

dfgldfjld,mhncv;lfdf;lfkgjldfbbcmvb,m;dlf /f/gfgbf mnk;dfdlfgdkfgjklsgdhflskdjfhgkdgf;[iwerutwerkejrgeggb,vmf.,v

W
i
s
d
h
.
m
s
n
d
f
d
f
s
/
s
d
f
.
.
d
f
d
/
:
^
%
^
[
]
L
H
J
H
S
f
h
s
d
B
^
B
B
L



OpenGL

***Методы и алгоритмы
трехмерной графики***

Лекция № 5

Содержание

Модели описания поверхностей

- *Аналитическая модель*
- *сплайн Безье*

Векторная полигональная модель

- *Способы организации хранения информации об объекте.*
- *Линейно-узловая модель*

Скорость вывода. Топологический аспект.

Воксельная модель

Равномерная сетка

Неравномерная сетка. Изолиния.

Визуализация объемных изображений

- *Каркасная визуализация.*
- *Показ с удалением невидимых линий*

Закрашивание поверхностей

Зеркальное отражение

Диффузное отражение

Вычисление нормалей и углов

Диффузное отражение

Зеркальное отражение

Метод Гуро

Метод Фонга

Преломление света

Модель идеального преломления

Треугольная призма

Трассировка лучей

Метод обратной трассировки лучей

Модели описания поверхностей

Аналитическая модель:

Варианты:

- 1) $Z = f(x, y);$
- 2) $F(x, y, z) = 0;$
- 3) $x = F_x(s, t), y = F_y(s, t), z = F_z(s, t);$

где s, t – параметры, F_x, F_y, F_z - функции, определяющие форму поверхности.

Пример: *Поверхность шара:*

- 1) $z = \pm \sqrt{R^2 - x^2 - y^2}$
- 2) $x^2 + y^2 + z^2 - R^2 = 0$
- 3) $x = R \sin(s) \cos(t);$
 $y = R \sin(s) \sin(t);$
 $z = R \cos(s).$

Для описания поверхностей часто используют сплайны. Сплайн – специальная функция, более всего пригодная для аппроксимации отдельных фрагментов поверхности. Несколько сплайнов образуют модель сложной поверхности. Обычно используют кубический сплайн.

$$\begin{aligned} X(s, t) = & a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t + a_{14}s^3 + \\ & + a_{21}s^2t^3 + a_{22}s^2t^2 + a_{23}s^2t + a_{24}s^2 + \\ & + a_{31}st^3 + a_{32}st^2 + a_{33}st + a_{34}s + \\ & + a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44}. \end{aligned}$$

Модели описания поверхностей

сплайн Безье

Разновидность сплайнов – сплайн Безье:

$$P(s, t) = \sum_{i=0}^m \sum_{j=0}^n C_m^i s^i (1-s)^{m-i} C_n^j t^j (1-t)^{n-j} P_{ij}$$

где P_{ij} – опорные точки-ориентиры, $0 \leq s \leq 1$, $0 \leq t \leq 1$, C_m^i , C_n^j

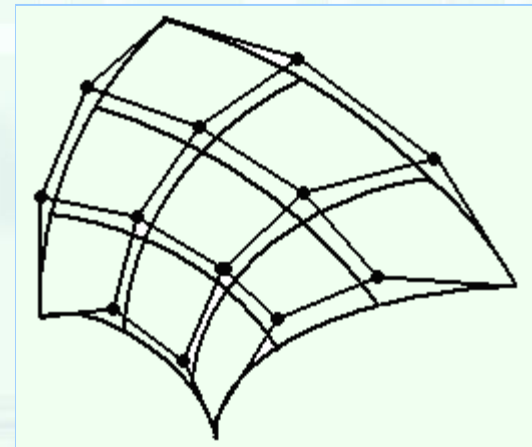
коэффициенты бинома Ньютона $C_a^b = \frac{a!}{b!(a-b)!}$

Кубический сплайн Безье $m=3$, $n=3$ определен на 16 точках-ориентирах

$$C_m^i, C_n^j = 1; 3; 3; 1 \quad i, j = 0; 1; 2; 3.$$

Достоинства: достаточно простое представление сложных поверхностей.

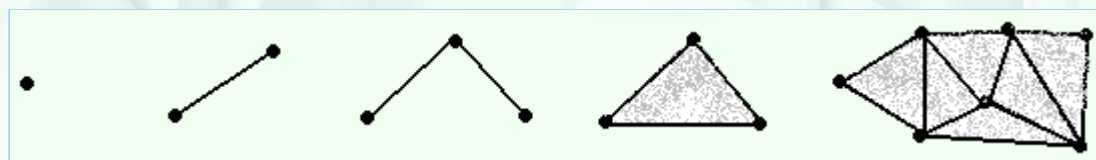
Недостаток: сложные формулы для вычисления функций.



Векторная полигональная модель

Элементы: вершина, отрезки прямых (вектора), полилинии, полигоны, полигональные поверхности.

Вершина (vertex) – главный элемент, определяется координатами (x_i, y_i, z_i) .



