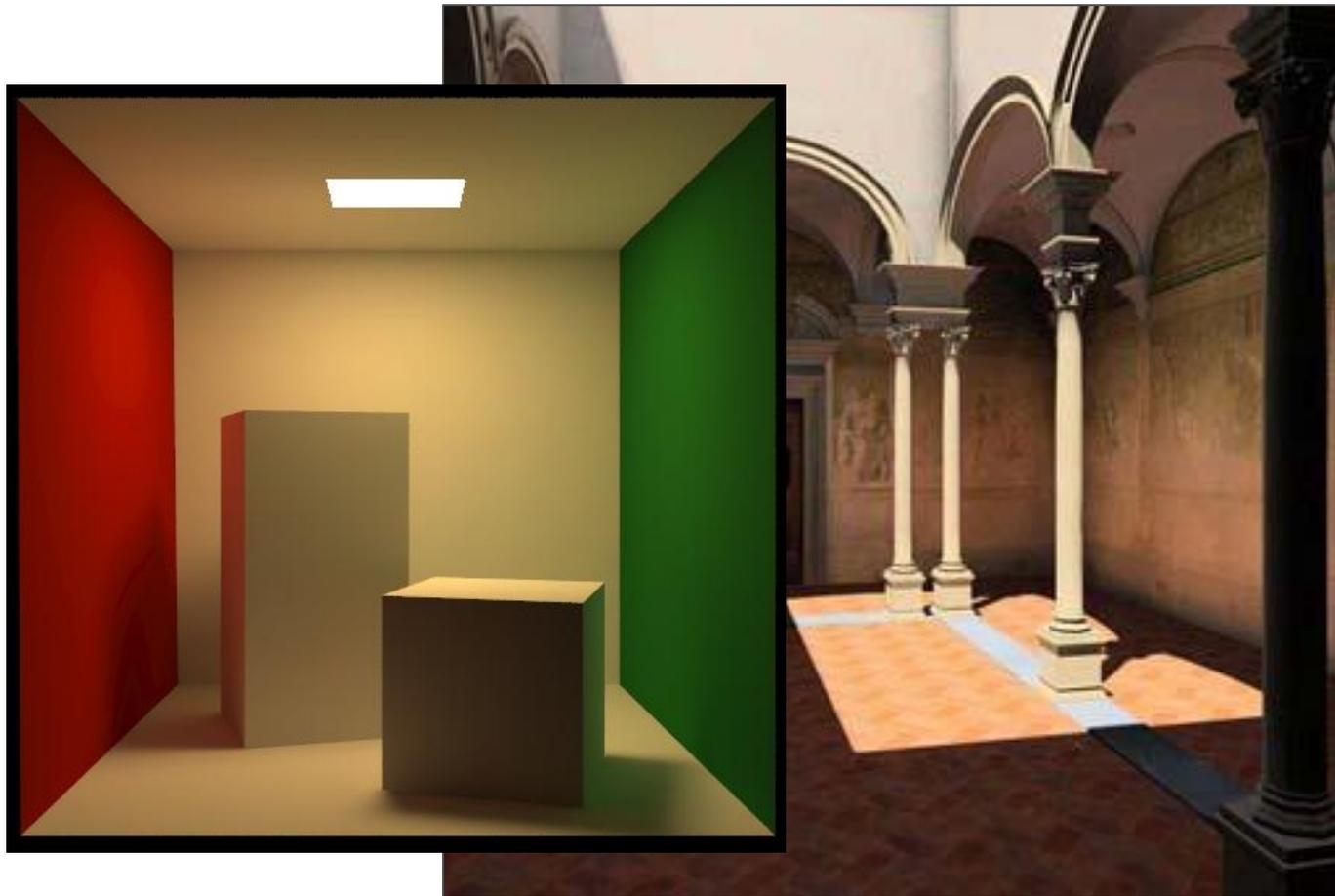






# Результаты работы метода излучательности

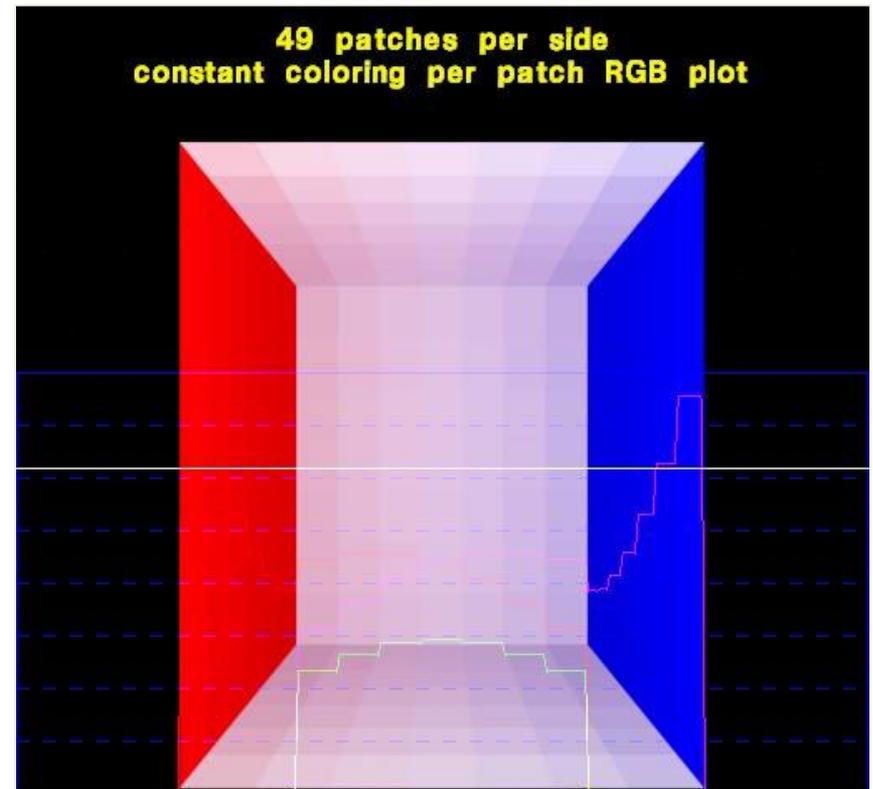
---



# Результаты работы метода излучательности

- Cornell box

Cindy M. Goral, Kenneth E. Torrance, and Donald P. Greenberg for the 1984 paper *Modeling the interaction of Light Between Diffuse Surfaces*, *Computer Graphics (SIGGRAPH '84 Proceedings)*, Vol. 18, No. 3, July 1984, pp. 213-222.

