

Лекция 14

Трёхмерная графика

(с) КИС, ФРФИКТ, БГУ,
2011

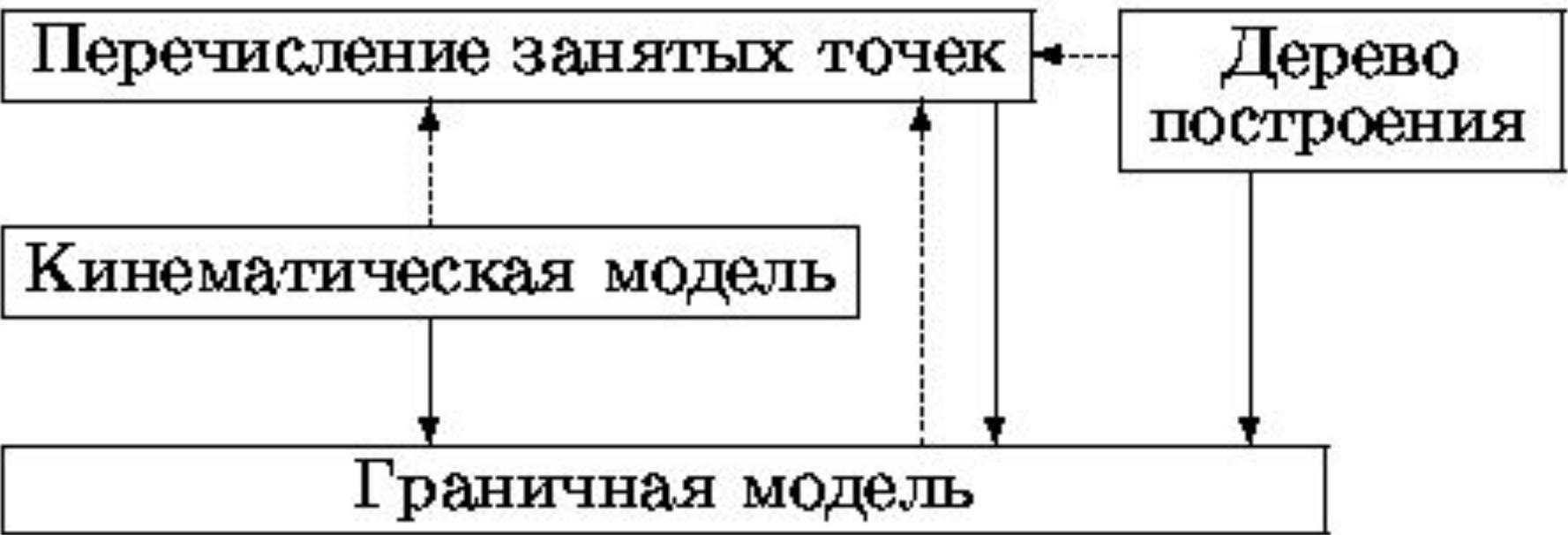
Ожидаемые вопросы:

Много 😊

Типы 3D моделей

Можно выделить два основных типа представлений 3D моделей:

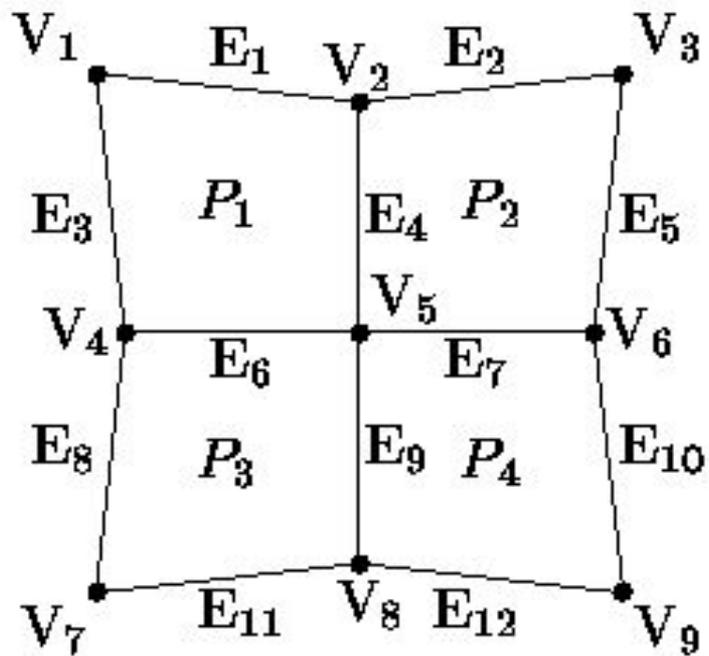
- *граничное*, когда в модели хранятся границы объекта, например, вершины, ребра, грани,
- в виде *дерева построения*, когда хранятся базовые объекты (призма, пирамида, цилиндр, конус и т.п.) из которых формировалось тело и использованные при этом операции; в узле дерева сохраняется операция формирования, а ветви представляют объекты.



Используются две основных разновидности способов представления поверхностей тела:

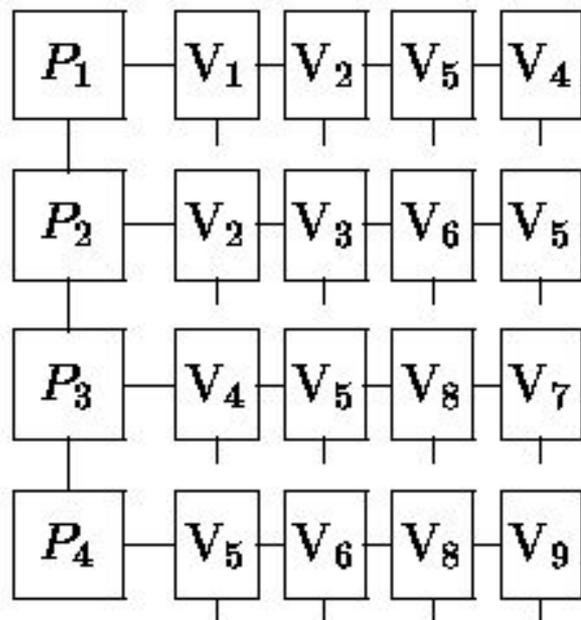
- представление в виде набора вершин, ребер и плоских многоугольников (полигональных сеток),
- представление с использованием параметрических бикубических площадок (кусков).