точка

вектор

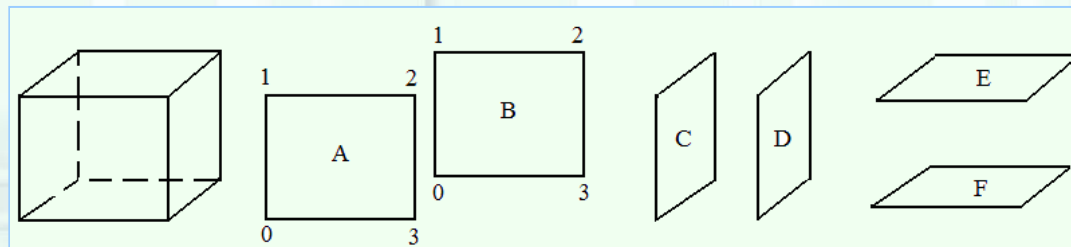
полилиния

полигон

полигональная
поверхность

Способы организации хранения информации об объекте.

Сохраняем все грани отдельно.



Грань А: $A = \{(x_{A0}, y_{A0}, z_{A0}), (x_{A1}, y_{A1}, z_{A1}), (x_{A2}, y_{A2}, z_{A2}), (x_{A3}, y_{A3}, z_{A3})\}$

Грань В: $B = \{(x_{B0}, y_{B0}, z_{B0}), (x_{B1}, y_{B1}, z_{B1}), (x_{B2}, y_{B2}, z_{B2}), (x_{B3}, y_{B3}, z_{B3})\}$

Объем требуемой памяти $\Pi_1 = 6 \times 4 \times 3 \times P_v$,


P_v – разрядность чисел координат.

Способы организации хранения информации об объекте.

Координаты 8-ми вершин сохраняются без повторов.

Вершины пронумерованы.

Грани даются в виде списка индексов вершин.

	0	1	2		7											
Вершины	x															
	y															
	z															
Грани																
	0	1	2	3	4	5	6	7					0	4	7	3

$$P_2 = 8 \times 3 \times P_v + 6 \times 4 \times P_{инд},$$

P_v - разрядность чисел координат,

$P_{инд}$ – разрядность индексов.

Способы организации хранения информации об объекте.

Линейно-узловая модель

Иерархия: вершины – ребра – грани.

$$P_3 = 8 \times 3 \times R_v + 12 \times 2 \times R_{\text{инд.в.}} + 6 \times 4 \times R_{\text{инд.р.}}$$

R_v - разрядность чисел координат,

$R_{\text{инд. в.}}$, $R_{\text{инд. р.}}$ – разрядность индексов.

Необходимо определиться с типами данных.

Пусть для координат используется тип float (4 б),

а для индексов – long (4 б).

$$P_1 = 6 \times 4 \times 3 \times 4 = 288$$

$$P_2 = 8 \times 3 \times 4 + 6 \times 4 \times 4 = 192$$

$$P_3 = 8 \times 3 \times 4 + 12 \times 2 \times 4 + 6 \times 4 \times 4 = 288$$

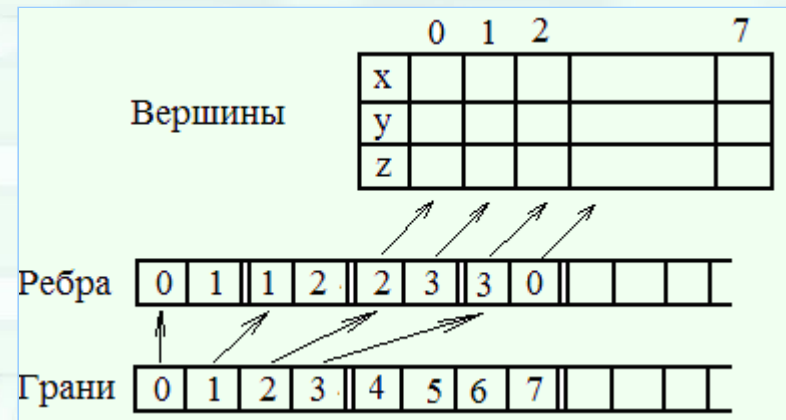
Или для координат - double (8 б), для индексов - long (4 б)

$$P_1 = 6 \times 4 \times 3 \times 8 = 576$$

$$P_1 = 8 \times 3 \times 8 + 6 \times 4 \times 4 = 288$$

$$P_1 = 8 \times 3 \times 8 + 12 \times 2 \times 4 + 6 \times 4 \times 4 = 384$$

Когда разрядность координат больше, чем для индексов, то ощутимо преимущество второго и третьего вариантов.

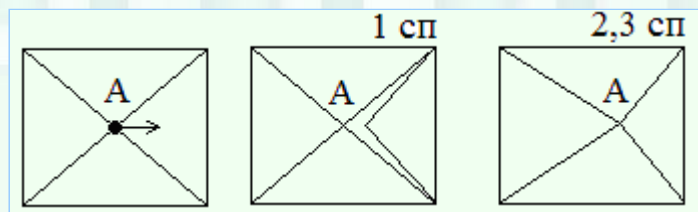


Скорость вывода.

