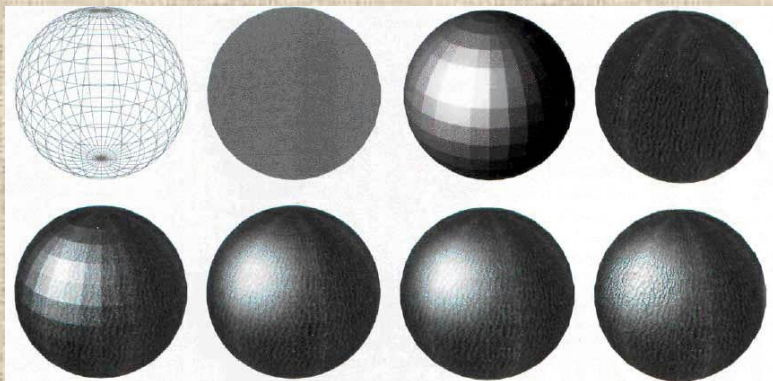

Затенение и тонирование

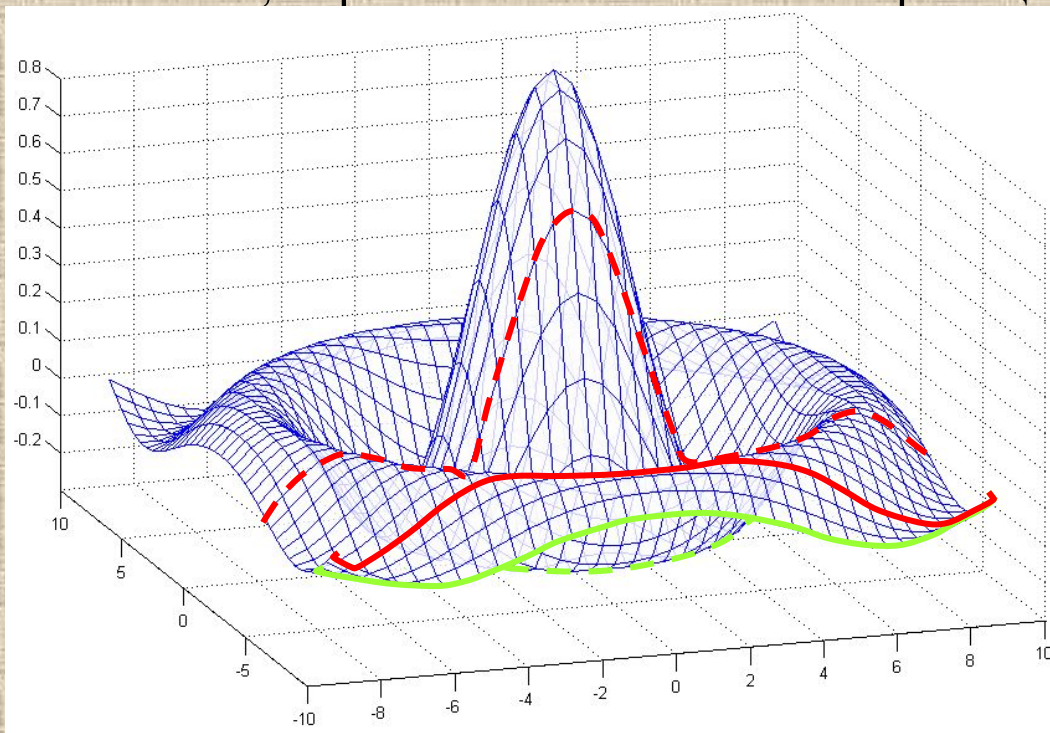


Будак Владимир Павлович,
Национальный исследовательский
университет «МЭИ»
кафедра светотехники



Алгоритм плавающего горизонта

Параллельная проекция поверхности функции двух переменных $z=f(x,y)$ - построение сетки граней, образуемой сечением последовательностью плоскостей, параллельных плоскости проекций



$$Z_{\max}^k(x) = \max_{1 \leq i \leq k-1} Z_i(x),$$

$$Z_{\min}^k(x) = \min_{1 \leq i \leq k-1} Z_i(x)$$

- линии горизонта (High-LOw)

$$Z_k(x) < Z_{\min}^k(x),$$

$$Z_k(x) > Z_{\max}^k(x)$$

При выводе полутоновых изображений используют обратный порядок вывода граней по мере их приближения

