

Нижегородский государственный университет им. Н.И.Лобачевского

Факультет вычислительной математики и кибернетики



LOBACHEVSKY STATE UNIVERSITY
of NIZHNI NOVGOROD
National Research University

Computing Mathematics and Cybernetics faculty

Software department

CS255. Computer Graphics Introduction Course

Элементы глобального освещения сцены. Классическая трассировка лучей

*Турлапов В.Е.
Боголепов Д.К.*

Global & Local Illumination

Алгоритмы моделирования освещения условно можно разделить на две группы: алгоритмы локального освещения (**local illumination**) и алгоритмы глобального освещения (**global illumination**).

Local illumination

Алгоритмы локального освещения - самые простые способы расчета освещенности поверхностей, рассматривают отражение или пропускание света для каждой поверхности 3D сцены независимо от других поверхностей, принимая во внимание только первичные источники света (задача Lighting/Shading: интенсивность, цвет, направление распространения отражённого и преломленного света).

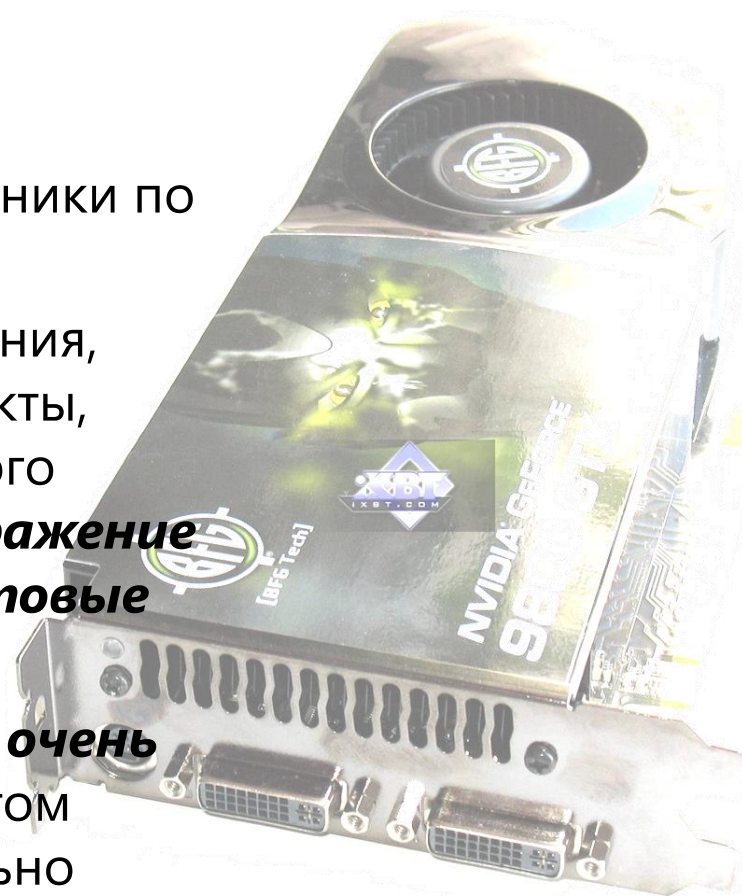
Алгоритмы *local illumination* могут использоваться в качестве элементарных составляющих расчета глобального освещения.

Алгоритмы *local illumination* используются в играх для расчета динамического освещения моделей.

Используются при аппаратной растеризации.

Метод растеризации

- ▶ Современные видеокарты формируют изображение с помощью **методов растеризации**, позволяющих быстро отрисовывать текстурированные треугольники по данным о вершинах
- ▶ Применяется упрощенная модель освещения, которая **не позволяет** реализовать эффекты, необходимые для получения реалистичного изображения сцены: **преломление** и **отражение** лучей света, **вторичное освещение**, **световые эффекты – каустики**
- ▶ **Правдоподобные эффекты освещения очень важны**: именно игра света и тени во многом определяет то, насколько живо и натурально выглядит трехмерная виртуальная реальность

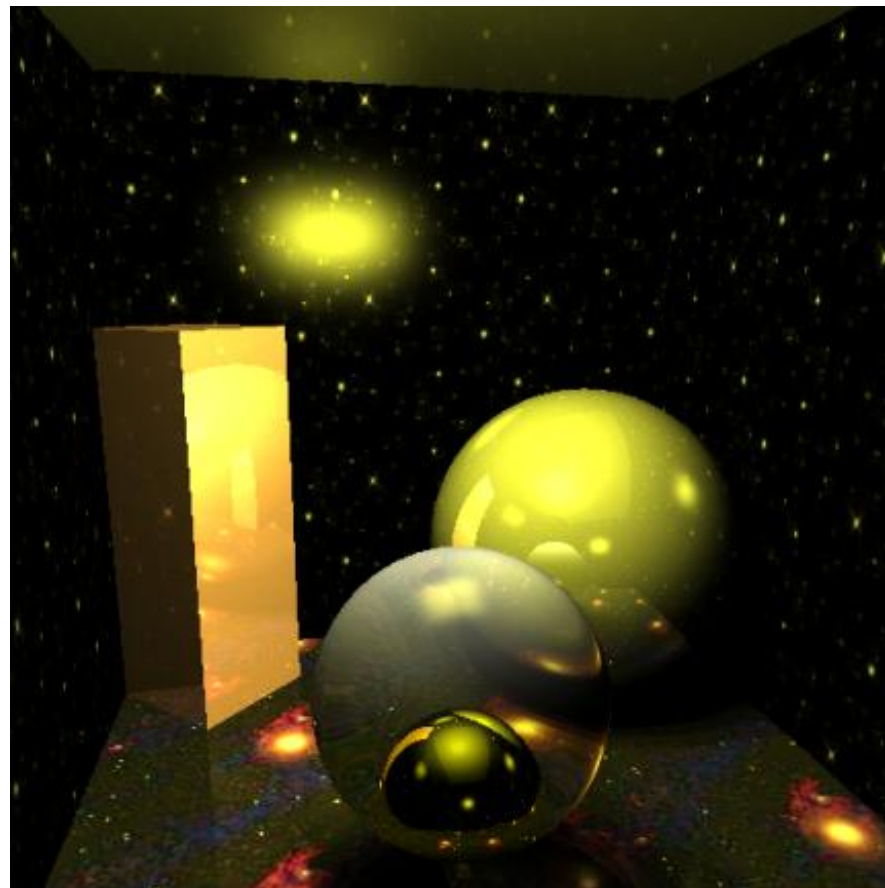
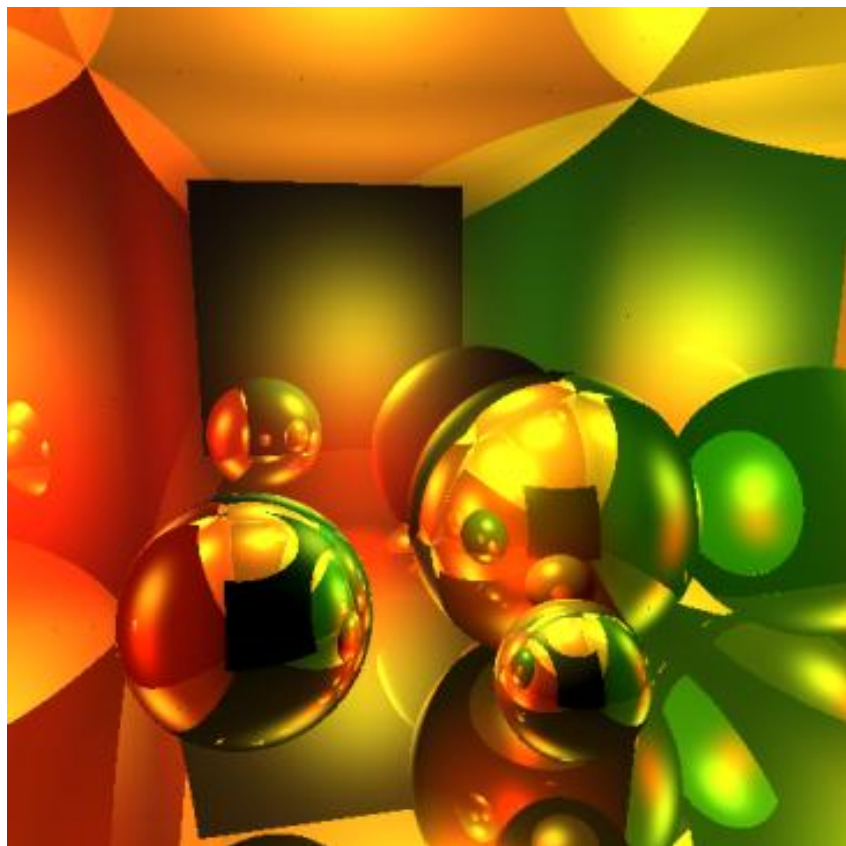