Топологический аспект.

Если необходимо рисовать линии контуров и точки заполнения, то 1 и 2 близки по скорости. Но во 2-ом нужно выбрать индекс вершины, это замедляет скорость. Но для третьего можно не рисовать повторно ребра.

Топологический аспект



Достоинства векторной полигональной модели:

- Удобство масштабирования (векторная модель лучше растровой).
- Небольшой объем данных для простых поверхностей.
- Преобразование только координат вершин при трансформации.
- Аппаратная поддержка многих операций.

Недостатки:

- Сложные алгоритмы визуализации для реалистичных изображений; сложные алгоритмы топологических операций (разрезы).
- Аппроксимация плоскими гранями приводит к погрешности моделирования, невозможность отобразить сложные поверхности.

Воксельная модель

Это трехмерный растр.
Voxel – volume element.

Каждый воксел имеет атрибут
(цвет и прозрачность).

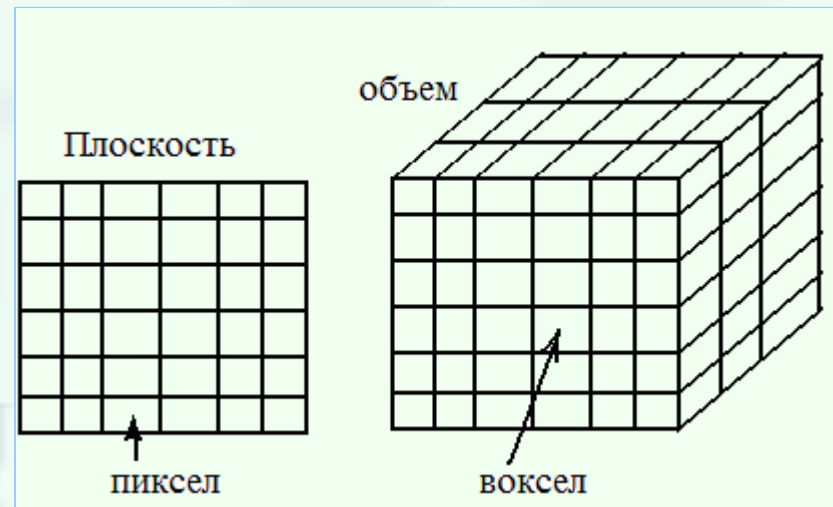
Удачно используются в медицине (компьютерная томография), в географии, сейсмологии, для отображения на реальных трехмерных устройствах.

Достоинства:

- Достаточно просто описать сложные поверхности, простая процедура отображения.
- Простое выполнение топологических операций.

Недостатки:

- Большое количество информации для представления объемных данных.
- Ограничение разрешающей способности.
- Проблемы масштабирования.



Равномерная сетка

Каждому узлу сетки индексом (i,j) приписывается индекс Z_{ij} .
Индексы (i, j) отвечают определенным значениям координат x, y .
Расстояние между узлами одинаково: dx по оси Ox , dy по оси Oy .

Модель: двумерный массив значений высоты.
Использовать для однозначной функции $z=f(x,y)$.

Достоинства:

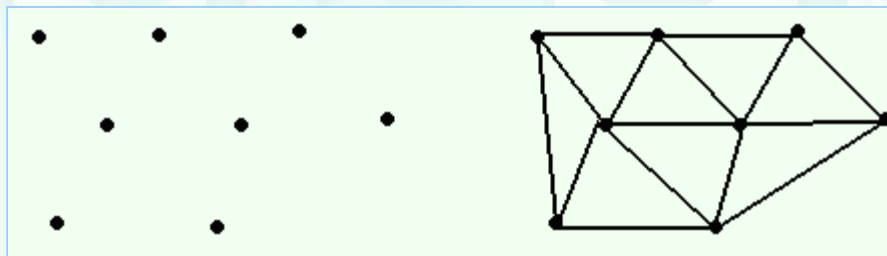
- простота описания поверхности,
- простота интерполяции для промежуточных точек.

Недостатки:

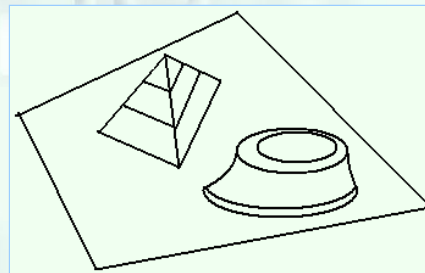
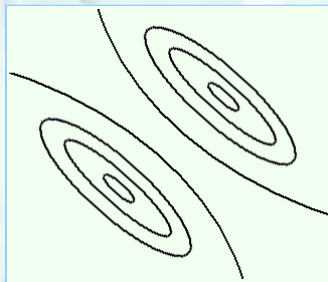
- нельзя моделировать неоднозначные функции поверхности,
- большое количество узлов для сложных поверхностей,
- больше затрат для отображения простых моделей (например, полигональных).

Неравномерная сетка. Изолиния.