Алгоритмы упорядочения

Алгоритмы затенений направлены на удаление проекций невидимых граней, ребер и точек, что существенно улучшает восприятие изображения

Все алгоритмы основываются на сортировке или упорядочиванию объектов так, что их отображение в порядке сортировки дает корректное изображение

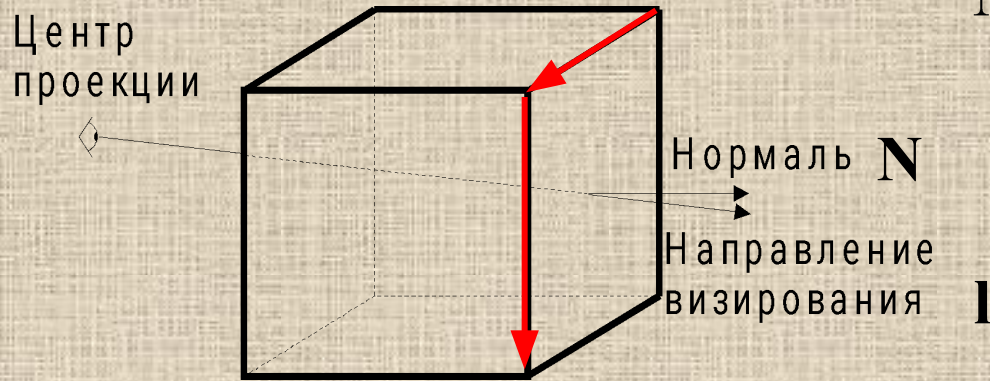
Общих алгоритмов нет, выделяют подходы:

1. Для каждого пикселя плоскости проекций определяются точки объекта, расположенные дальше всего от плоскости
трудоемкость $\sim N^2$
2. Сравнение точек в пространстве объектов и определение затенений
трудоемкость $\sim nN$

Анализ проще проводить в пространстве изображений

z-буфер - сортировка точек сцены по глубине с последующим выводом их в порядке приближения

Выбраковка (Culling) – отбрасывание нелицевых



$$\hat{\mathbf{N}} \cdot \hat{\mathbf{I}} = N_x l_x + N_y l_y + N_z l_z = \cos \theta$$

$$\theta > 90^\circ, \quad \text{если} \quad \hat{\mathbf{N}} \cdot \hat{\mathbf{I}} > 0;$$

$$\theta = 90^\circ, \quad \text{если} \quad \hat{\mathbf{N}} \cdot \hat{\mathbf{I}} = 0;$$

$$\theta < 90^\circ, \quad \text{если} \quad \hat{\mathbf{N}} \cdot \hat{\mathbf{I}} < 0.$$

$$\mathbf{N} = \mathbf{A} \times \mathbf{B} = \begin{vmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix}, \quad |\mathbf{A} \times \mathbf{B}| = AB \sin \theta = \sqrt{N_x^2 + N_y^2 + N_z^2}, \quad \hat{\mathbf{N}} = \frac{\mathbf{N}}{|\mathbf{N}|}$$

Для выпуклых многогранников процедуры удаления нелицевых граней достаточно полного анализа затенений

Сортировка по глубине

1. Отбрасывание нелицевых граней
2. сортировка оставшихся граней сцены по их расстоянию от центра проекции
3. вычерчивание объектов, в обратной последовательности определенному расстоянию: При этом каждая грань заполняется некоторым внутренним цветом и закрашивает предыдущий, как бы естественным образом его затеняя – алгоритм живописца

Алгоритм реализовать особенно просто, если при проектировании каждой из вершин сохранять значение координаты z' - расстояния до картинной плоскости



Для невыпуклых тел невозможно определить какая из граней ближе - дальше от центра проекции и случай пересечение граней