Метод трассировки лучей

- ▶ Главной целью алгоритмов **global illumination** является воспроизведение **реального освещения**. Существуют методы, позволяющие довольно реалистично моделировать освещение сцены. Тестирование на Корнуэльской комнате.
- ▶ Важнейшее место среди таких подходов занимает **трассировка лучей** (*ray tracing* или *рейтрейсинг*): исторически - первый из подходов
- ▶ **Трассировка лучей моделирует оптические законы прохождения лучей света через изображаемую сцену.** Фотореализм достигается путем математического моделирования оптических свойств взаимодействия света с объектами

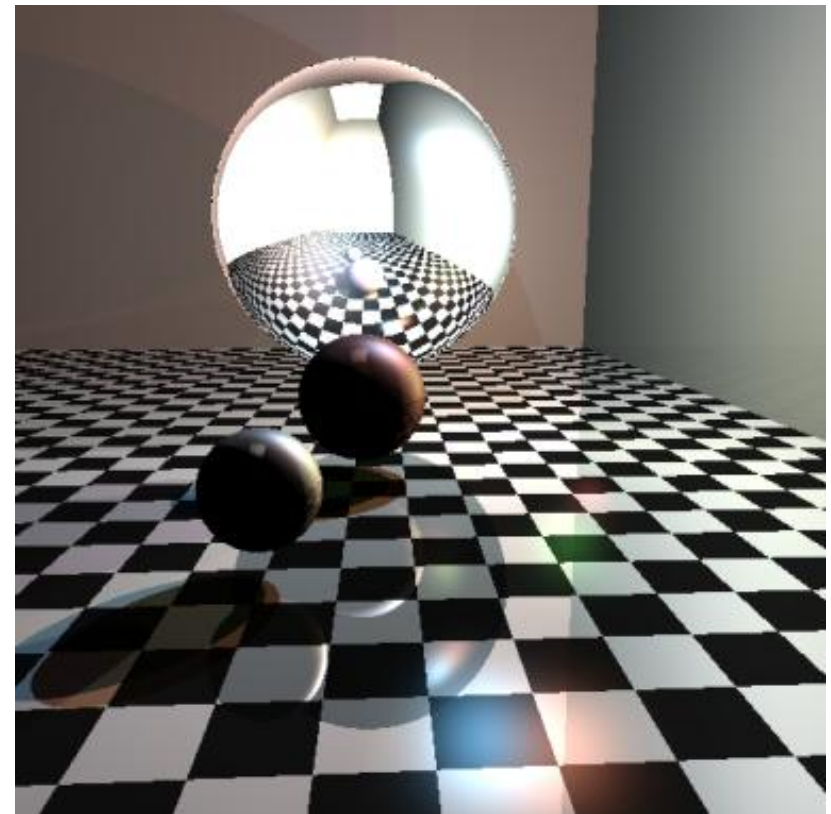
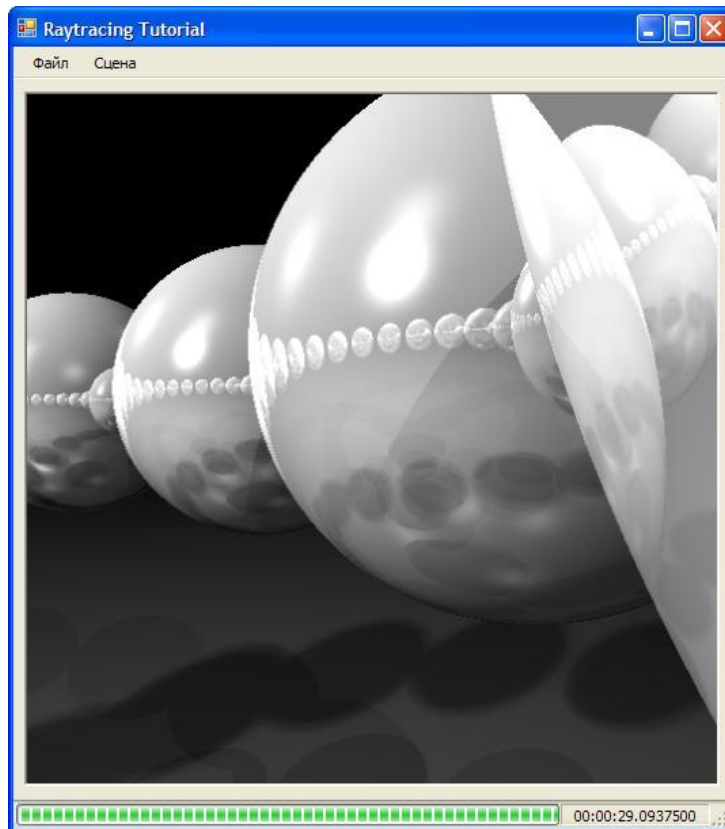
Проблема реального времени и подходы к упрощению расчета освещенности

Результаты трассировки лучей на процессоре Core 2 Duo, полученные стажерами Зимней школы 2008 по компьютерной графике (Intel-ННГУ). 512x512pix. Время – порядка 1 мин.



Проблема реального времени и подходы к упрощению расчета освещенности

Результаты трассировки лучей на процессоре Core 2 Duo, полученные стажерами Зимней школы 2008 по компьютерной графике (Intel-ННГУ). 512x512pix. Время – порядка 1 мин.



Примеры синтезированных изображений. Как?



"Glasses" by Gilles Tran (2006)

Примеры синтезированных изображений. Как?