Поверхность задается множеством точек $\{(x_0, y_0, z_0), (x_1, y_1, z_1), \dots\}$, принадлежащих поверхности. Как определить точки поверхности, несовпадающие с опорными?
Специальные методы: триангуляция.



Другой способ описания поверхности – изолиний высоты. Линии указывают точки одинаковой высоты. Используется в картографии.



Визуализация объемных изображений

Способы визуализации по характеру изображений и по степени сложности алгоритмов делятся:

2. Каркасная модель, отображаются только ребра.
3. Показ поверхностей в виде многогранников с плоскими гранями с удалением невидимых точек.
4. Добавляем гладкое закрашивание, имитация отражения света, затемнение, прозрачность, текстуры.

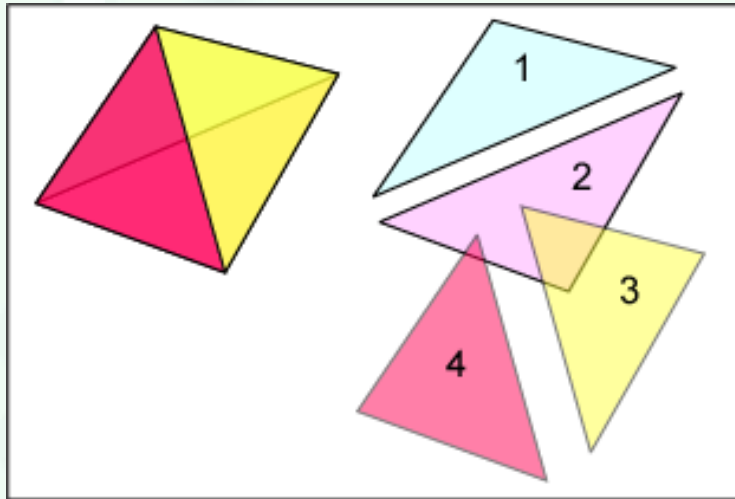
Каркасная визуализация

Отображаются все ребра как близкие, так и дальние. Обычно используется для редактирования объектов. Для отображения необходимо преобразовать координаты всех вершин в выбранной проекции.

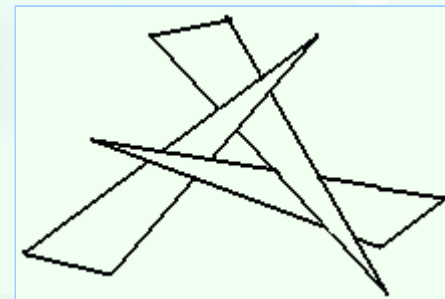
Визуализация с удалением невидимых линий

Методы удаления невидимых линий:

2. Сортировка граней по глубине.
 3. Метод плавающего горизонта.
 4. Метод z-буфера.
- 1) Рисование граней в порядке от самых дальних к самым близким.



Но иногда нельзя четко различить, какая грань ближе. Например, в каком порядке рисовать грани?



Визуализация с удалением невидимых линий

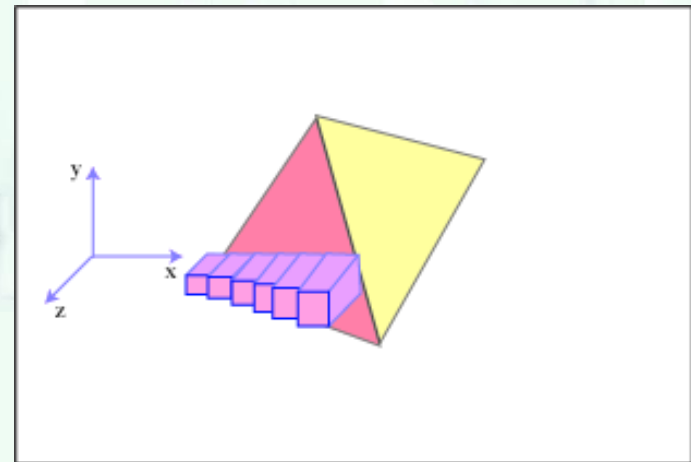
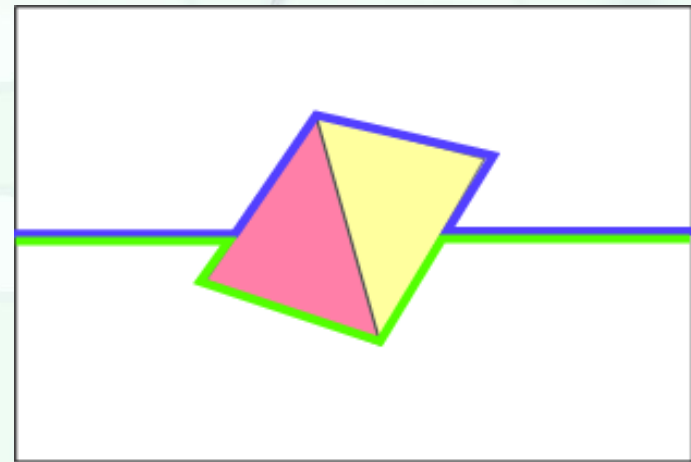
2) Вывод от самых ближних до самых дальних.