Невыпуклые тела

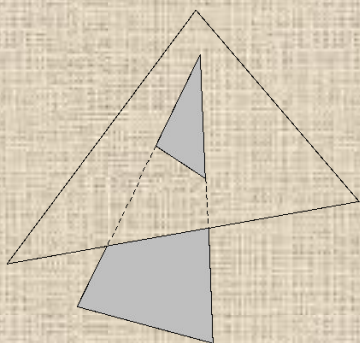
1. Поиск максимумов и минимумов по Z по всем вершинам граней и проверка, не является ли максимум одной меньше минимума другой
2. Поиск максимумов и минимумов по X и Y и проверка перекрываются ли их области проекций
3. Проверка, не является ли одна из граней полностью ниже или выше другой

$$d = (\mathbf{r}_Q - \mathbf{r}_P, \hat{\mathbf{N}})$$

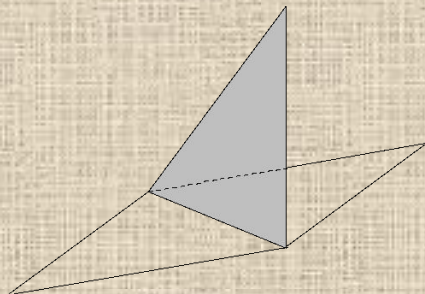
4. Определение пересечения граней друг другом выполняется по анализу пересечения проекций ребер граней – если совпадает координата z
5. Если ребра не пересекаются, но так как проверка на положение граней выше – ниже уже выполнена,

то возможны только три ситуации

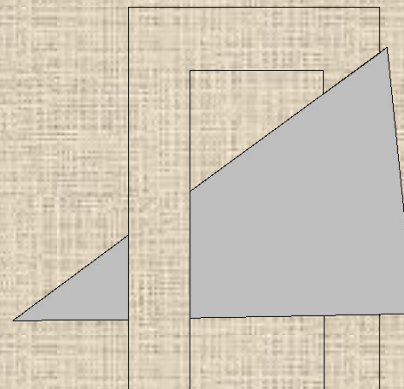
Особые ситуации



а.



б.



с.

- а) одна грань окружается другой и пересекается ее внутри;
- б) одно из ребер одной грани целиком лежит в другой грани;
- с) линия пересечения плоскостей граней для одной из них находится вне пределов, очерченных ребрами.

Можно заметно сократить количество проверок, если разбить плоскость проекций на сетку равных прямоугольных ячеек

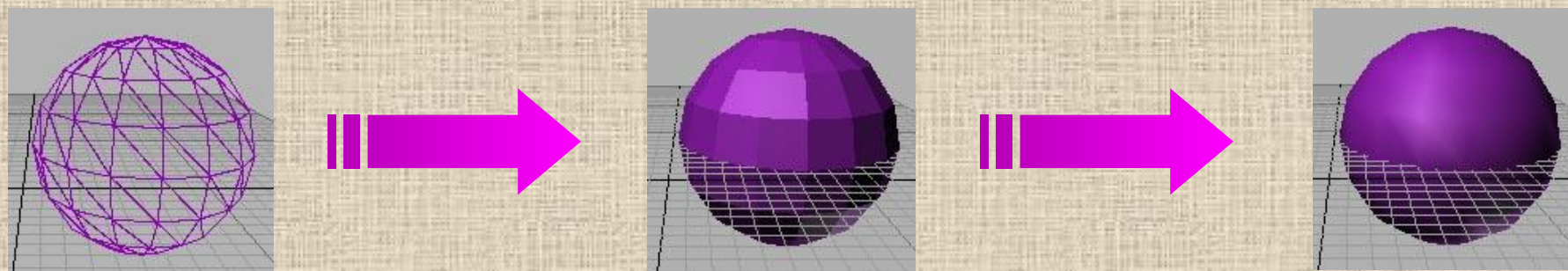
Закрашивание (тони́рование)

Закрашивание – определение яркости граней с учетом распределения освещенности по поверхности и законов отражения

Чтобы увидеть сцену, ее нужно осветить, т.е. трехмерная сцена дополняется источниками света

Строгий и реалистичный алгоритм закрашивания приводит к задачам глобального освещения и уравнению визуализации.

На первом этапе пренебрежем многократными отражениями от поверхностей сцены.



