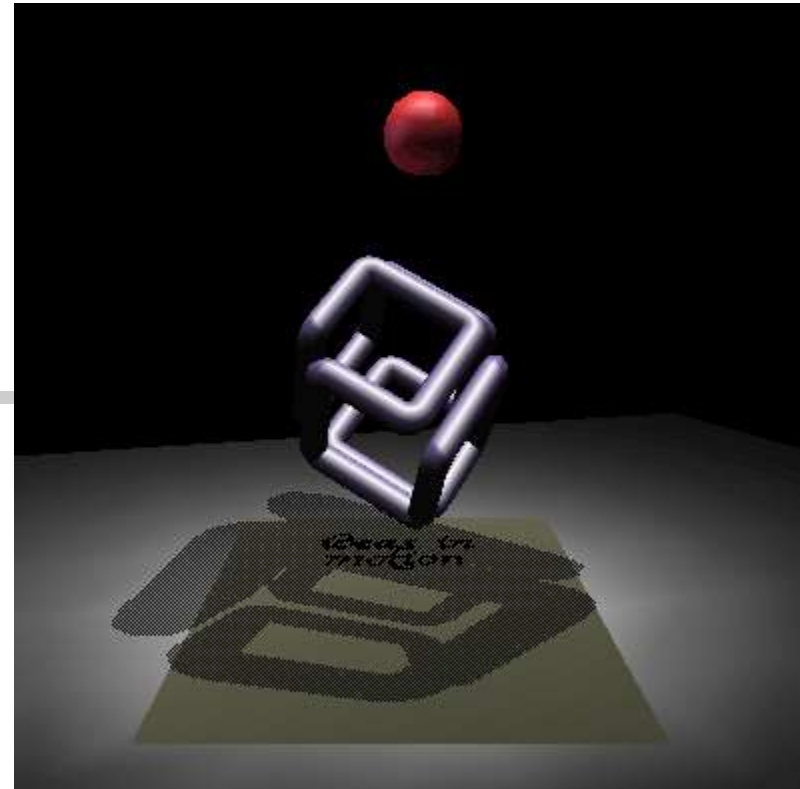


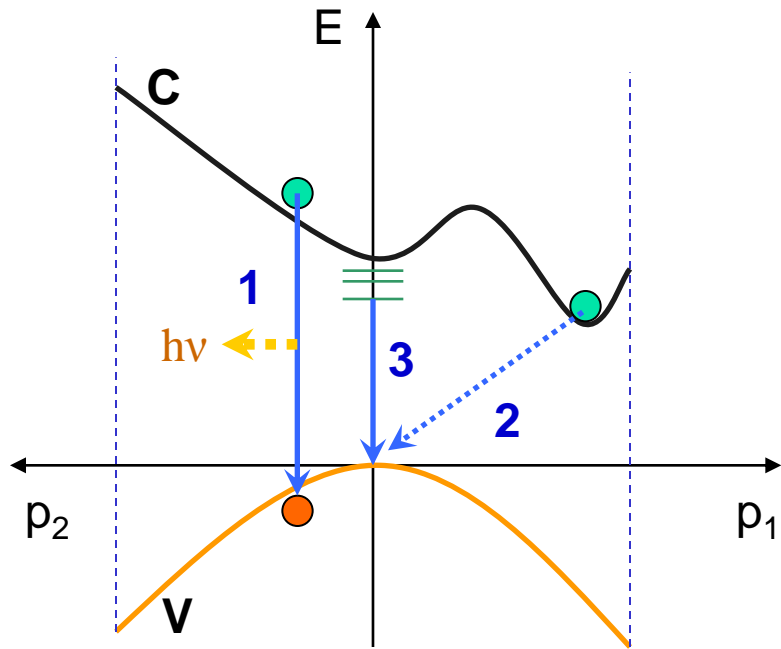
Компьютерная графика

лекция 4

1. Свет и цвет в трёхмерных моделях



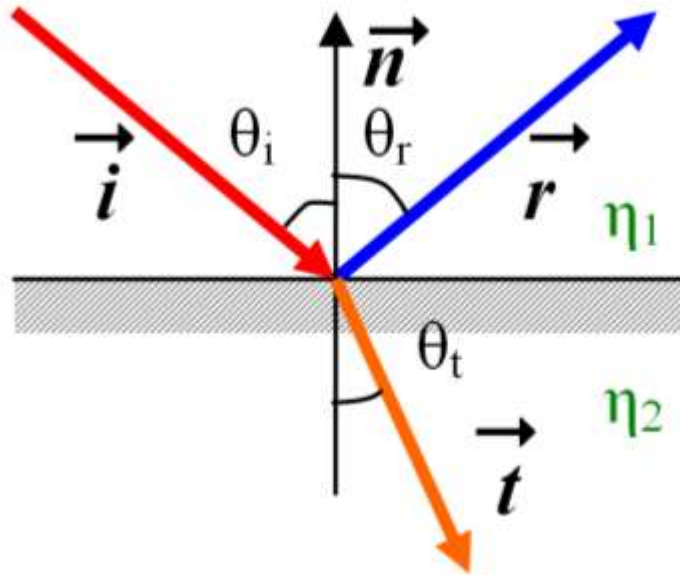
Оптические свойства объектов трехмерных сцен



В настоящее время основное отличие «реалистичных» моделей компьютерной графики от реальной физики взаимодействия света с веществом заключается в **игнорировании** зависимости коэффициентов отражения и преломления света от длины волны падающего излучения, **неучёта** явлений поляризации и **пренебрежение** механизмами рассеяния света веществом.

Оптические свойства объектов трехмерных сцен:

идеальное зеркальное отражение и преломление



$$\vec{r} = \vec{i} + 2(\vec{i} \cdot \vec{n}) \cdot \vec{n}$$

$$\vec{t} = \eta \vec{i} + \left(\eta C - \sqrt{1 + \eta^2 \cdot (C^2 - 1)} \right) \cdot \vec{n},$$

$$\eta = \frac{\eta_1}{\eta_2}, C = (\vec{i} \cdot \vec{n}) = \cos(\theta_i)$$

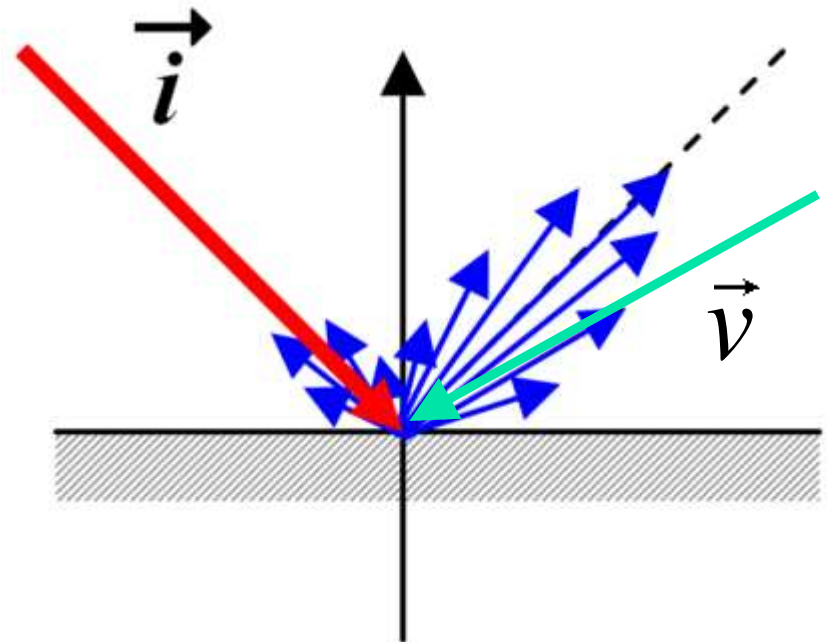
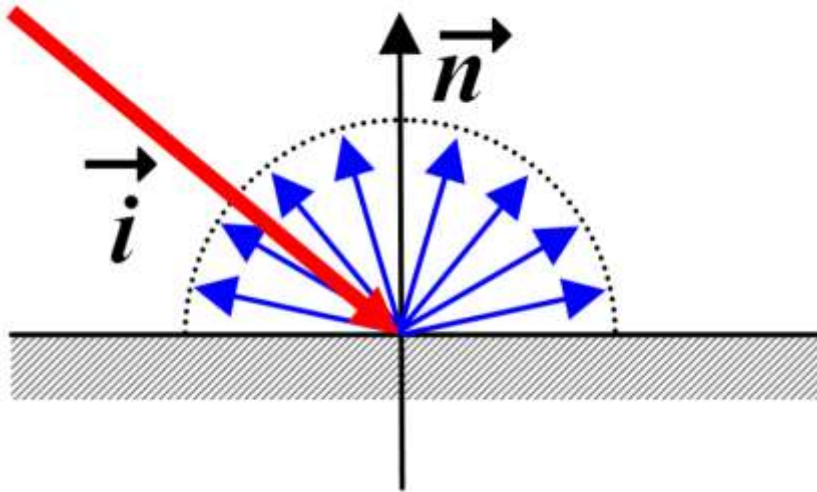
Оптические свойства объектов трехмерных сцен:
диффузно отражающие поверхности