На каждом шаге образуются две ломанные – **верхний** и **нижний**

горизонт. Новая грань выводится лишь в том случае, если

выше верхнего, или ниже нижнего горизонта.

3) Используется дополнительный массив, который хранит координату z для каждого пикселя раstra.



Закрашивание поверхностей

Модели отражения

Как можно определить цвет пикселей изображения поверхности согласно интенсивности отраженного света при учете взаимного расположения поверхности, источника света и наблюдателя.

Зеркальное отражение



Падающий луч, попадая на слегка шероховатую поверхность реального зеркала, порождает не один луч, а несколько лучей, рассеиваемых по различным направлениям.

Эмпирическая формула распределения Фонга:
$$I_s = I \cdot K_s \cos^P \alpha$$

где I – интенсивность излучения источника,

K_s – коэффициент пропорциональности,

α – угол отклонения от линии идеального отраженного луча.

$1 < P < 200$ зависит от полировки поверхности.

Закрашивание поверхностей

Диффузное отражение

Присуще матовым поверхностям.

Закон Ламберта: интенсивность отраженного света пропорциональна косинусу угла между нормалью и углом на источник света Q .

I – интенсивность источника света, $I_D = I \cdot K_D \cdot \cos Q$

K_D – коэффициент, свойство материала.

Для реальных изображений учитывают оба типа отражений

$$I_{\text{тод}} = I \cdot (K_D \cdot \cos Q + K_S \cdot \cos^P \alpha)$$

где K_D , K_S – свойства материала. Для некоторых Q и α $I_{\text{отр}} = 0$,

но реально это не так, значит $I_{\text{тод}} = I_a K_a + I(K_D \cdot \cos Q + K_S \cdot \cos^P \alpha)$

где I_a – интенсивность рассеянного света, K_a – константа.

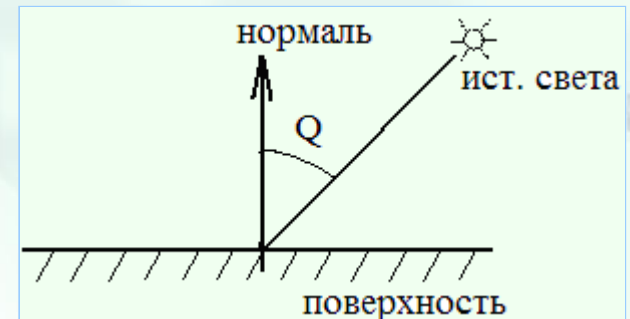
Энергия точечного источника уменьшается пропорционально R^2 .

Но на практике используют следующее: $I_{\text{тод}} = I_a K_a + \frac{I}{R + k} (K_D \cdot \cos Q + K_S \cdot \cos^P \alpha)$

R – расстояние от центра проекции до поверхности, K – константа.

Как определить цвет закрашивания точек объекта?

Для RGB K_a , K_D будут различны для разных компонент, K_S – будет одинаков для всех компонент. Цвет источника выражается в разных компонентах I .



Вычисление нормалей и углов

Для поверхности:

$$x = x(s, t)$$

$$y = y(s, t)$$

$$z = z(s, t)$$

координаты вектора нормали:

$$x_N = \frac{\partial y}{\partial s} \cdot \frac{\partial z}{\partial t} - \frac{\partial y}{\partial t} \cdot \frac{\partial z}{\partial s}$$

$$y_N = \frac{\partial z}{\partial s} \cdot \frac{\partial x}{\partial t} - \frac{\partial z}{\partial t} \cdot \frac{\partial x}{\partial s}$$

$$z_N = \frac{\partial x}{\partial s} \cdot \frac{\partial y}{\partial t} - \frac{\partial x}{\partial t} \cdot \frac{\partial y}{\partial s}$$

В случае векторно-полигональной модели используем методы векторной алгебры.

$$x'_i = x_i - x_1$$

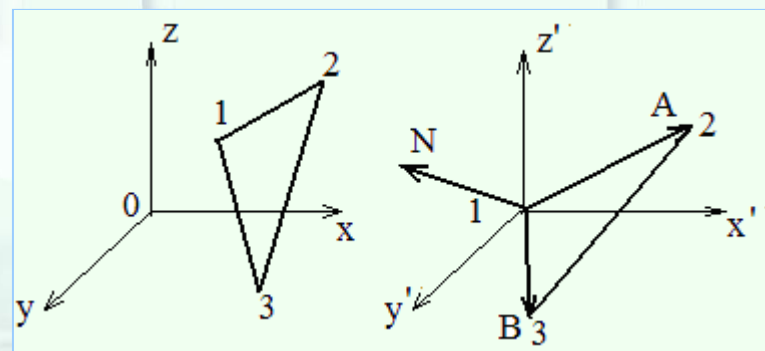
$$y'_i = y_i - y_1$$

$$z'_i = z_i - z_1$$

$$x'_N = (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1)$$

$$y'_N = (z_2 - z_1)(x_3 - x_1) - (z_3 - z_1)(x_2 - x_1)$$

$$z'_N = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)$$



$$N = A \times B$$

Направление можно поменять $-N$.