Проанализируем основные понятия в алгоритмах закрашивания при учете распределения только прямой освещенности

Диффузное отражения

$$L_d = \frac{\rho_d}{\pi} \int_{(\hat{\mathbf{I}}, \hat{\mathbf{N}}) \leq 0} L_i(\hat{\mathbf{I}}) |(\hat{\mathbf{I}}, \hat{\mathbf{N}})| d\hat{\mathbf{I}} = \frac{\rho_d I_0}{\pi r^2} |(\hat{\mathbf{I}}, \hat{\mathbf{N}})|$$

Ограниченный динамический диапазон яркостей монитора экрана приводит к нереалистическому изображению

$$L_d = \frac{\rho_d I_0}{\pi(r + d)} |(\hat{\mathbf{I}}, \hat{\mathbf{N}})|$$

При освещении только прямым светом от точечных источников сцена имеет очень резко-контрастный характер с «марсианскими» тенями:

неосвещенные грани являются полностью темными

В реальности все тени сглаживаются многократными переотражениями — глобальное освещение

$$L_d = L = \frac{\rho_d I_0}{\pi r^2} |(\hat{\mathbf{I}}, \hat{\mathbf{N}})| + \frac{\rho_a}{\pi} E_a$$

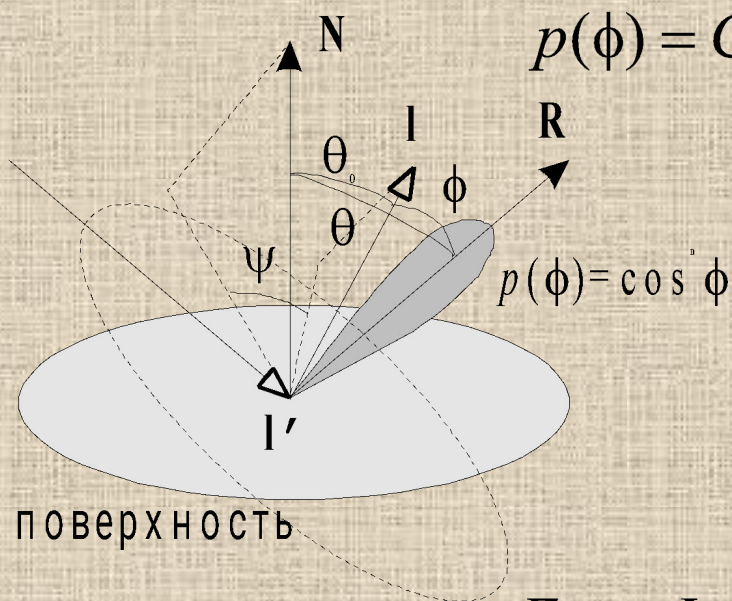
Ambient — источник подсветки — имитация многократных отражений

Зеркальное отражение

Буи-Туонг Фонг (*Bui-Tong Phong*) предложил аппроксимацию индикатрисы диффузно-направленного отражения законом

$$p(\phi) = C_0 \cos^n \phi = C_0(\hat{\mathbf{R}}, \hat{\mathbf{I}}), \quad \frac{1}{\pi} \int_{\Omega_+} p(\hat{\mathbf{I}}, \hat{\mathbf{I}}')(\hat{\mathbf{N}}, \hat{\mathbf{I}}') d\hat{\mathbf{I}}' = 1$$

$$\hat{\mathbf{R}} = \hat{\mathbf{I}}' - 2\hat{\mathbf{N}}(\hat{\mathbf{I}}', \hat{\mathbf{N}})$$



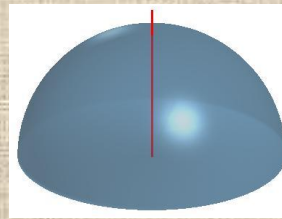
$$\rho(\hat{\mathbf{I}}, \hat{\mathbf{I}}') = \rho_s w(\hat{\mathbf{N}}; \hat{\mathbf{I}}, \hat{\mathbf{I}}') \frac{n+2}{2|(\hat{\mathbf{N}}, \hat{\mathbf{I}}')|} (\hat{\mathbf{R}}, \hat{\mathbf{I}})^n$$

$$L = \rho_a \frac{E_a}{\pi} + \frac{I_0}{\pi r^2} \left\{ \rho_d(\hat{\mathbf{I}}, \hat{\mathbf{N}}) + \rho_s \frac{n+2}{2} \left((\hat{\mathbf{I}}, \hat{\mathbf{I}}') - 2(\hat{\mathbf{I}}, \hat{\mathbf{N}})(\hat{\mathbf{I}}', \hat{\mathbf{N}}) \right)^n \right\}$$

При наличии нескольких источников света суммирование отраженного света в данном направлении от каждого