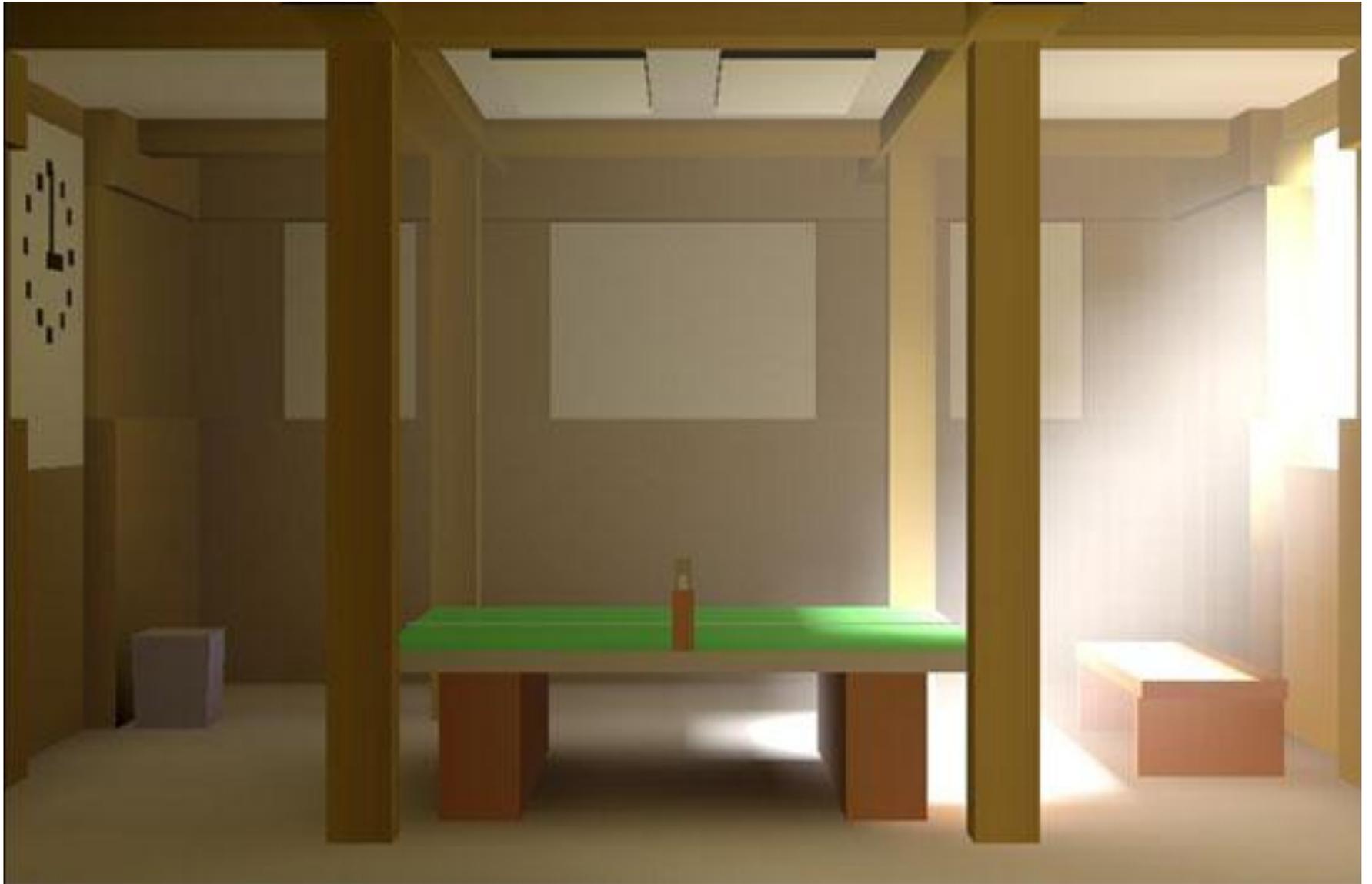


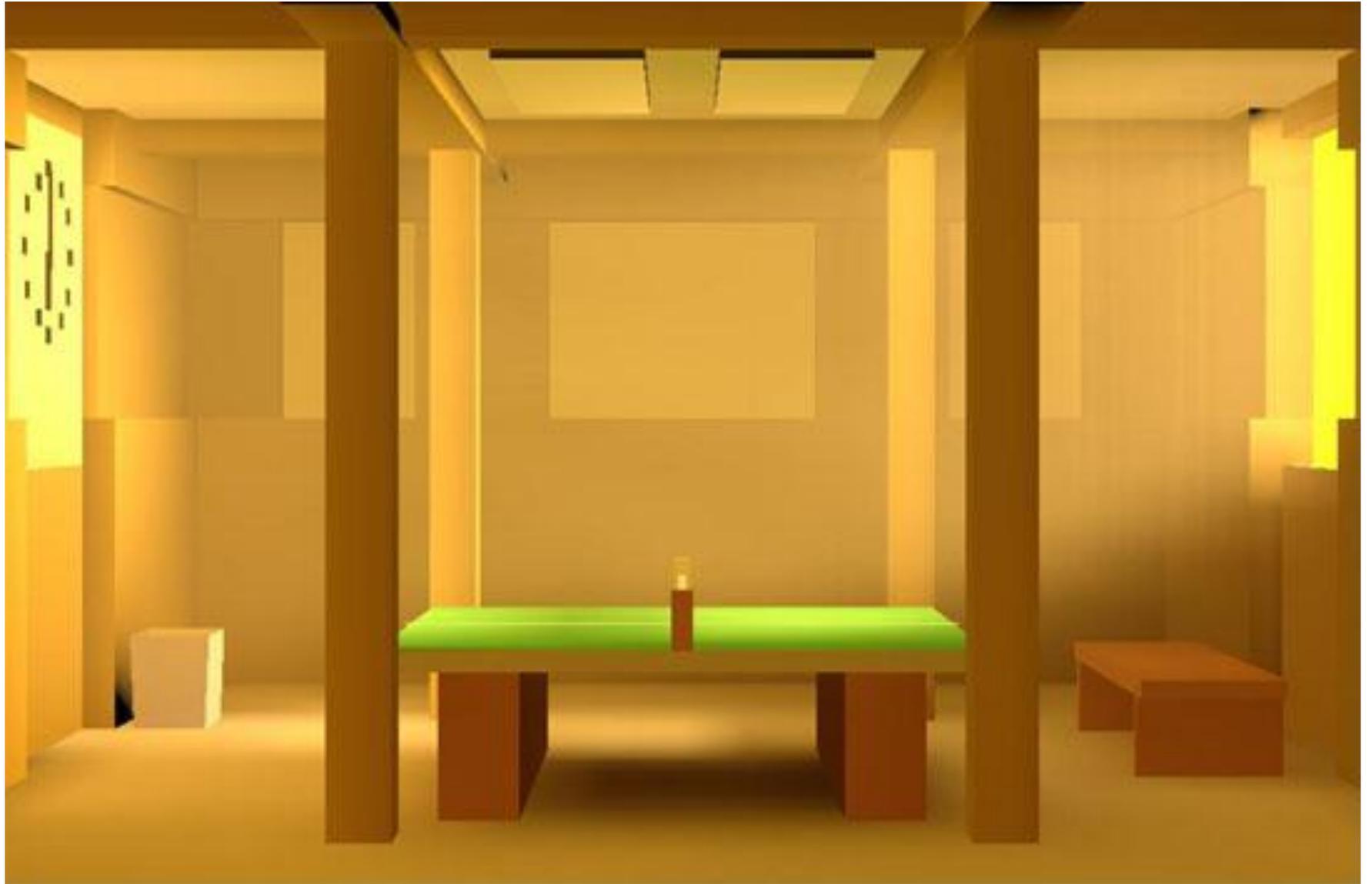
# The Cornell Box

- Michael F. Cohen and Donald P. Greenberg for the 1985 paper **The Hemi-Cube, A Radiosity Solution for Complex Environments**, Vol. 19, No. 3, July 1985, pp. 31-40.