"Evening at the River" by Christoph Gerber (2002 and 2004)

Трассировка лучей (Ray tracing)

Прямая и обратная трассировка

Классический ray tracing, или метод трассировки лучей, предложен Артуром Аппелем (Arthur Appel) в 1968 году и дополнен алгоритмом общей рекурсии, разработанным Whitted в 1980 году. Понадобилось почти 12 лет эволюции вычислительных систем, прежде чем этот алгоритм стал доступен для широкого применения в практических приложениях.

Суть метода: отслеживание траекторий лучей и расчета взаимодействий с лежащими на траекториях объектами, от момента испускания лучей источником света до момента попадания в камеру.

Под взаимодействием луча с объектами понимаются процессы диффузного (в смысле модели локальной освещенности), многократного зеркального отражения от их поверхности и прохождение лучей сквозь прозрачные объекты.

Ray tracing – первый метод расчета глобального освещения, рассматривающий освещение, затенение (расчет тени), многократные отражения и преломления.

Различают два подхода к трассировке лучей: **метод прямой трассировки** – forward ray tracing, и **метод обратной трассировки** – backward ray tracing.

Path tracing и Ray tracing по Thomas Funkhouser (Princeton University, COS 526, Fall 2002).

Ray tracing не использует модели шейдинга (растеризации). Расчет освещенности "честно" выполняется во всех точках пересечения лучей и объектов.

Трассировка лучей (Ray tracing)

Обратная трассировка. Ограничения метода

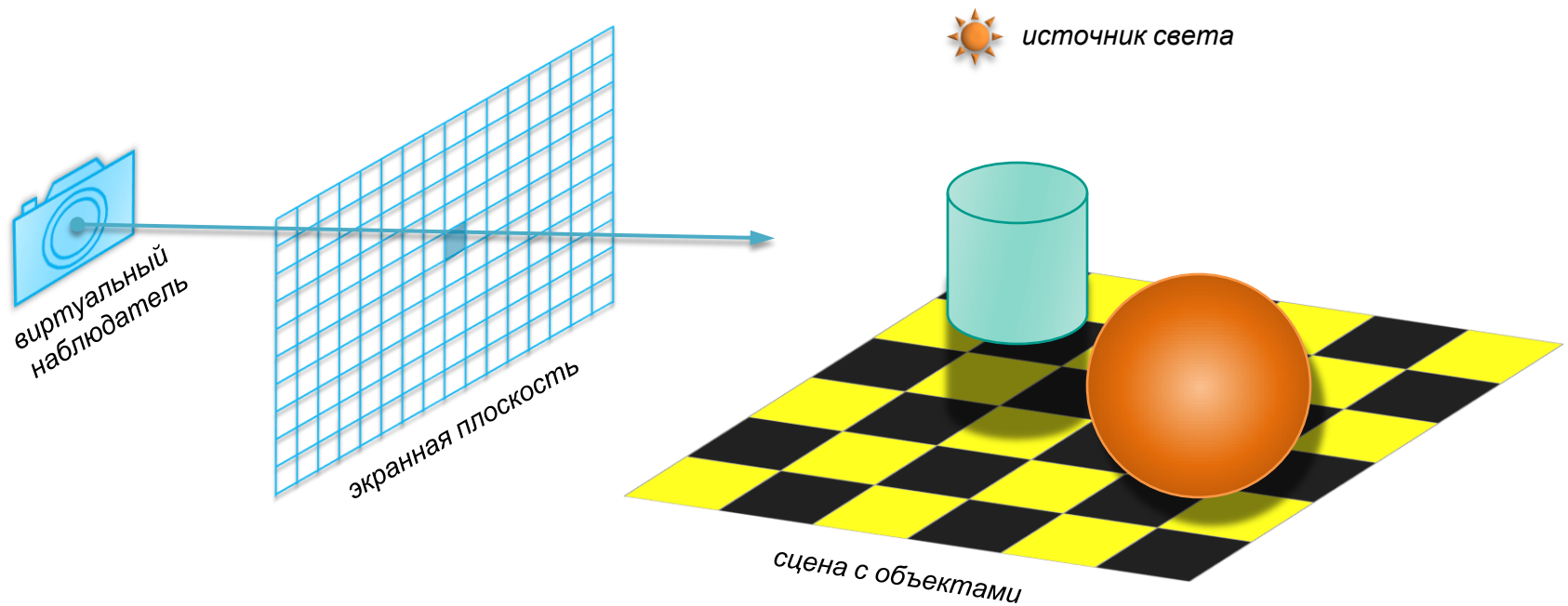
Ограничения метода ради повышения производительности

1. Выделяют источники света. Они могут только излучать, но не отражать и преломлять. Ограничивают многообразие источников точечными.
2. Свойства отражающих поверхностей задаются суммой диффузной и зеркальной компонент
3. Зеркальность описывается составляющими reflection и specular. Reflection учитывает отражение других объектов не являющихся источниками. Строится только 1 зеркально отраженный луч r для дальнейшей трассировки. Specular дает световые блики от источников. Для этого определяются углы α отраженного луча обратной трассировки со всеми источниками.
4. При диффузном отражении учитываются только лучи от источников света (от зеркал игнорируются)
5. Для прозрачных (transparent) объектов обычно не учитываются зависимость преломления от λ . Иногда прозрачность моделируется без преломления.
6. Для учета освещенности объектов светом, рассеиваемым другими объектами вводится фоновая составляющая (ambient)
7. Завершают трассировку либо по числу итераций, либо по величине приращения освещенности, оказавшейся на данной итерации ниже пороговой



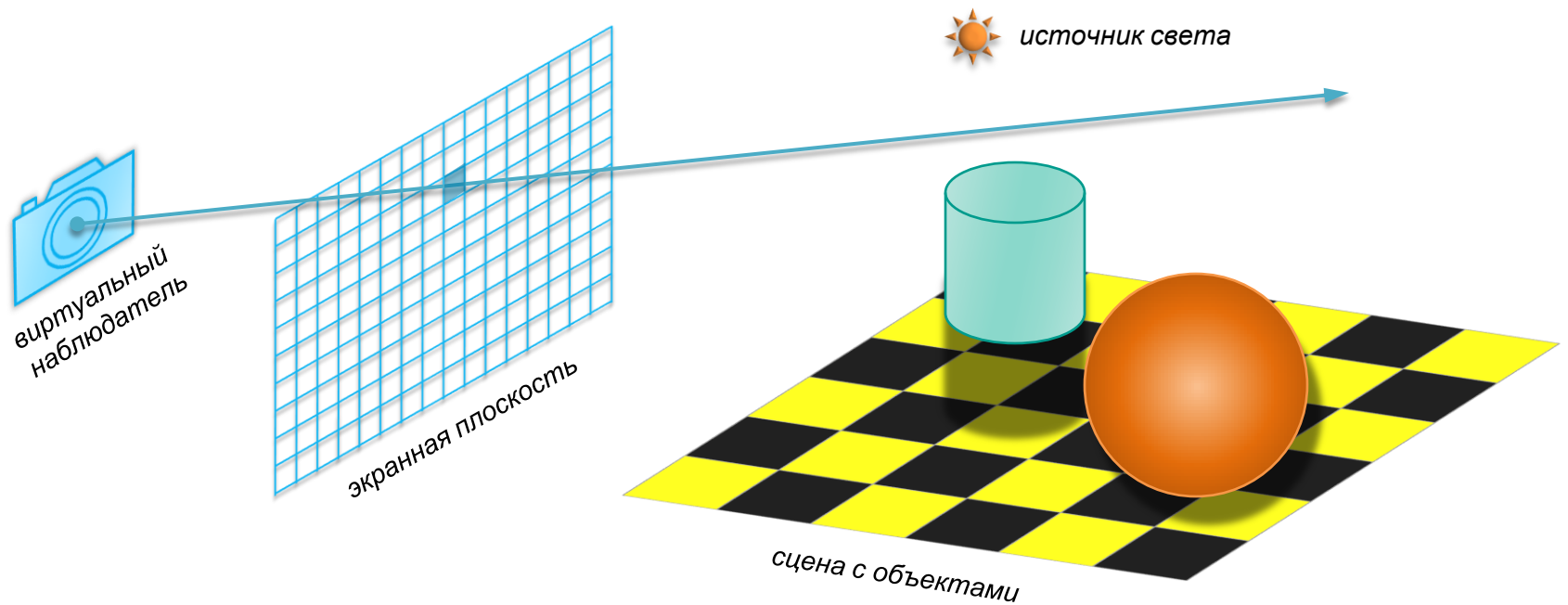
Генерация первичного луча...