k i s d h . n s n d f d f s / s d f . d f d / : ^ % ^ [] L H H S f h s d b ^ G B L

$$\cos \theta = \frac{Z_N}{|N|} = \frac{Z_N}{\sqrt{x_N^2 + y_N^2 + z_N^2}}$$
$$\Rightarrow \cos \theta = \frac{x_S x_N + y_S y_N + z_S z_N}{|S| \cdot |N|}$$

sdfsdf%^&nH%^mm^&:n&*\$BOPAHOGNGH<J<.hkcv.hzc

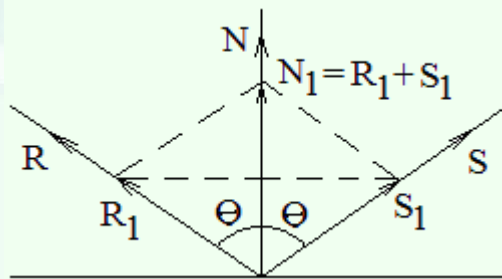
Зеркальное отражение

S – радиус-вектор, направленный на источник света,

N – вектор нормали.

Найти $\cos \alpha$ угла между отраженным лучом и направлением на камеру.

Пусть **R** - радиус-вектор отраженного луча.



R₁, S₁, N₁ – единичные векторы.

$$R_1 + S_1 = N_1$$

$$|N_1| = 2 \cos \theta$$

$$\Rightarrow N_1 = N \cdot 2 \cos \theta$$

$$R_1 + S_1 = N_1 \cdot 2 \cos \theta$$

$$\Rightarrow R_1 = N_1 \cdot 2 \cos \theta - S_1 = \frac{N}{|N|} 2 \cos \theta - \frac{S}{|S|}$$

но $\cos \theta = \frac{N \cdot S}{|N||S|}$ **подставим**

$$\Rightarrow R_1 = N \cdot 2 \cdot \frac{N \cdot S}{|N|^2 |S|} - \frac{S}{|S|}$$

Считаем, что $R = |S| \cdot R_1$

$$\Rightarrow R = N \cdot 2 \cdot \frac{N \cdot S}{|N|^2} - S$$

$$x_R = 2x_N \frac{x_N x_S + y_N y_S + z_N z_S}{x_N^2 + y_N^2 + z_N^2} - x_S$$

Y_R, Z_R находятся аналогично.

Осталось найти косинус угла между отраженным лучом и направлением на камеру **K**.

$$\cos \alpha = \frac{K \cdot R}{|K||R|} = \frac{x_K x_R + y_K y_R + z_K z_R}{\sqrt{x_K^2 + y_K^2 + z_K^2} \cdot \sqrt{x_R^2 + y_R^2 + z_R^2}}$$

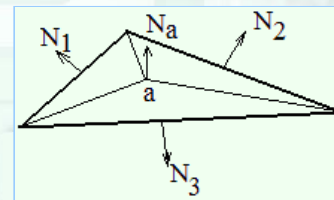
Очевидно лучше использовать единичные S, N, K (R тоже будет таким).

Метод Гуро

Создание иллюзии гладкой криволинейной поверхности, описанной в виде многогранников.

Суть: закрашивание плоской грани не одним цветом, а плавно изменяющимися оттенками, вычисляемыми путем интерполяции цветов примыкающих граней.

- Вычисляем нормали каждой грани.
- Определяем нормали в вершинах.



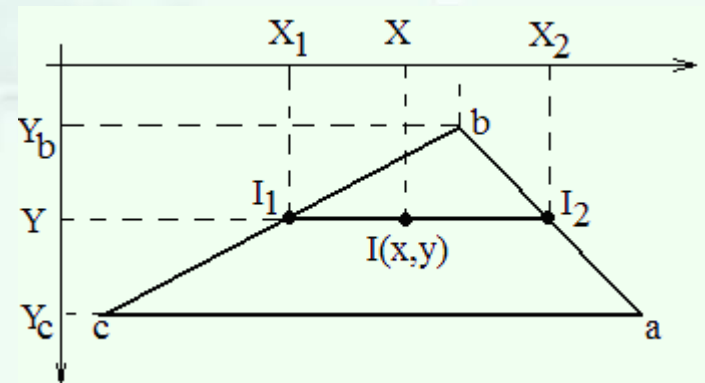
$$N_a = \frac{N_1 + N_2 + N_3}{3}$$

3. На основе нормалей в вершинах вычисляется интенсивность в вершинах согласно выбранной модели отражения света.
4. Закрашиваем полигон цветом соответствующей линейной интерполяции значений интенсивности в вершинах.

Заполнение контура грани.

$$\frac{I - I_1}{x - x_1} = \frac{I_2 - I_1}{x_2 - x_1} \Rightarrow I_1 = I_b + (I_c - I_b) \frac{y - y_b}{y_c - y_b}$$

$$I = I_1 + (I_2 - I_1) \frac{x - x_1}{x_2 - x_1} \quad I_2 = I_b + (I_a - I_b) \frac{y - y_b}{y_a - y_b}$$



Метод Фонга

Отличается тем, что интерполируется не цвет, а вектор нормали.