- The hemi-cube















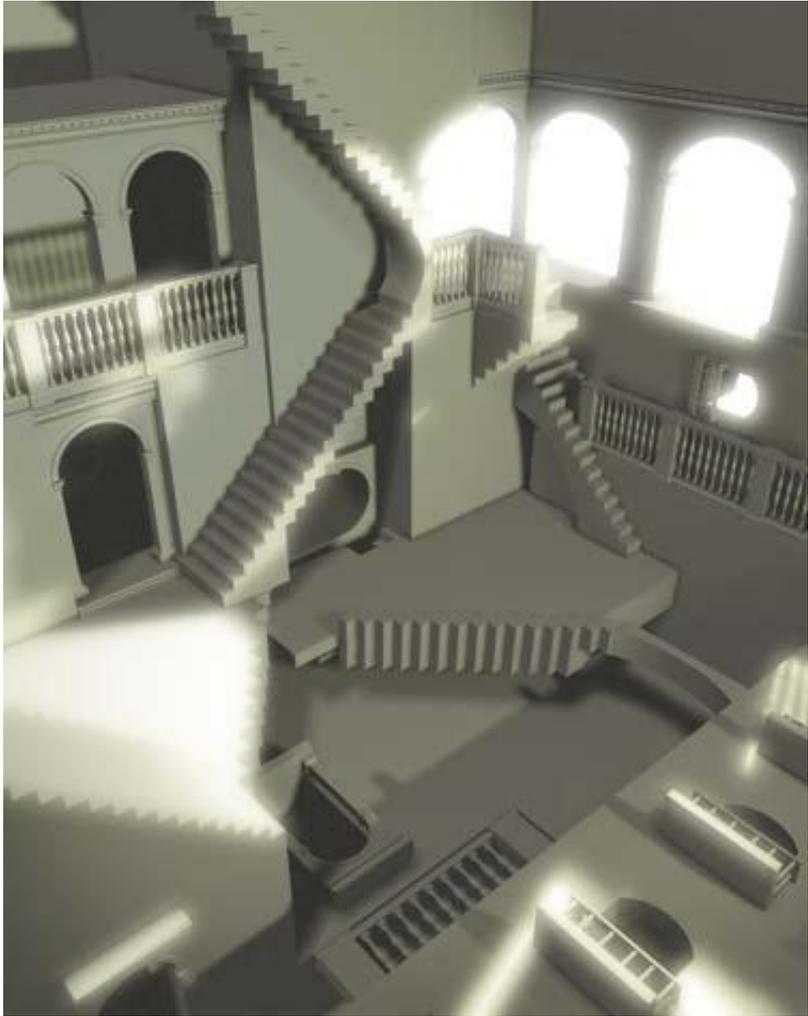
















# Итоги

---

- Визуализация с помощью растеризации – очень быстро, но сложно реализовать некоторые эффекты: вторичное освещение, тени, отражения и преломления
- Трассировка лучей: «бросаем» луч через каждый пиксель экрана, находим пересечения с ближайшей точкой поверхностей сцены, затем для расчета освещения создаем новые лучи
- Излучательность: рассчитываем перенос энергии между поверхностями сцены, начиная с источников света