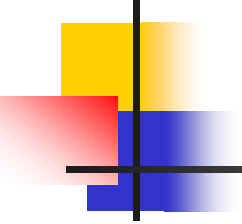


идеальное диффузное отражение

реалистичное отражение

$$I = I_0 (\vec{i} \cdot \vec{n})$$





Оптические свойства объектов трехмерных сцен:
диффузное отражение и преломление

$$R(\theta) = \frac{1}{2} \left\{ \left(\frac{\cos \theta_i - \eta \cdot \cos \theta_t}{\cos \theta_i + \eta \cdot \cos \theta_t} \right)^2 + \left(\frac{\eta \cdot \cos \theta_i - \cos \theta_t}{\eta \cdot \cos \theta_i + \cos \theta_t} \right)^2 \right\}$$

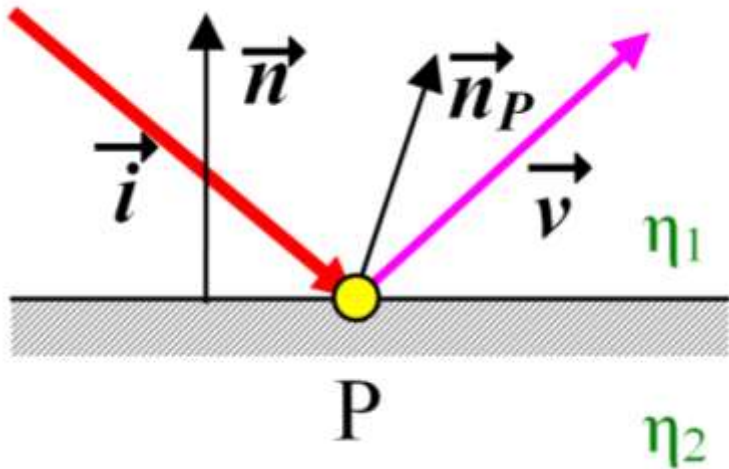
$$T(\theta) = \frac{\eta \cdot \cos \theta_t}{2 \cdot \cos \theta_i} \left\{ \left(\frac{2 \cdot \cos \theta_i}{\cos \theta_i + \eta \cdot \cos \theta_t} \right)^2 + \left(\frac{2 \cdot \cos \theta_i}{\eta \cdot \cos \theta_i + \cos \theta_t} \right)^2 \right\}$$

~~$\eta(\lambda)$~~

$$I = I_0 \cdot \frac{R(\theta_i)}{(\vec{n} \cdot \vec{i}) \cdot (\vec{n} \cdot \vec{v})}$$

$R = K \cdot (\vec{v} \cdot \vec{i})^p$ - модель Уиттеда

Оптические свойства объектов трехмерных сцен:
шероховатая поверхность



$$D(\gamma) = C e^{-\left(\frac{\gamma}{m}\right)^2}$$

$$I = I_0 \cdot \frac{R(\lambda, \theta_i) \cdot D(\gamma)}{(\vec{n} \cdot \vec{i}) \cdot (\vec{n} \cdot \vec{v})}$$

Оптические свойства объектов трехмерных сцен:

поглощение света

$$I = I_0 \cdot e^{-\alpha \cdot x} \quad \text{--- } \alpha(\lambda)$$

перспективное текстурирование

- наложение bitmap-изображений на поверхность (грань) с целью изменения параметров отображения микрограней (цвета, прозрачности, положения вектора нормали, самосвечения и др.)

- point-sampling
- bi-linear filtering
- mip-mapping
- tri-linear filtering



Освещение в трехмерных сценах:

ИСТОЧНИКИ СВЕТА

1. Рассеянный свет:

$$I = I_0 + I_a k_a$$

2. Точечный источник:

$$I = I_0 + I_l \frac{\cos \theta_i}{d^2}$$

3. Реалистичный
точечный источник:

$$I = I_0 + I_l \frac{\cos \theta_i}{d + k}$$

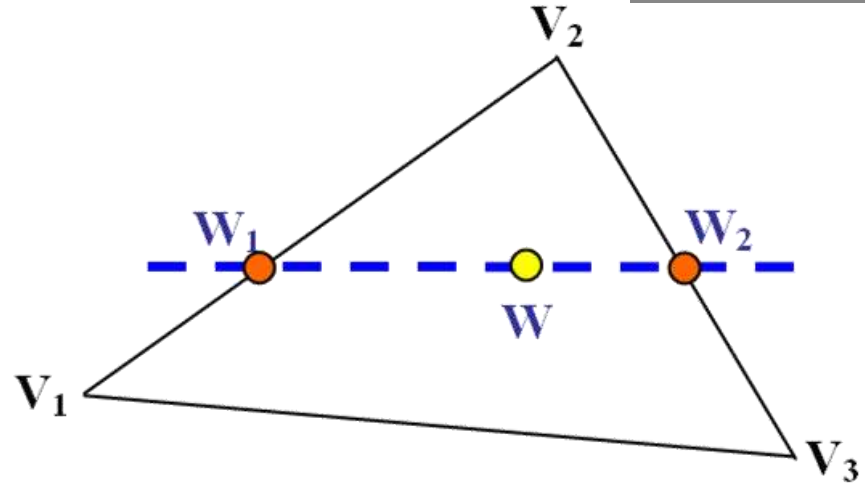
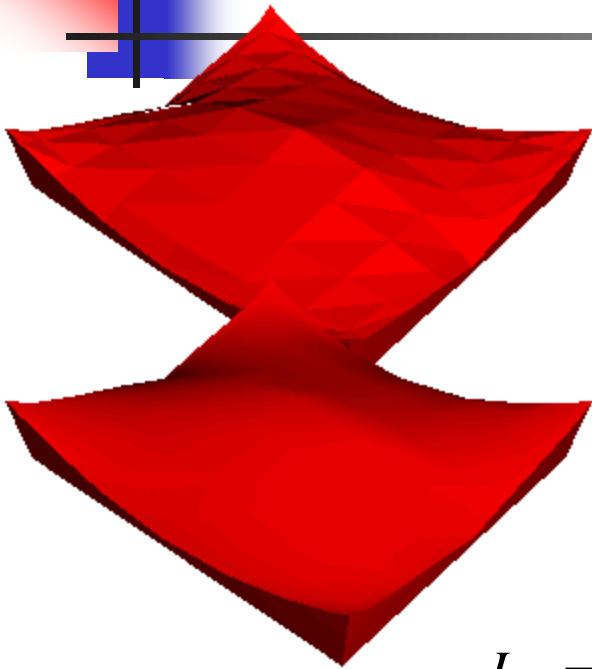
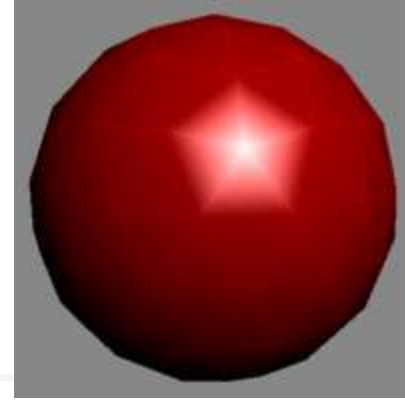
4. Протяженный источник:

требуется знать геометрию источника излучения, моделирование которого может осуществляться суммированием вкладов от отдельных точечных источников, заменяющих фрагменты излучающей поверхности

5. Направленный источник:

может быть любой из вышеперечисленных источников, излучение которого ограничено заданным телесным углом или маской

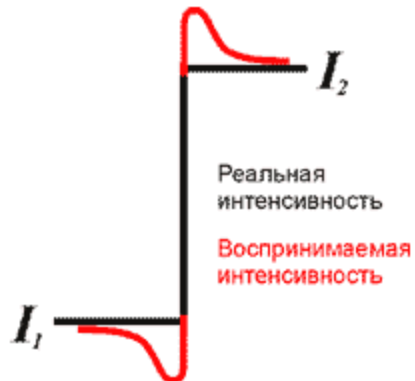
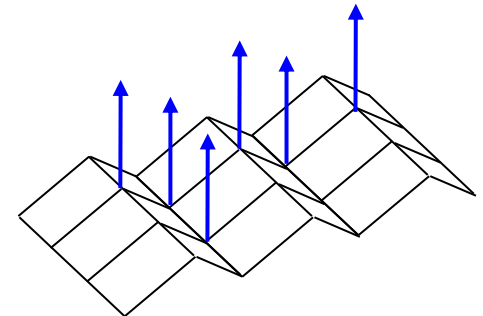
Реалистичная закраска граней: метод Гуро (Gouraud shading)



$$I_{W_1} = (1-t_1)I_{V_1} + t_1I_{V_2}, \text{ где } t_1 = \frac{|W_1V_1|}{|V_1V_2|}$$

$$I_{W_2} = (1-t_2)I_{V_3} + t_2I_{V_2}, \text{ где } t_2 = \frac{|W_2V_3|}{|V_2V_3|}$$

$$I_W = (1-t_3)I_{W_1} + t_3I_{W_2}, \text{ где } t_3 = \frac{|WW_1|}{|W_1W_2|}$$

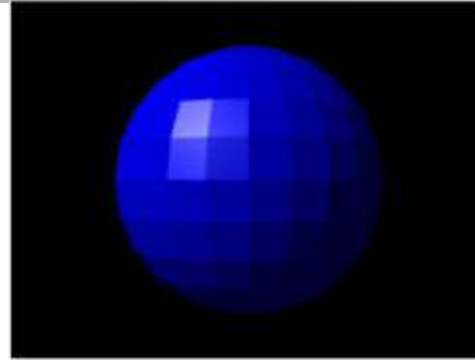


Реалистичная закраска граней: метод Фонга (Phong shading)

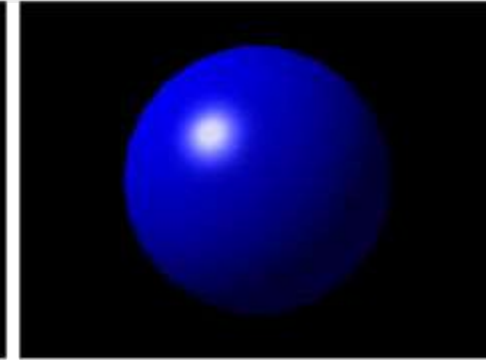
$$\vec{n}_{W_1} = (1 - t_1)\vec{n}_{V_1} + t_1\vec{n}_{V_2}, \text{ где } t_1 = \frac{|W_1V_1|}{|V_1V_2|}$$

$$\vec{n}_{W_2} = (1 - t_2)\vec{n}_{V_3} + t_2\vec{n}_{V_2}, \text{ где } t_2 = \frac{|W_2V_3|}{|V_2V_3|}$$

$$\vec{n}_W = (1 - t_3)\vec{n}_{W_1} + t_3\vec{n}_{W_2}, \text{ где } t_3 = \frac{|WW_1|}{|W_1W_2|}$$

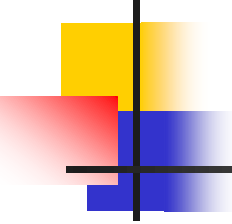


FLAT SHADING

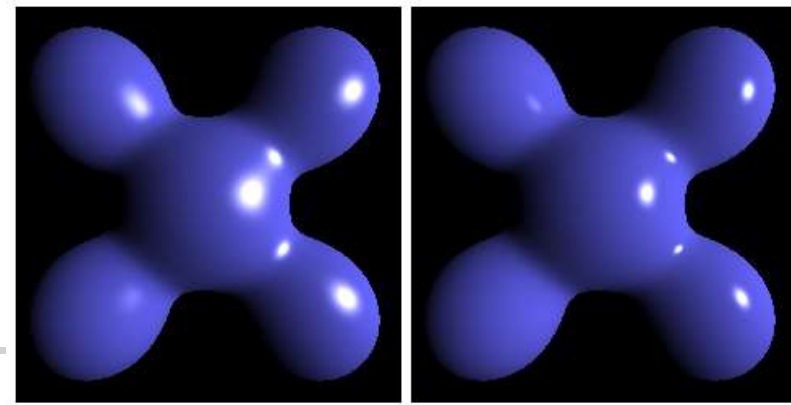


PHONG SHADING

1. Определяются нормали к граням
2. По нормальям к граням определяются нормали в вершинах
3. В каждой точке закрасиваемой грани определяется интерполированный вектор нормали
4. По направлению векторов нормали определяется цвет точек грани в соответствии с выбранной моделью отражения света