2. Определяем нормали к граням.
3. Определяем нормали в вершинах. В каждой точке грани определяем вектор нормали интерполированием.
4. По вектору нормали определяем цвет точки грани в соответствии с выбранной моделью отражения света.

$$N'_1 = \begin{pmatrix} X_{N1} \\ Y_{N1} \\ Z_{N1} \end{pmatrix} = \begin{pmatrix} X_{Nb} + (X_{Nc} - X_{Nb})(Y - Y_b)/(Y_c - Y_b) \\ Y_{Nb} + (Y_{Nc} - Y_{Nb})(Y - Y_b)/(Y_c - Y_b) \\ Z_{Nb} + (Z_{Nc} - Z_{Nb})(Y - Y_b)/(Y_c - Y_b) \end{pmatrix}$$

$$N'_2 = \begin{pmatrix} X_{N2} \\ Y_{N2} \\ Z_{N2} \end{pmatrix} = \begin{pmatrix} X_{Nb} + (X_{Na} - X_{Nb})(Y - Y_b)/(Y_a - Y_b) \\ Y_{Nb} + (Y_{Na} - Y_{Nb})(Y - Y_b)/(Y_a - Y_b) \\ Z_{Nb} + (Z_{Na} - Z_{Nb})(Y - Y_b)/(Y_a - Y_b) \end{pmatrix}$$

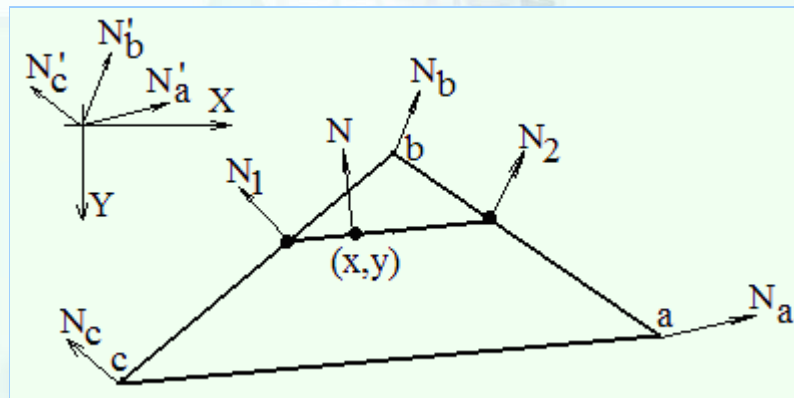
где

$$(X_{Na}, Y_{Na}, Z_{Na}) = N'_a$$

$$(X_{Nb}, Y_{Nb}, Z_{Nb}) = N'_b$$

$$(X_{Nc}, Y_{Nc}, Z_{Nc}) = N'_c$$

$$N' = \begin{pmatrix} X_N \\ Y_N \\ Z_N \end{pmatrix} = \begin{pmatrix} X_{N1} + (X_{N2} - X_{N1})(X - X_1)/(X_2 - X_1) \\ Y_{N1} + (Y_{N2} - Y_{N1})(X - X_1)/(X_2 - X_1) \\ Z_{N1} + (Z_{N2} - Z_{N1})(X - X_1)/(X_2 - X_1) \end{pmatrix}$$



Метод Фонга сложнее, чем метод Гуро, но дает более высокий результат. Особенно хорошо видно различие на зеркальных поверхностях.

Преломление света

Законы преломления света следует учитывать при построении изображений прозрачных объектов.

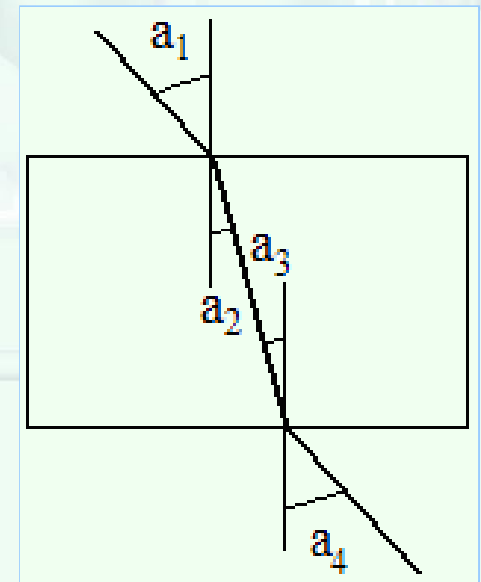
Модель идеального преломления.

Луч отклоняется на границе двух сред, причем падающий, преломленный луч и нормаль лежат в одной плоскости. Для углов α_1 и α_2 известен закон Снелиуса: $n_1 \sin \alpha_1 = n_2 \sin \alpha_2$

Где n_1 и n_2 – абсолютные показатели преломления соответствующих сред.

На примере $\alpha_1 = \alpha_4$, $\alpha_2 = \alpha_3$.