- ▶ В **традиционной** трассировке лучей лучи света обрабатываются в **обратном** направлении (*backward ray tracing*): луч испускается из камеры вглубь сцены через пиксель окна вывода



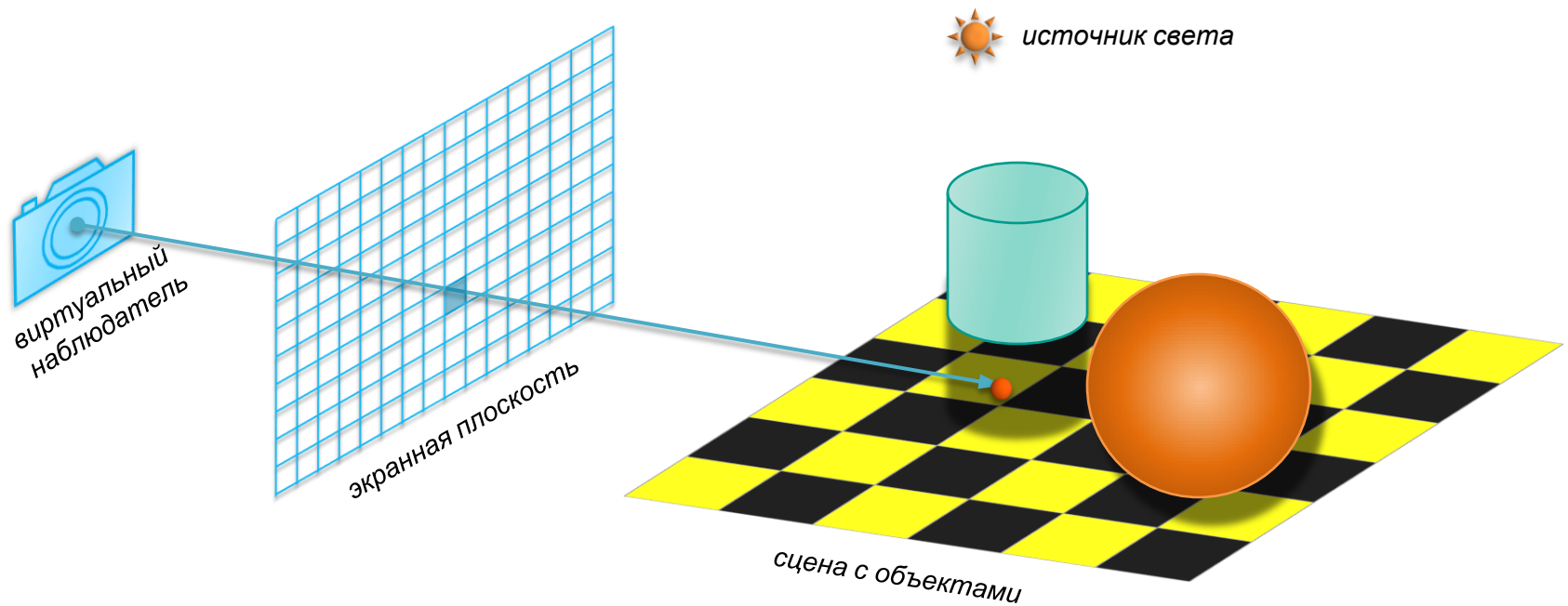
Луч проходит мимо сцены

- ▶ Первичный луч может **не столкнуться** ни с одним объектом сцены. В этом случае процесс обрывается – пиксель закрашивается цветом фона



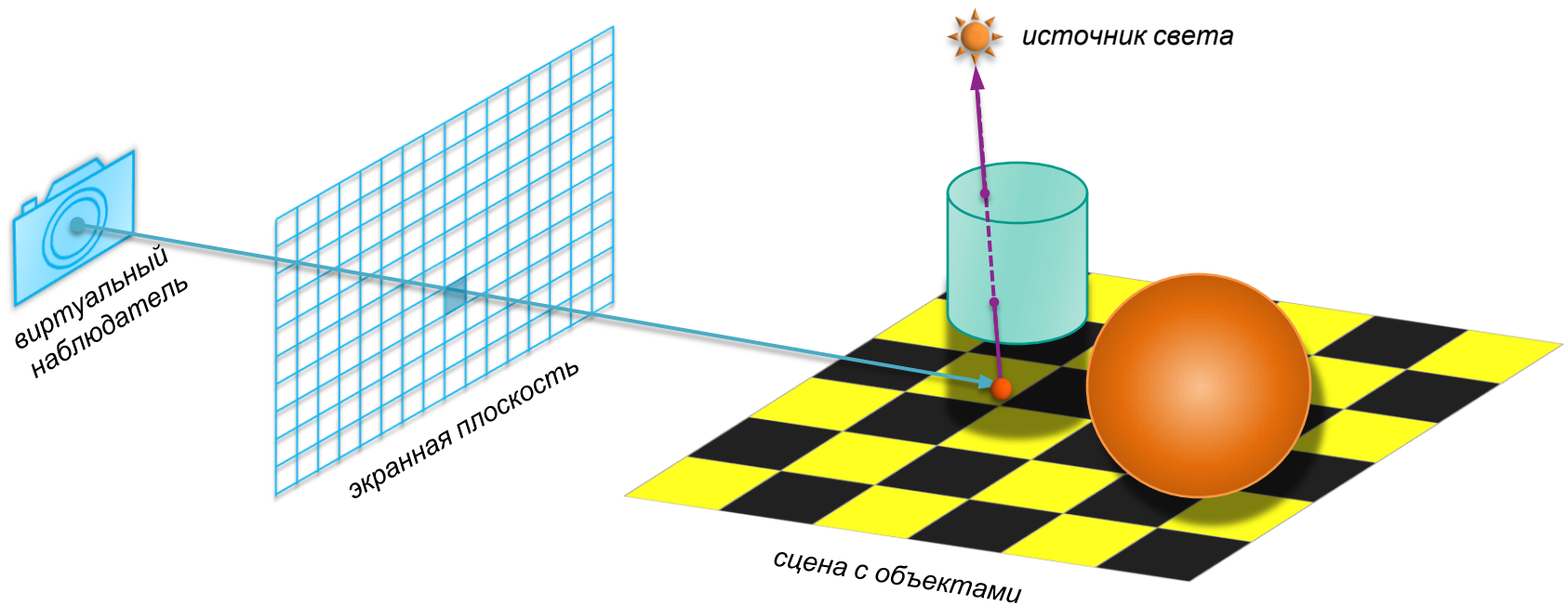
Луч соударяется с объектом сцены

- ▶ В остальных случаях луч соударяется с некоторым объектом. Необходимо рассчитать **прямое освещение** и **вторичное освещение**, а также выяснить, **находится ли точка в тени**