Реалистичная покраска граней: другие модели



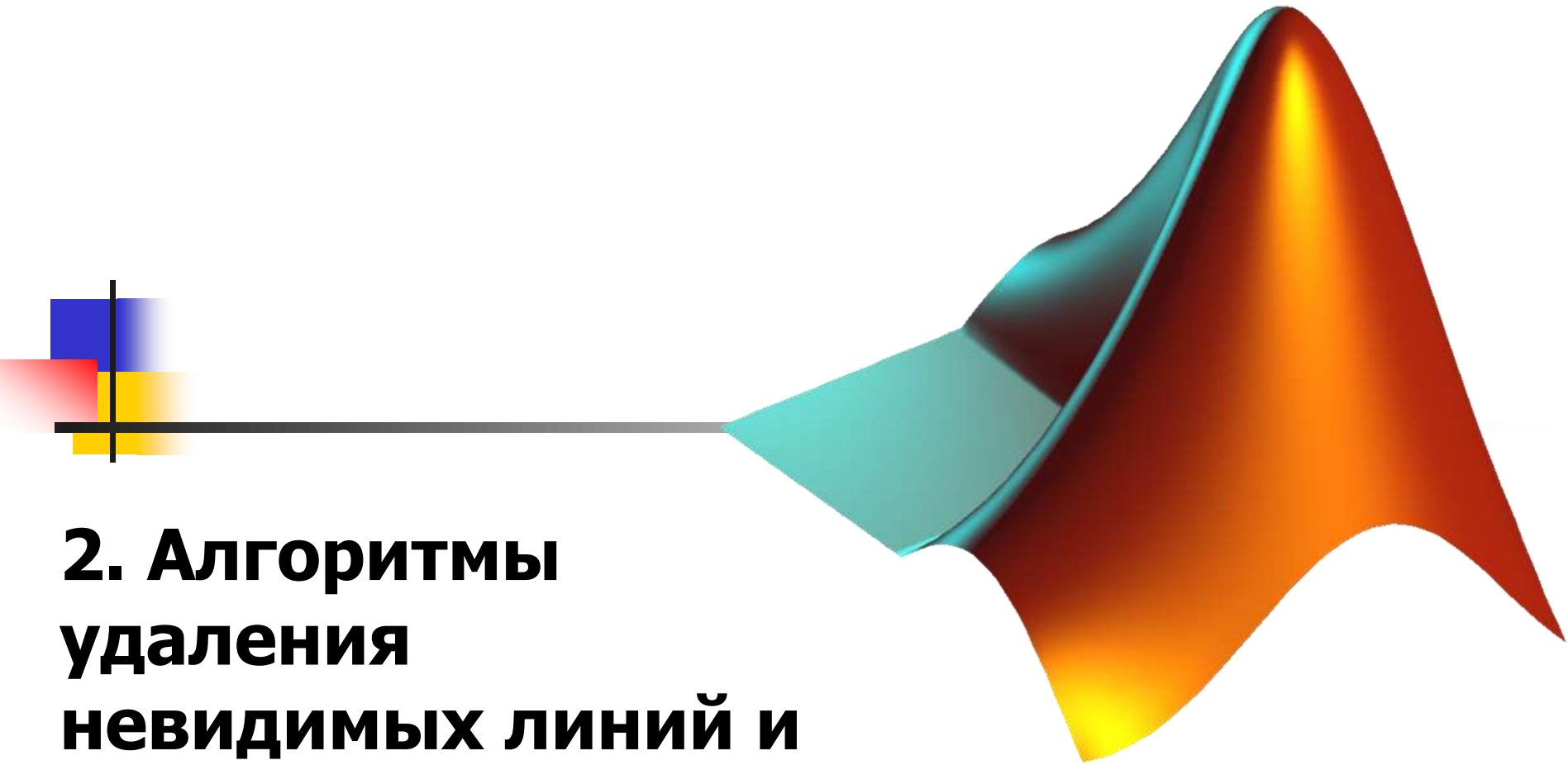
Blinn-Phong

Phong

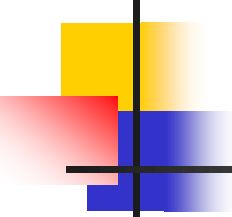
Модель затенения Blinn видоизменяет размер зеркальной подсветки модели Phong в зависимости не только от направления на источник света, но и от направления на наблюдателя. Чем меньше угол между поверхностью и направлением на наблюдателя, тем больше размер подсветки.

Модель затенения Cook-Torrance является логичным развитием модели Blinn, делая подсветку зависящей еще и от длины волны. Эта модель позволяет получать дополнительные цветовые переходы на краях подсветки для шероховатых металлических и пластиковых материалов.

Модель затенения Ward позволяет определять преимущественное направление шероховатостей поверхности и изменять форму подсветки в зависимости от такого направления.



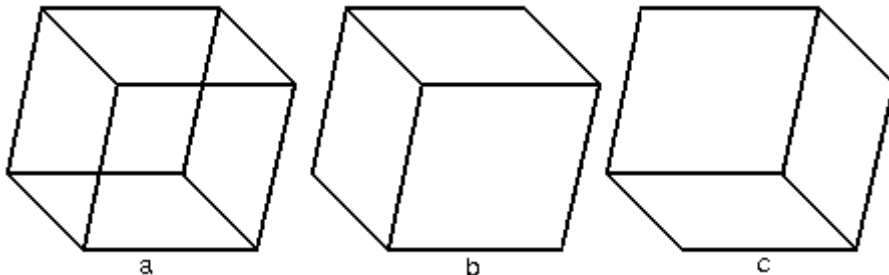
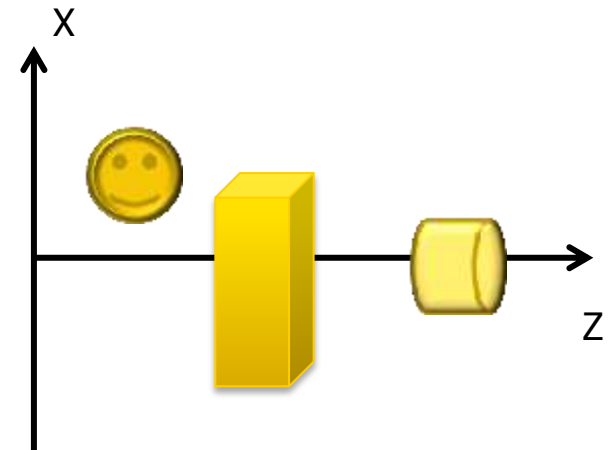
2. Алгоритмы удаления невидимых линий и поверхностей



Основная идея – сортировка по глубине

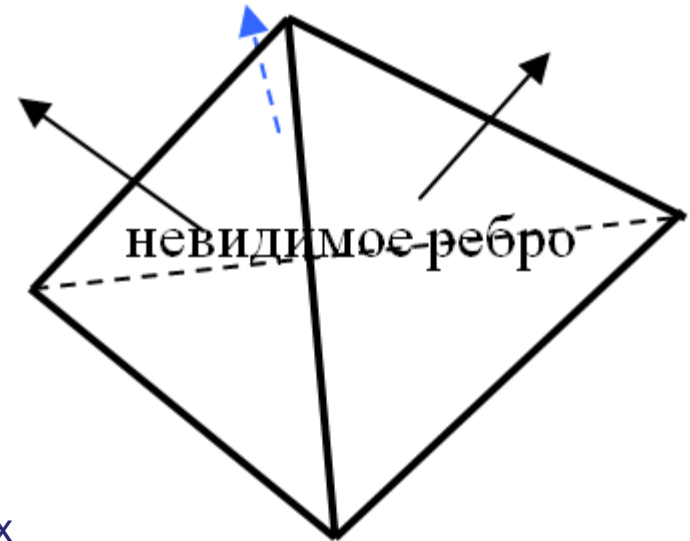
Алгоритмы удаления невидимых линий и поверхностей служат для определения *ребер, поверхностей* или *тел*, которые видимы или невидимы для *наблюдателя*, находящегося в заданной точке пространства.

- 1. Алгоритмы объектного пространства:**
за основу берется система мировых координат.
- 2. Алгоритмы пространства изображений:**
за основу берется система двумерных экранных координат.



Алгоритм Робертса

нелицевая грань



Если вектор нормали грани составляет с вектором направления проецирования тупой угол, эта грань невидима и называется **нелицевой**. Пересечение двух нелицевых граней даёт невидимое (нелицевое) ребро.