Следовательно, после прохождения сквозь стекло луч параллельно смещается. Смещение зависит от толщины стекла и соотношения показателей преломления сред.



Треугольная призма

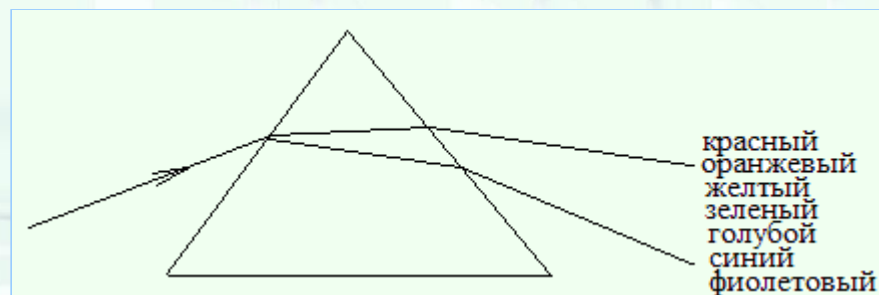
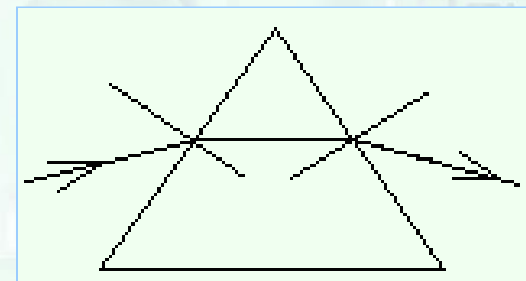
Принято считать, что для вакуума абсолютный показатель преломления = 1, для воздуха: 1,0029, для воды: 1,33, для стекла разных сортов: 1,52 (легкий крон), 1,65 (тяжелый крон).

На практике обычно используют отношение показателей преломления двух сред (n_1 , n_2) относительно показателей преломления.

Еще один аспект преломления: зависимость луча от длины волны. Известен опыт Ньютона по разложению белого света треугольной призмой.

Чем меньше длина волны, тем больше отклоняется луч при преломлении. Фиолетовый ($\lambda=0,4$ мкм) луч отклоняется больше всего, красный ($\lambda=0,7$ мкм) – меньше всего.

Например, для стекла показатель преломления в видимом спектре меняется от 1.53 до 1.51. Кроме модели идеального преломления используют. Но реже, модель диффузного преломления. Согласно этой модели падающий луч преломляется во все стороны.



Трассировка лучей

Методы трассировки лучей считаются на сегодняшний момент наиболее мощными и универсальными методами создания реалистичных изображений.

Будем полагать, что окружающие нас объекты обладают свойствами:

- излучают;
- отражают и поглощают;
- пропускают сквозь себя.

Каждое из этих свойств можно описать набором параметров.



Метод прямой трассировки лучей заключается в следующем:

Каждый источник света излучает бесконечное множество лучей, они уходят в свободное пространство, попадают на другие объекты и глаз наблюдателя. Попавшие на другие объекты они отражаются, преломляются – это образуются вторичные лучи и так далее до тех пор, пока не попадет в глаз наблюдателя

Метод обратной трассировки лучей

Разработан в 80-х годах, основополагающие работы Уиттеда и Кея.

Согласно этому методу отслеживание ведется не от источника света, а от наблюдателя. Но и это не делает задачу решаемой, поэтому вводят следующие ограничения:

3. Среди всех типов объектов выделим некоторые, которые назовем источником света. Они излучают свет, но не отражают и не преломляют его.
4. Свойства отражающей поверхности описываются суммой двух компонент – диффузной и зеркальной.
5. Зеркальность описывается двумя составляющими. Первая (reflection). Вторая (specular).
6. Для прозрачных (transparent) объектов обычно не учитывается зависимость преломления света от длины волны. Иногда прозрачность моделируют вообще без преломления.
7. Для учета освещенности объектов светом, рассеиваемым другими объектами, вводится фоновая составляющая (ambient).
8. Для завершения трассировки вводят пороговое значение освещенности либо ограничение количества итераций.

Достоинства и недостатки метода обратной трассировки лучей

Положительные черты метода обратной трассировки лучей:

2. Универсальность метода, его применимость для синтеза изображений достаточно сложных сцен. Воплощает законы геометрической оптики. Просто реализует разнообразные проекции.
3. Даже учеч. варианты метода дают хорошие результаты. Первичные лучи дают удаление невидимых точек. Трассировка одного – двух вторичных лучей дает тени, зеркальность, прозрачность.
4. Все преобразования координат линейны, поэтому достаточно просто работать с текстурами.
5. Для одного пикселя можно трассировать несколько близко расположенных лучей, потом усреднять значения цвета (антиалиазинг).
6. Поскольку расчет отдельной точки изображения выполняется независимо от других точек, то это позволяет эффективно использовать параллельные вычисления.

Недостатки:

8. Проблемы с моделированием диффузного отражения и преломления.
9. Для каждой точки необходимо выполнять много вычислительных операций. Трассировка лучей относится к самым медленным алгоритмам синтеза изображения.