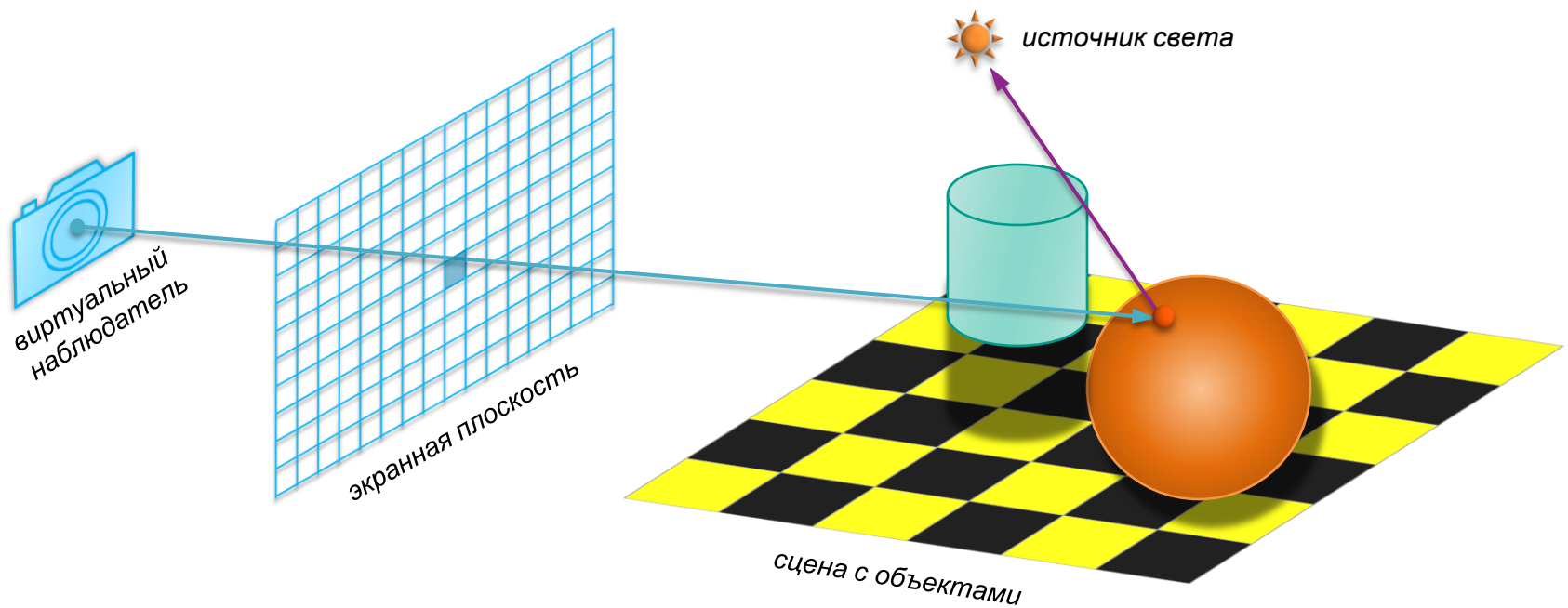
Находится ли точка в тени?

- ▶ Для выяснения того, находится ли точка в тени, следует выпустить **вторичный теневой луч** из точки к источнику света. Если теневой луч столкнется с объектом сцены, то точка находится в тени



Расчет прямого освещения...

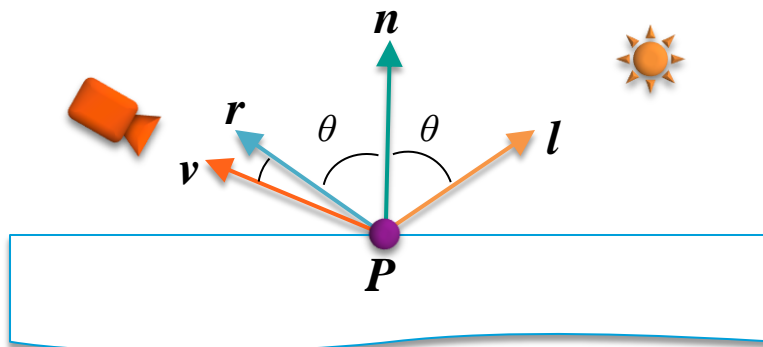
- ▶ Если лучи от источника света достигают точки, то следует рассчитать вклад от **прямого освещения** данным источником (используются различные **модели освещения** – эмпирические или физически корректные)



Расчет прямого освещения

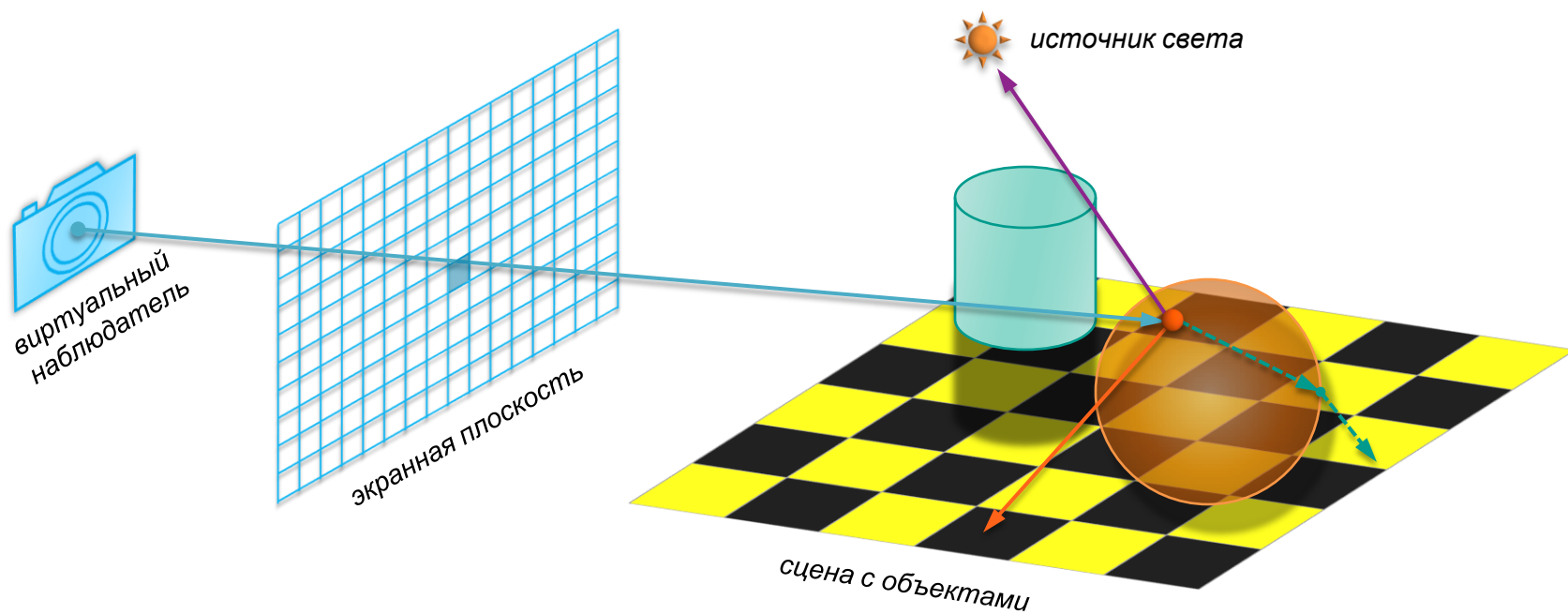
- Наиболее распространенная **эмпирическая модель** – **освещение по Фонгу**