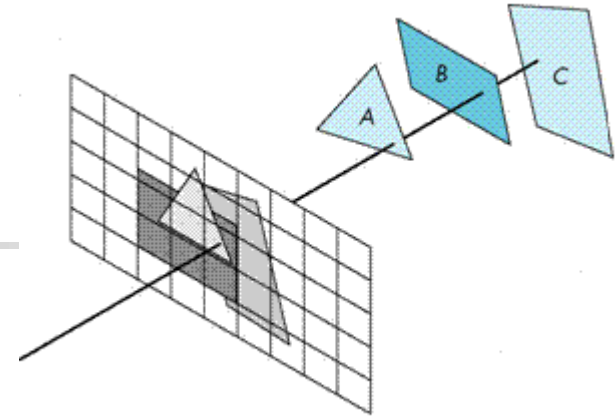
1. Все рассчитываемые объекты сцены должны быть выпуклыми. В частности – каждая грань есть выпуклый многогранник.
2. Из каждого тела удаляются рёбра и грани, являющиеся нелицевыми.
3. Каждое из оставшихся видимых рёбер каждого тела сравнивается с каждым из оставшихся видимых тел для определения того, какая его часть экранируется этими телами.

Временные затраты $\sim N^2$,
где N – число всех граней
сцены.



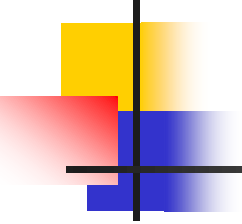
Алгоритм Z-буфера

Сопоставим каждому пикселю (x, y) картинной плоскости, кроме цвета, хранящегося в видеопамяти, его **расстояние до картинной плоскости** вдоль направления проектирования z (глубину). Изначально массив глубин инициализируется $+\infty$. Для вывода на картинную плоскость произвольной грани она переводится в **свое растровое представление** на картинной плоскости и для каждого пикселя этой грани находится его глубина. В случае, если эта глубина меньше значения глубины, хранящегося в z-буфере, **пиксель рисуется** и его глубина заносится в z-буфер.



Скорость работы алгоритма зависит от предварительной сортировки объектов сцены по глубине.

Обычно информацию о глубине следует обрабатывать с высокой точностью, аппаратные реализации z-буфера обычно 32-разрядные.



Алгоритм Z-буфера - пример расчёта

Если известно уравнение плоскости, несущей каждую грань, то вычисление глубины каждого пикселя на сканирующей строке можно осуществить итерационным методом.

$$aX + bY + cZ + d = 0$$

(X, Y, Z) – экранные координаты!

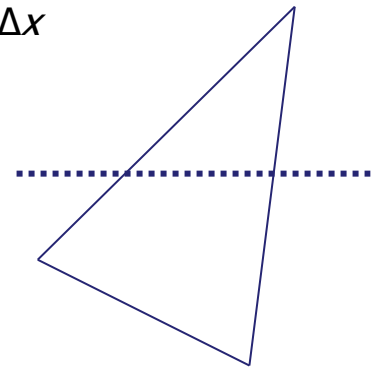
$$Z = -(aX + bY + d)/c$$

Для сканирующей строки $Y = \text{const}$, $X_1 = X + \Delta x$

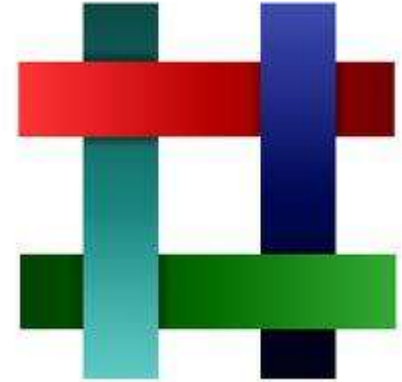
Тогда $Z_1 - Z = a(X - X_1)/c$, подставив выражение для X_1 : $Z_1 = Z - (a/c) \Delta x$

Учитывая, что $\Delta x = 1$, получаем:

$$Z_1 = Z - (a/c).$$



Алгоритмы упорядочения

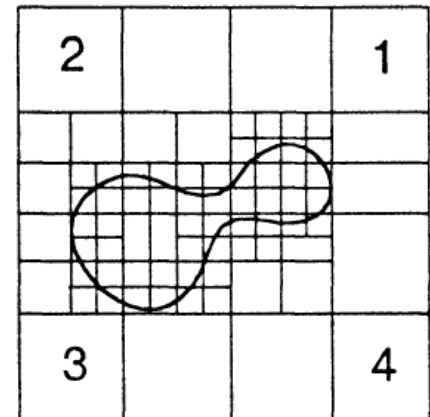


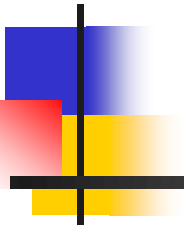
1. Алгоритм Варнока (Warnock)

В пространстве изображения рассматривается окно и решается вопрос о том, пусто ли оно, или его содержимое достаточно просто для визуализации. Если это не так, то окно разбивается на фрагменты до тех пор, пока содержимое подокна не станет достаточно простым для визуализации, или его размер не достигнет требуемого предела разрешения

2. Алгоритм художника

Каждая выводимая грань сразу же закрашивается, причём грани выводятся в порядке приближения к картинной плоскости. Таким образом, грани, расположенные ближе к плоскости экрана, выводятся позже и закрывают собой ранее построенное изображение.

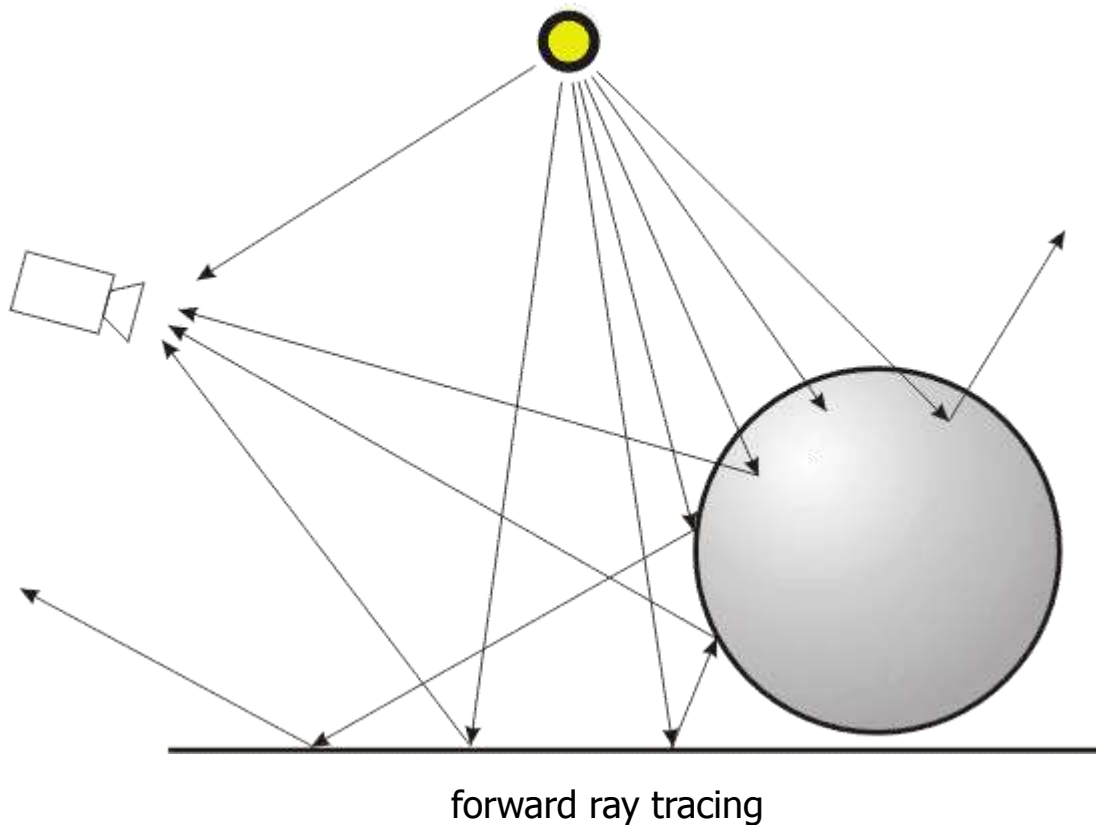
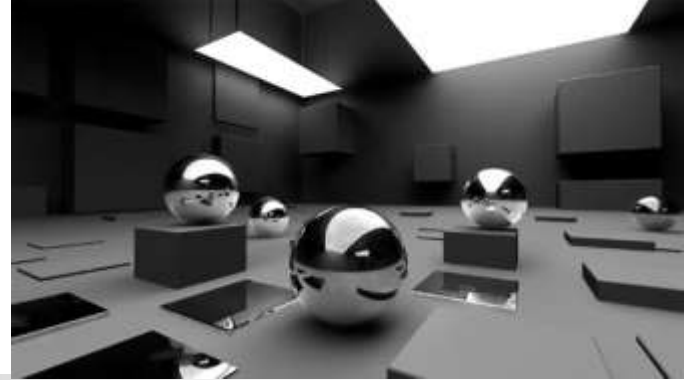