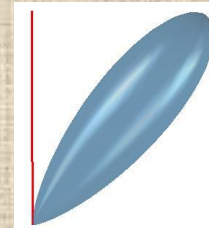
Коэффициент яркости материалов

$$\rho(\mathbf{r}; \hat{\mathbf{l}}', \hat{\mathbf{l}}) =$$



Diffuse

+



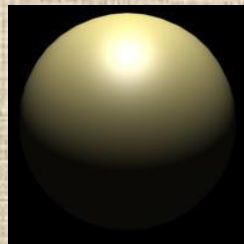
Specular

$$\rho_0 \cos^n \alpha$$

Диффузное
отражение



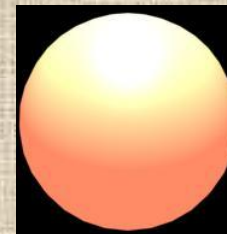
Зеркальное
отражение



Пропускание



Само
свечение



Металл



*Создание и редактирование 3D сцены на основе алгоритмов
T&L, а визуализация излучательность & трассировка*

Тени

Изображение трехмерных сцен с тенями выглядит значительно реалистичнее

Точечные источники создают только резко очерченную тень, а пространственно распределенные создают как тень, так и полутень.

Тень бывает собственная и проекционная

1. Собственная тень связана с тем, что объект препятствует проникновению света к некоторым своим граням
 - находятся аналогично нелицевым граням при наблюдении из точки расположения источника
2. Если один объект препятствует проникновению света на другой объект, то это проекционная тень
 - получается в результате центральной проекции объекта на соседние объекты

Поскольку удаление невидимых граней и поиск теней аналогичен, то оба эти алгоритма соединяют вместе

Цвет

$$R = \int_{\lambda_1}^{\lambda_2} \rho_1(\lambda) \cdot \dots \cdot \rho_n(\lambda) \varphi(\lambda) r(\lambda) d\lambda$$

Метод дискретный ординат Hardy

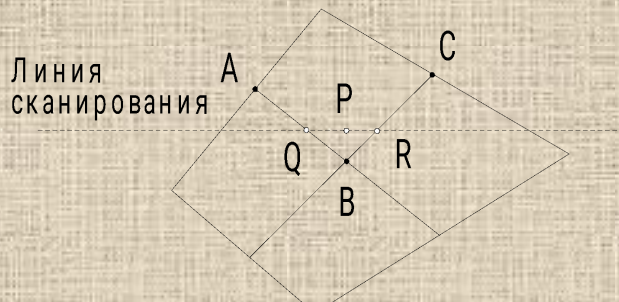
$$R = \int_{\lambda_1}^{\lambda_2} \rho_1(\lambda) \cdot \dots \cdot \rho_n(\lambda) \varphi(\lambda) r(\lambda) d\lambda \approx \sum_{i=1}^N \left(\prod_{k=1}^M \rho_k(\lambda_i) \right) \varphi(\lambda_i) r(\lambda_i) \Delta\lambda_i$$

Простейший случай три длины волны – RGB – задается цвет источника и объекта

Каждый цвет преобразуется независимо

Сглаживание

1. Сеткой граней представляются гладкие поверхности и алгоритм приводит к неестественному граненому изображению
2. Сетка аналогична табличному заданию функции – интерполяция - сглаживание
3. Для скорости используется простейшая линейная интерполяция
4. Алгоритмы сглаживания:
 - а) Гуро (H.Gouraud) – линейная интерполяция освещенности между вершинами – диффузные объекты
 - б) Фонга (Bui-Tong Phong) – линейная интерполяция направления нормали между вершинами – блик объектов



$$L_Q = tL_A + (1-t)L_B, \quad t = \frac{AQ}{AB} \quad L_R = tL_B + (1-t)L_C, \quad t = \frac{BR}{BC}$$
$$L_P = tL_Q + (1-t)L_R, \quad t = \frac{QP}{QR} \quad \hat{N}_Q = t\hat{N}_A + (1-t)\hat{N}_B$$

*В компьютерной графике под текстурой понимается
детализация строения поверхности*

Текстуры

Повышение реалистичности изображения сцены приводит к проблеме огромного объема работы:

- Ворсинки ковра
- Листья дерева
- Рисунок поверхности и т.д.



Текстура коэффициента яркости



Текстура свечения



Текстура пропускания



Текстура неровностей

*В компьютерной графике под текстурой понимается
детализация строения поверхности*