$$C_{out} = C \cdot [k_a + k_d \cdot \max(0, (\mathbf{n} \cdot \mathbf{l}))] + l \cdot k_s \cdot \max(0, (\mathbf{v} \cdot \mathbf{r}))^p,$$
$$\mathbf{r} = \text{reflect}(-\mathbf{v}, \mathbf{n})$$



Расчет вторичного освещения

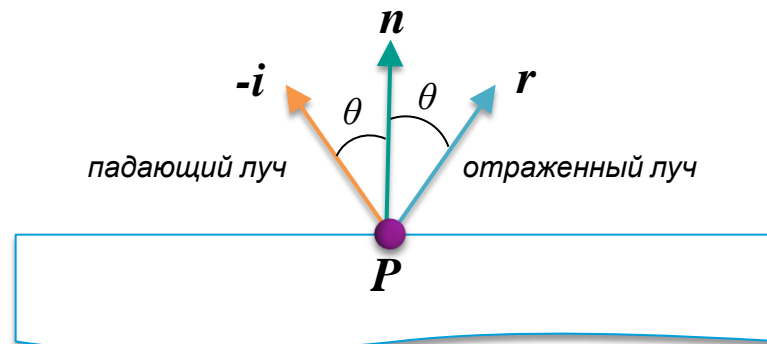
- ▶ Однако точка может получать энергию и от **вторичного освещения** посредством **отраженных** и **преломленных** лучей



Учет отражения света

- ▶ Если поверхность обладает отражающими свойствами, то строится вторичный **луч отражения**. Направление луча определяется по **закону отражения** (геометрическая оптика):

$$\mathbf{r} = \mathbf{i} - 2 \cdot \mathbf{n} \cdot (\mathbf{n} \cdot \mathbf{i})$$

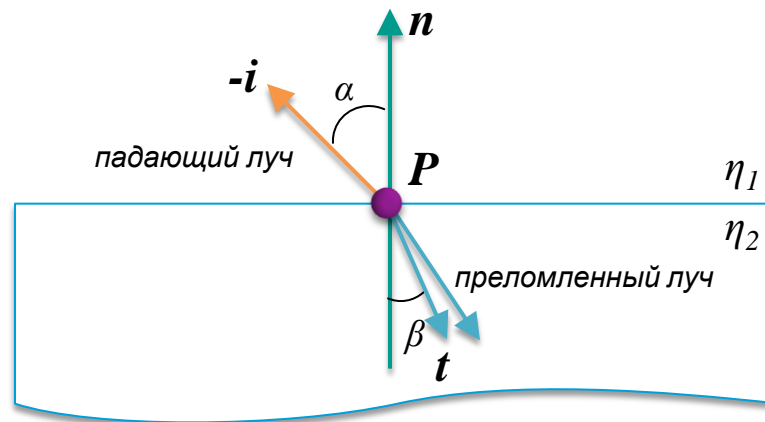


- ▶ Для отраженного луча также определяется возможность его взаимодействия с объектами сцены. Если соударений нет, то возвращается цвет фона

Учет преломления света

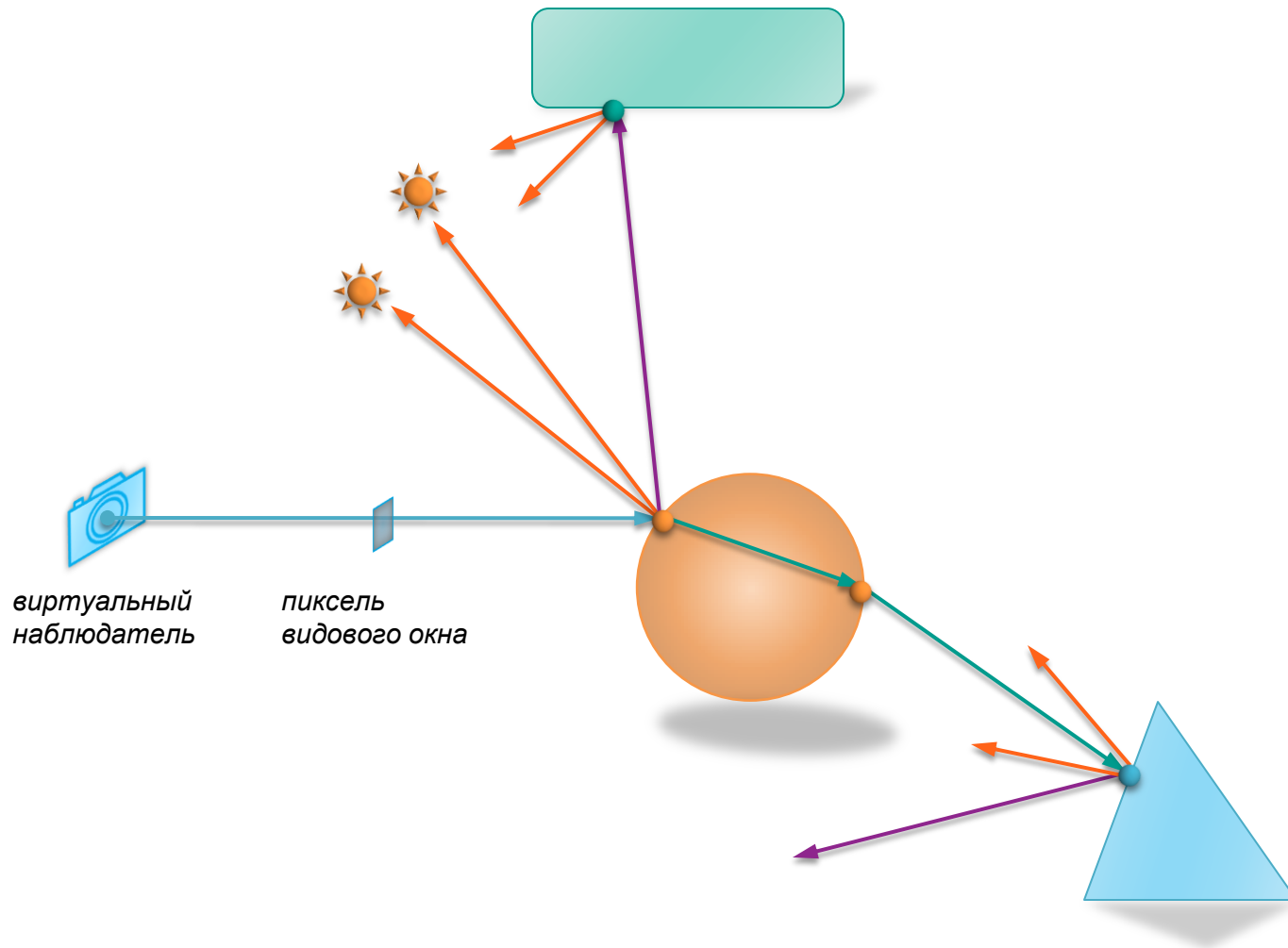
- ▶ Если же поверхность прозрачна, то строится еще и вторичный **луч прозрачности** (*transparency ray*). Для определения направления луча используется **закон преломления** (геометрическая оптика):