3. Принципы создания реалистичных изображений

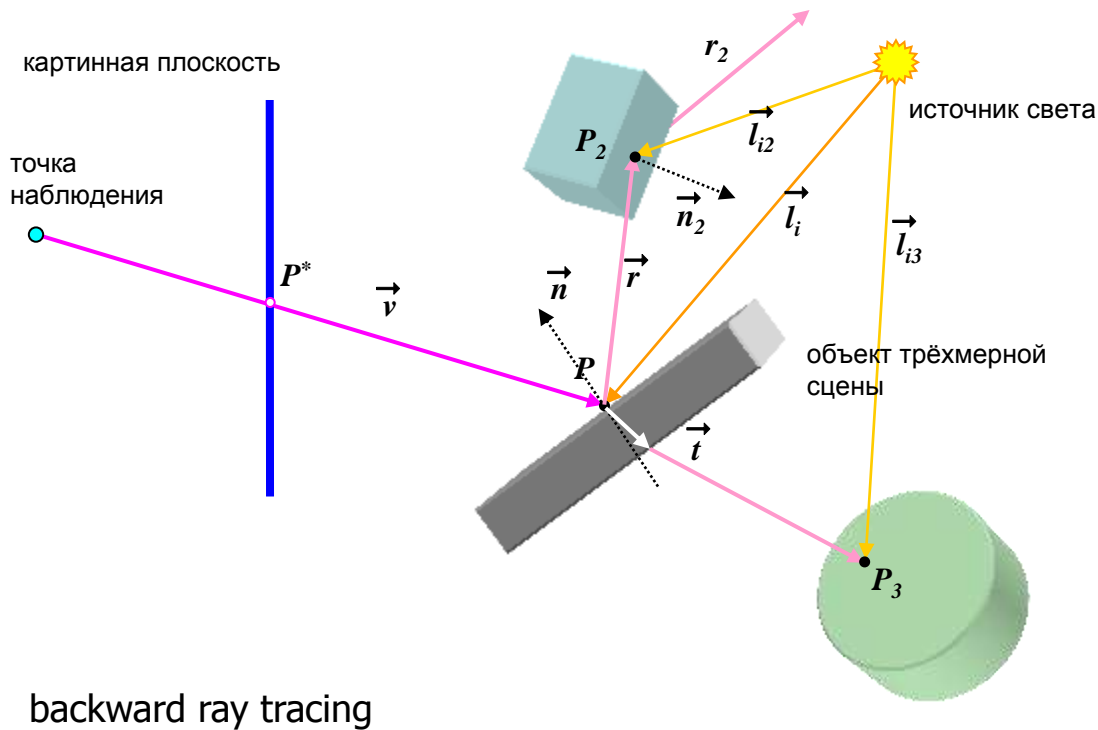
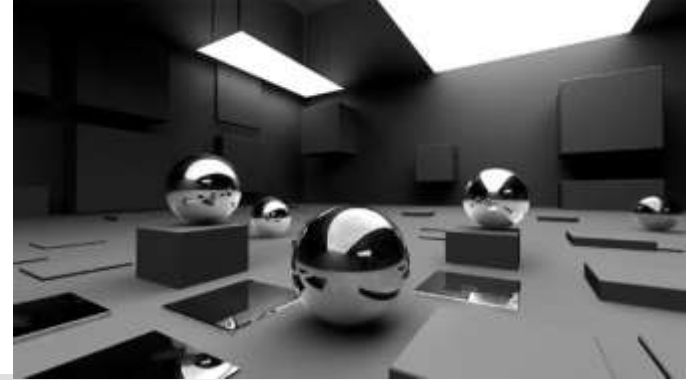


Модели трассировки лучей: прямая трассировка



Рассчитываются пути лучей света, идущие от источников до объектов сцены. Если такой первичный луч попадает в прозрачный объект, то преломляясь и теряя интенсивность, он идёт дальше и, возможно, попадает на другой объект, в свою очередь отражаясь от него. Так, многократно отражаясь и преломляясь, какая-то часть лучей приходят в точку наблюдения – т.е. эти лучи определяют цвет некоторых пикселей экранной плоскости.

Модели трассировки лучей: обратная трассировка



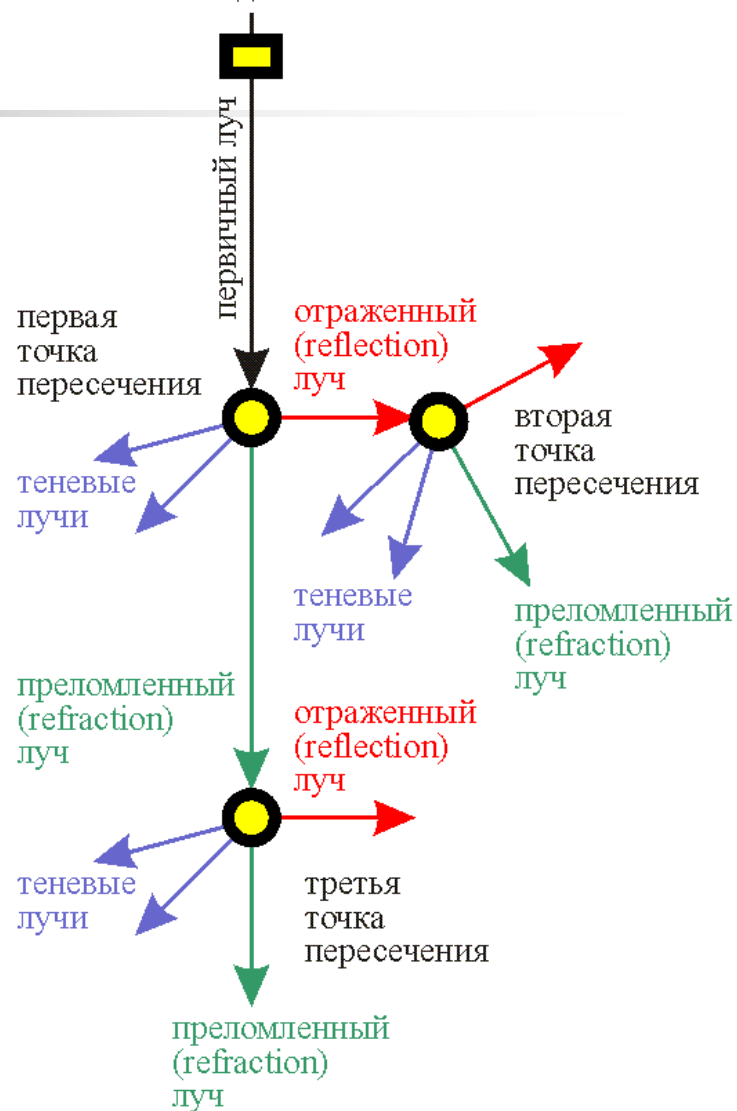
Каждый луч вдоль заданного направления продляется от наблюдателя вглубь трехмерной сцены, и для каждой траектории выполняется проверка на пересечение со всеми объектами сцены и с отсекающими плоскостями. Расчёт цвета пикселя P^* в общем случае определяется освещённостью всех поверхностей сцены, видимых по лучу зрения с учётом многократного отражения, преломления и поглощения.