$$\sin(\alpha) / \sin(\beta) = \eta_2 / \eta_1$$



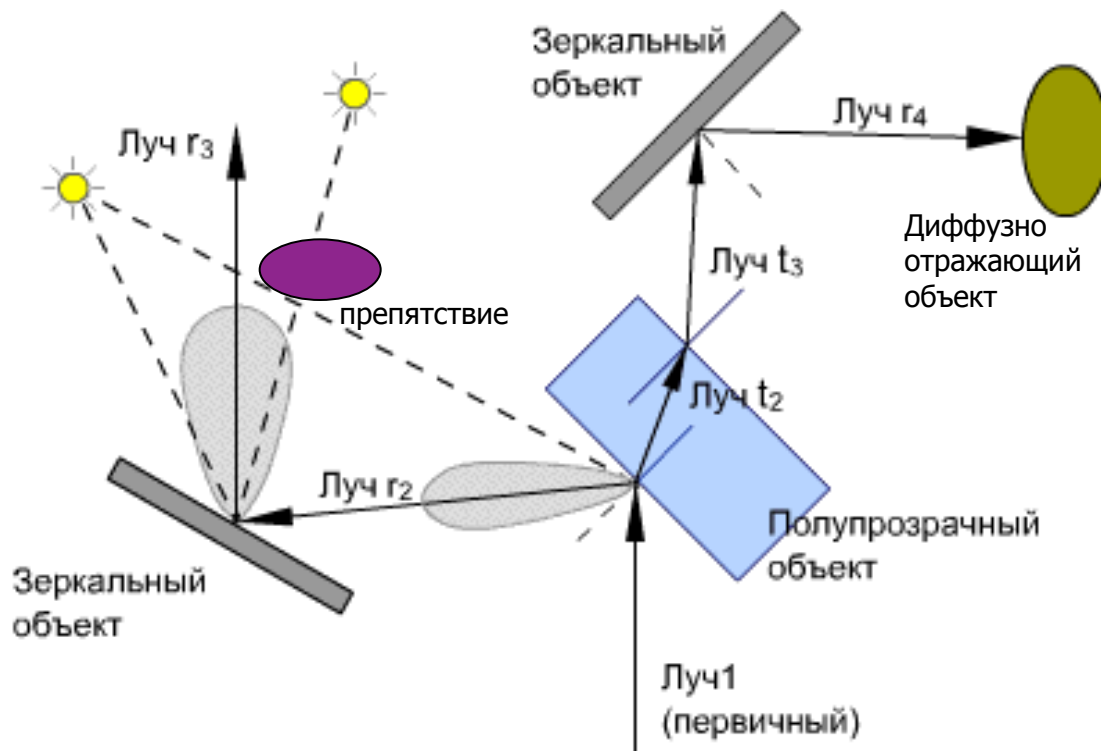
$$\mathbf{t} = (\eta_1 / \eta_2) \cdot \mathbf{i} - [\cos(\beta) + (\eta_1 / \eta_2) \cdot (\mathbf{n} \cdot \mathbf{i})] \cdot \mathbf{n},$$
$$\cos(\beta) = \text{sqrt}[1 - (\eta_1 / \eta_2)^2 \cdot (1 - (\mathbf{n} \cdot \mathbf{i})^2)]$$

Процесс трассировки...



Рекурсивная процедура трассировки.

Обратная трассировка. Метод разработан в 80-х годах. Рекурсивная процедура

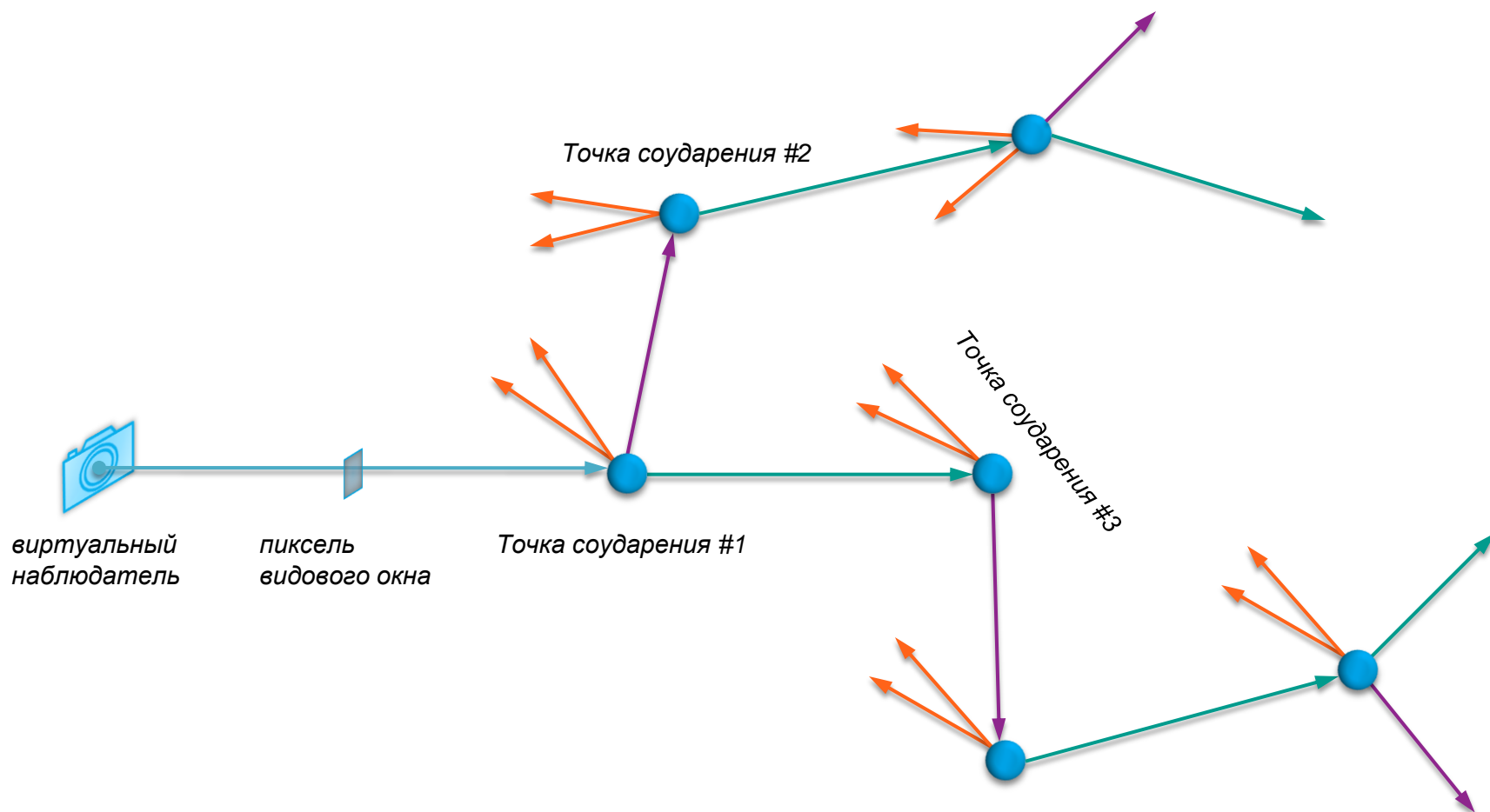


Пояснение :
Луч t – преломленный;
Луч r – отраженный;

Процесс трассировки...

- ▶ В каждой точке пересечения *теневые лучи* строятся всегда. Если источник света видим, то по *локальной модели освещения* находится освещенность в точке пересечения
- ▶ *Преломленный* и *отраженный* лучи строятся при условии, что поверхность обладает свойствами отражения или преломления. Если объект такими свойствами не обладает, в то луч обрывается
- ▶ В идеале, процесс продолжается до тех пор, пока все лучи не будут рассеяны на чисто диффузных поверхностях или не выйдут за пределы видимой области
- ▶ В результате получаем дерево трассировки

Процесс трассировки



Процесс трассировки

- ▶ Если древовидная структура построена, то можно выполнить расчет освещенности точки
- ▶ Освещенность в общем случае вычисляется по формуле:

$$I_{total} = I_{local} + k_{reflection} \cdot I_{reflection} + k_{refraction} \cdot I_{refraction}$$

- ▶ Данная формула используется для расчета освещенности очередного узла на пути от листьев дерева (чисто диффузные поверхности) к его корню
- ▶ Таким образом, по мере приближения к первой точке соударения, мы накапливаем освещенность, полученную в результате взаимодействия с другими объектами сцены – “вторичную” освещенность

Рекурсивная процедура трассировки.

Модель Уиттеда (Whitted) для цвета точки:

$$I(\lambda) = k_a L_a(\lambda) \cdot C(\lambda) + k_d L_d(\lambda) \cdot C(\lambda) + k_s L_s(\lambda) + k_r L_r(\lambda) + k_t L_t(\lambda), \quad (*)$$

где λ - длина волны света; $C(\lambda)$ – исходный цвет материала в точке

k_a, k_d, k_s, k_r, k_t - коэффициенты, учитывающие свойства материала в отношении фоновой подсветки, диффузного рассеивания, зеркальности, отражения и прозрачности; L – соответствующие интенсивности. При решении задачи в цвете формула (*) применяется для каждой цветовой составляющей (RGB).

Для каждой точки окна осуществляется внешнее обращение к базовой рекурсивной процедуре:

$I = \text{ЛУЧ}(1, \text{первичный, направление}, 0)$; // см. следующий слайд

Оптимизация: В данной процедуре следует при каждом вызове проверять (вычислять) наличие пересечения с гранями объектов → перебор всех граней. Для ускорения процесса применяется метод оболочек (для отбрасывания заведомо неприемлемых кандидатов). Оболочки могут образовывать древовидную структуру. Это позволяет существенно ускорить перебор и сделать его теоретически пропорциональным логарифму от числа граней (аналогия с количеством информации).

Рекурсивная процедура трассировки.

Базовая операция обратной трассировки – вычисление интенсивности для трассируемого луча:

```
ЛУЧ(номер итерации ind, тип луча, направление луча dir, номер объекта no) {  
Находим точку пересечения луча с ближайшим объектом (гранью); Если точка найдена, то  
{ no=номер пересекаемого объекта; // использовать метод оболочек  
Вычисляем нормаль к видимой стороне пересекаемой грани;  
Если ( $k_d > 0$ ), то // задано свойство диффузного отражения  
  { $L_d$ =Сумма интенсивности диффузного отражения для всех источников;}  
Если ( $k_s > 0$ ), то // зеркальные блики от источников света  
  {Определяем направление отраженного луча dirR;  
   $L_s$ =Интенсивность зеркального блика с учетом  $\alpha$  для всех источников;}  
Если ( $k_r > 0$ ), то // зеркальное отражение других объектов  
  {Определяем направление отраженного луча dirR;  
   $L_r$ =ЛУЧ(ind+1, отраженный, dirR, no);} // рекурсия  
Если ( $k_t > 0$ ), то // объект полупрозрачный  
  {Определяем направление преломленного луча dirT;  
   $L_t$ =ЛУЧ(ind+1, преломленный, dirT, no);} // рекурсия  
return  $k_a \cdot L_a \cdot C + k_d \cdot L_d \cdot C + k_s \cdot L_s + k_r \cdot L_r + k_t \cdot L_t$ ; //  $k_r L$  и  $C$  - вектора (для RGB)  
}  
иначе {Луч уходит в свободное пространство; return Значение по умолчанию (цвет фона)}}}
```

Итоги

- ▶ Трассировка лучей – первый метод расчета *глобального освещения*, который позволил визуализировать *тени, многократные отражения и преломления* → значительно повысилась реалистичность изображений.
- ▶ Высокая степень параллельности вычислений
- ▶ Недостатки:
 - ▶ Упрощения (ограничения) метода, неучет вторичного освещения от диффузно отраженного объектами, неучет прямой трассировки света (каустик, радуги);
 - ▶ высокая вычислительная стоимость расчетов;
 - ▶ резкие границы цветовых переходов и “зазубренность” линий (часто необходим AA);

Источники...

- ▶ *Overview of Ray Tracing*
<http://www.siggraph.org/education/materials/HyperGraph/raytrace/rtrace1.htm>
- ▶ A.Reshetov, A.Soupikov, J.Hurley (Intel) [Multi-Level Ray Tracing Algorithm](#)
- ▶ *Механика виртуальности: Ray Tracing*
<http://www.computerra.ru/print/hitech/36685>
- ▶ *Трассировка лучей с помощью Pov-Ray*
<http://www.osp.ru/os/1997/06/179311>
- ▶ *Как компьютер рассчитывает изображения*
<http://www.fcenter.ru/online.shtml?articles/hardware/videos/8749>
- ▶ Ф.Хилл. *OpenGL: Программирование компьютерной графики.* – СПб.: Питер, 2002.