Модели трассировки лучей: обратная трассировка

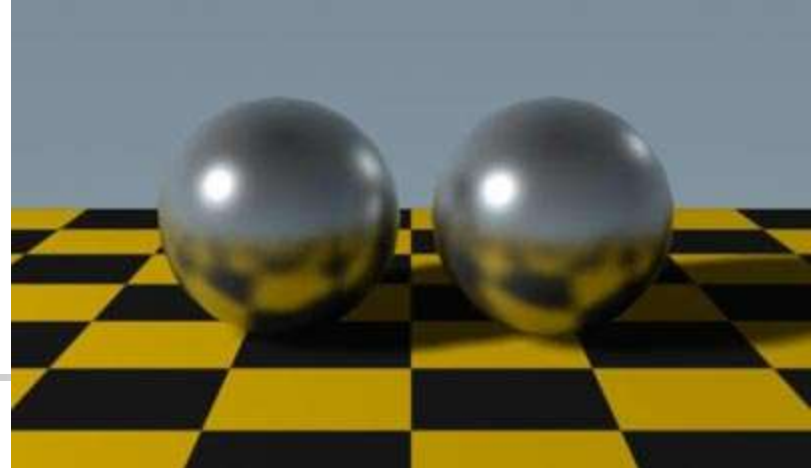
Вариант расчётного соотношения:

$$\begin{aligned}
 I &= k_a I_a + k_d \sum_i \frac{I_{l_i}}{d_{l_i} + k_{l_i}} (\vec{n} \cdot \vec{l}_i) + \\
 &+ k_s \sum_i \frac{I_{l_i}}{d_{l_i} + k_{l_i}} \cdot \frac{R(\theta_{l_i}) \cdot D(\gamma)}{(\vec{n} \cdot \vec{l}_i) \cdot (\vec{n} \cdot \vec{v})} + \\
 &+ k_s I_r R(\theta_r) \cdot e^{-\alpha_r d_r} + k_t I_t T(\theta_t) \cdot e^{-\alpha_t d_t}
 \end{aligned}$$

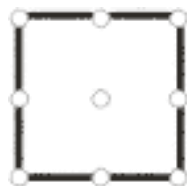
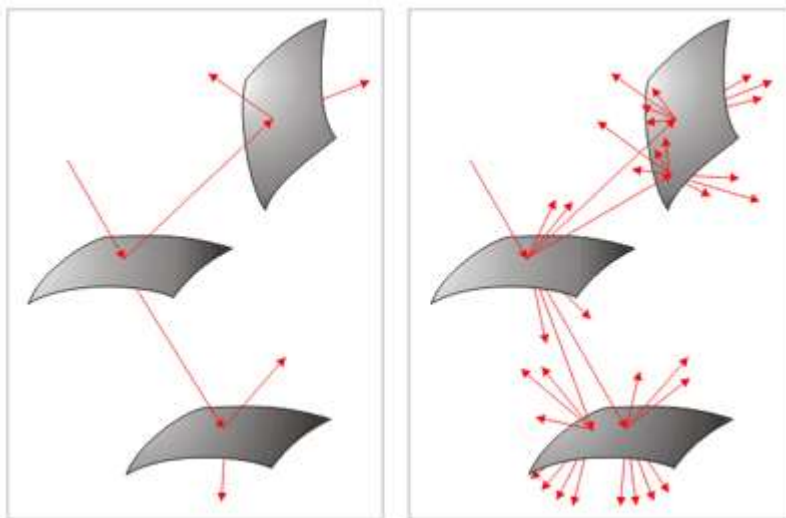
наблюдатель
×
пиксел видового окна



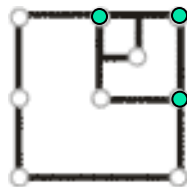
Модели трассировки лучей: ЭВОЛЮЦИЯ АЛГОРИТМОВ



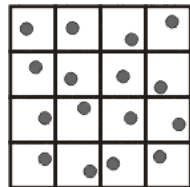
- supersampling
- adaptive supersampling
- stochastic sampling
- distribute ray tracing (DRT)



способ расчета
пикселя изображения $\frac{1}{n} \sum_{i=1}^n I_i$

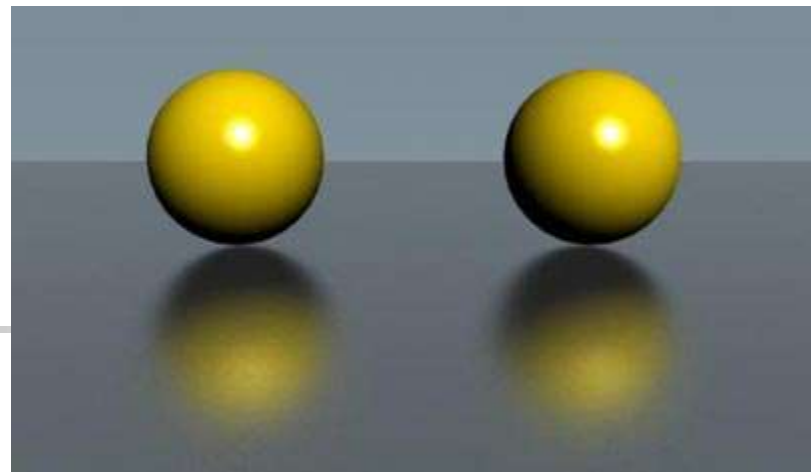


предварительное сравнение
интенсивности в точках



случайный выбор позиции по
субпикселю и разброса для
угла в каждом луче зрения

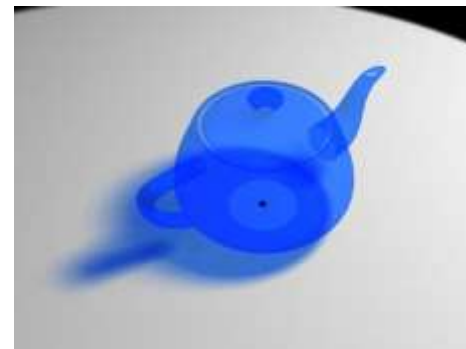
Модели трассировки лучей: DRT



DRT позволяет просчитывать такие эффекты, как:

- gloss (или scattered reflectance – размытые отражения)
- translucency (размытая прозрачность)
- penumbra (мягкая тень с размытыми краями)
- depth of field (размывание объектов в зависимости от их положения относительно фокуса камеры)
- motion blur (истинное размытие в движении)

Мягкие тени в DRT получаются в результате вычисления видимости лучей, направленных на источник освещения. Освещенность точки вычисляется как интенсивность освещения, умноженная на отношение количества достигших источника лучей к общему количеству испущенных из точки лучей.



Модель расчета освещенности: МЕТОД ИЗЛУЧАТЕЛЬНОСТИ

$$B_i = E_i + \rho_i \sum_{\substack{j=1 \\ j \neq i}}^n F_{ij} B_j, \quad i = 1 \dots n.$$

$$\sum_{j=1}^n F_{ij} < 1$$

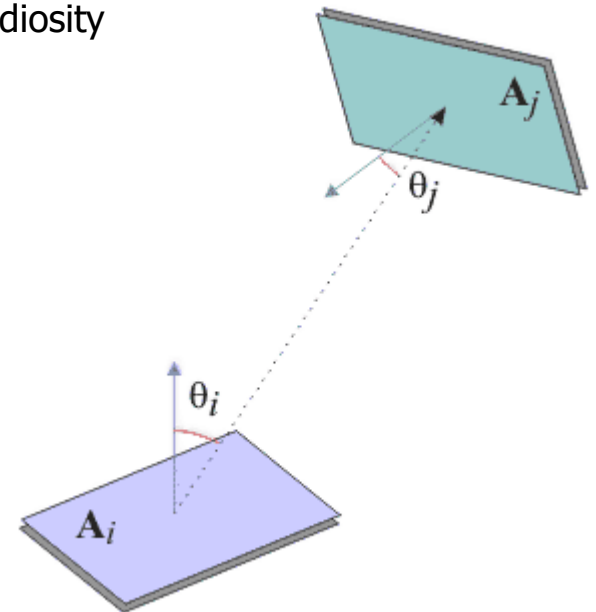
$$B_i^{(k+1)} = E_i + \rho_i \sum_{\substack{j=1 \\ j \neq i}}^n F_{ij} B_j^{(k)}, \quad i = 1 \dots n.$$

$$B_i^{(0)} = E_i$$

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \varphi_i \cdot \cos \varphi_j}{\pi \cdot r^2} H_{ij} dA_j dA_i$$

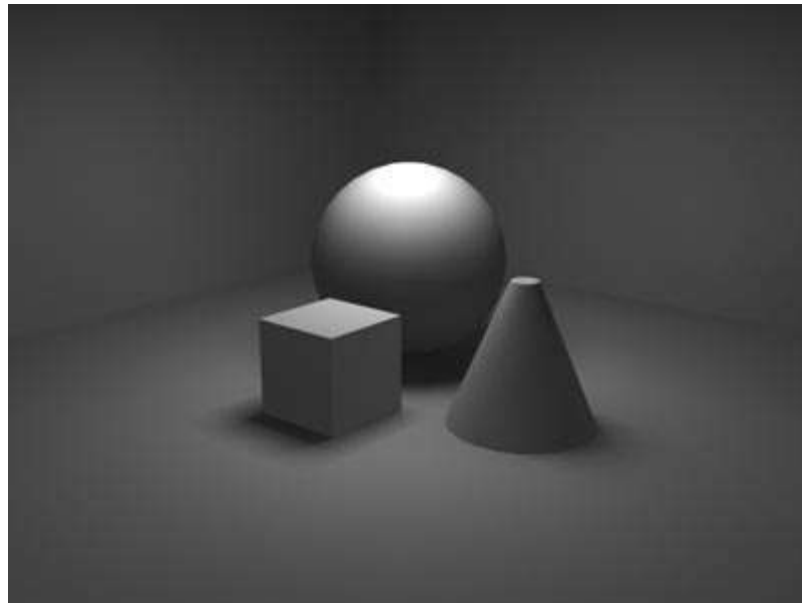
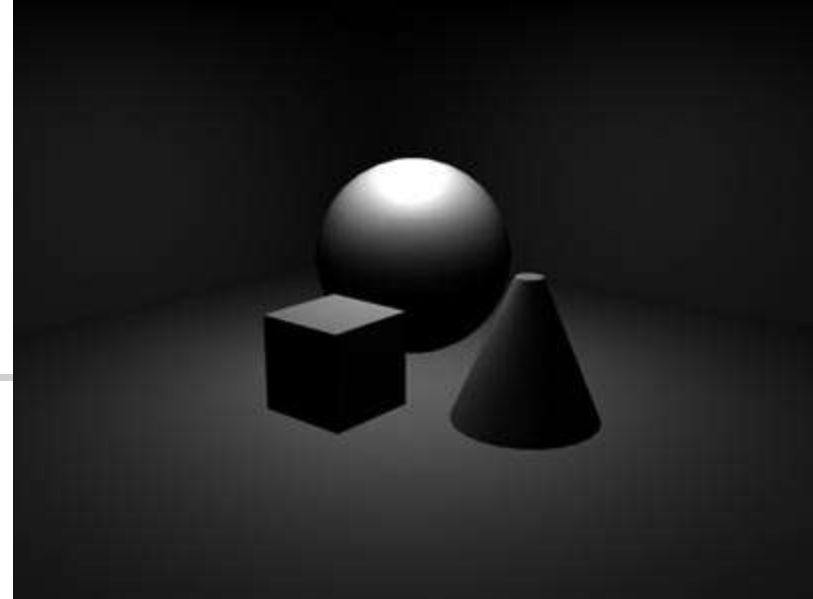


Radiosity

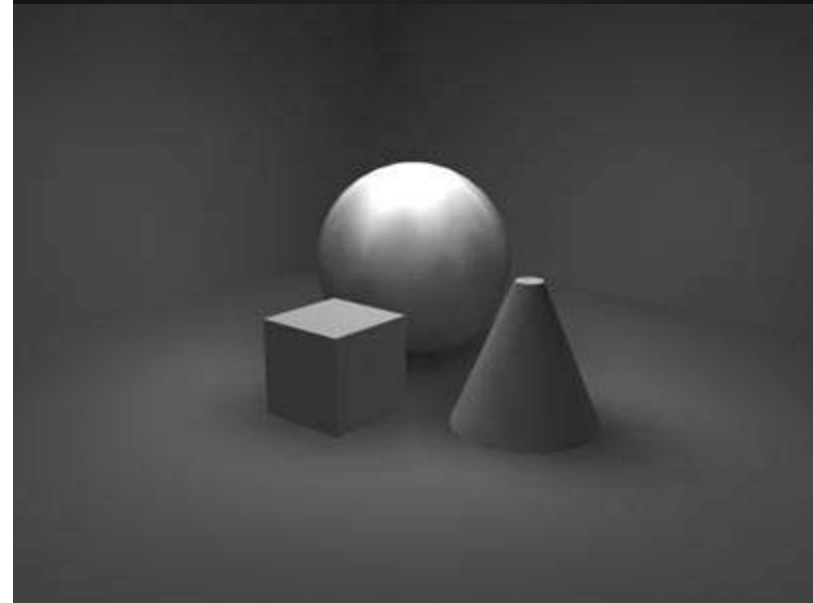




Модели расчета изображений: примеры



Соединение расчетов методами трассировки лучей и излучательности



Сверху – метод обратной трассировки,
снизу – метод излучательности

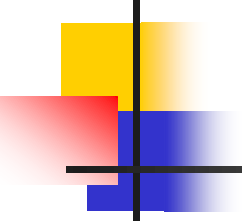


Модель расчета освещенности: ЭВОЛЮЦИЯ АЛГОРИТМОВ



- matrix radiosity (решение СЛАУ методом Зейделя)
- progressive refinement (сортировка фрагментов по освещенности)
- wavelet radiosity (вариация размеров фрагментов)
- Stochastic Relaxation Radiosity (SRR, решение СЛАУ методом Монте-Карло)
- **Global Illumination using Photon Maps** (метод «фотонных карт»)

Метод фотонных карт состоит в расчете изображения трехмерных объектов в два прохода. На первом проходе выполняется **прямая** трассировка испущенных источником света полных траекторий лучей от момента их испускания источником через все возможные многократные взаимодействия с поверхностями и до момента поглощения света какой-либо из них. Результат отслеживания траекторий сохраняется в записях "фотонных карт" поверхностей. На втором проходе выполняется расчет освещенности пикселей изображения методом **обратного рейтресинга со стохастическим сэмпингом** с использованием данных фотонных карт.



Модель расчета освещенности:
взаимодействие света со средой



bidirectional surface scattering
reflectance distribution function