Полигональная сетка представляет собой набор вершин, ребер и плоских многоугольников. Вершины соединяются ребрами. Многоугольники рассматриваются либо как последовательность вершин или ребер. Можно предложить много способов внутреннего представления полигональных сеток.

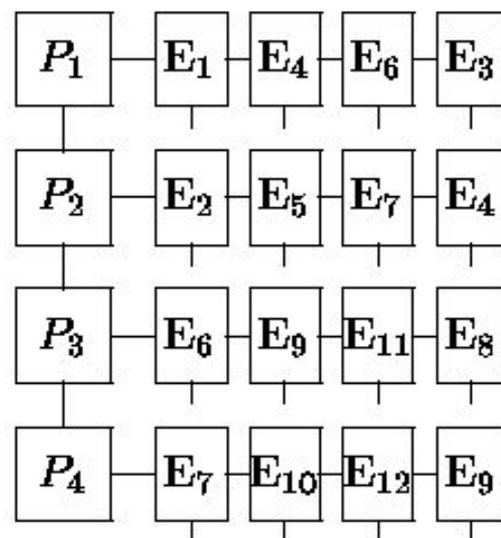


P_1	P_2	P_3	P_4
V_1	V_2	V_4	V_5
V_2	V_3	V_5	V_6
V_5	V_6	V_8	V_9
V_4	V_5	V_7	V_8

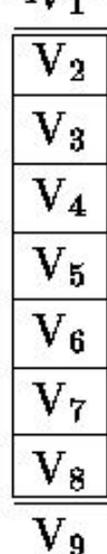
Списки вершин



Списки ребер



Список вершин



Классификация методов удаления невидимых частей

Методы удаления невидимых частей сцены можно классифицировать:

По выбору удаляемых частей:

удаление невидимых линий, ребер, поверхностей, объемов.

По порядку обработки элементов сцены:

удаление в произвольном порядке и в порядке, определяемом процессом визуализации.

По системе координат:

алгоритмы работающие в пространстве объектов, когда каждая из N граней объекта сравнивается с остальными $N-1$ гранями (объем вычислений растет как N^2),

алгоритмы работающие в пространстве изображения, когда для каждого пиксела изображения определяется какая из N граней объекта видна (при разрешении экрана $M \times M$ объем вычислений растет как $M^2 \times N$).

Алгоритмы удаления линий

Применение - векторные устройства. Могут применяться и в растровых для ускорения процесса визуализации, но при этом не используется основное ценное качество растрового дисплея - возможность закраски поверхностей.

Алгоритм удаления поверхностей с Z-буфером

Тогда при C не равном нулю

Если известно уравнение плоскости, в которой лежит обрабатываемый многоугольник, то можно обойтись без хранения Z-координат вершин. Пусть уравнение плоскости имеет вид:

$$A \cdot x + B \cdot y + C \cdot z + D = 0$$

$$z = -(A \cdot x + B \cdot y + D) / C$$

Найдем приращение Z-координаты пиксела при шаге по X на dx , помня, что Y очередной обрабатываемой строки - константа.

$$dz = -(A \cdot (x+dx) + D)/C + (A \cdot x + D)/C = -A \cdot dx/C \text{ но } dx = 1, \text{ поэтому}$$

$$dz = -A/C.$$

Общая схема

- Инициализировать кадровый и Z-буфера. Кадровый буфер закрашивается фоном. Z-буфер закрашивается минимальным значением Z.
- Выполнить преобразование каждого многоугольника сцены в растровую форму. При этом для каждого пиксела вычисляется его глубина z . Если вычисленная глубина больше, чем глубина, уже имеющаяся в Z-буфере, то занести в буфера атрибуты пиксела и его глубину, иначе никаких занесений не выполнять.
- Выполнить, если это было предусмотрено, усреднение изображения с понижением разрешения.

Построчный алгоритм с Z-буфером

Рассмотрим теперь алгоритм с Z-буфером размером в одну строку, который представляет собой обобщение алгоритма построчной заливки многоугольника. Модификация должна учесть то, что для каждой строки сканирования теперь может обрабатываться не один многоугольник.

Общая схема такого алгоритма следующая:

- Подготовка данных.
- Для каждого многоугольника определить максимальную Y-координату.
- Занести многоугольник в группу многоугольников, соответствующую данной Y-координате.
- Собственно заливка.

Построчный алгоритм

Уоткинса

В алгоритмах построчного сканирования результирующее изображение генерируется построчно причем, используется связность соседних растровых строк изображения. Отличие состоит в том, что учитываются все, а не один многоугольник.

Алгоритм работает в пространстве изображения с окном высотой в одну строку и шириной в экран, тем самым трехмерная задача сводится к двумерной.

Последовательность шагов алгоритма:

- построение списка ребер,
- построение списка многоугольников,
- построение списка активных ребер - создается таблица ребер, включающая все негоризонтальные ребра многоугольников, причем элементы таблицы по значению Y-координаты отсортированы по группам.

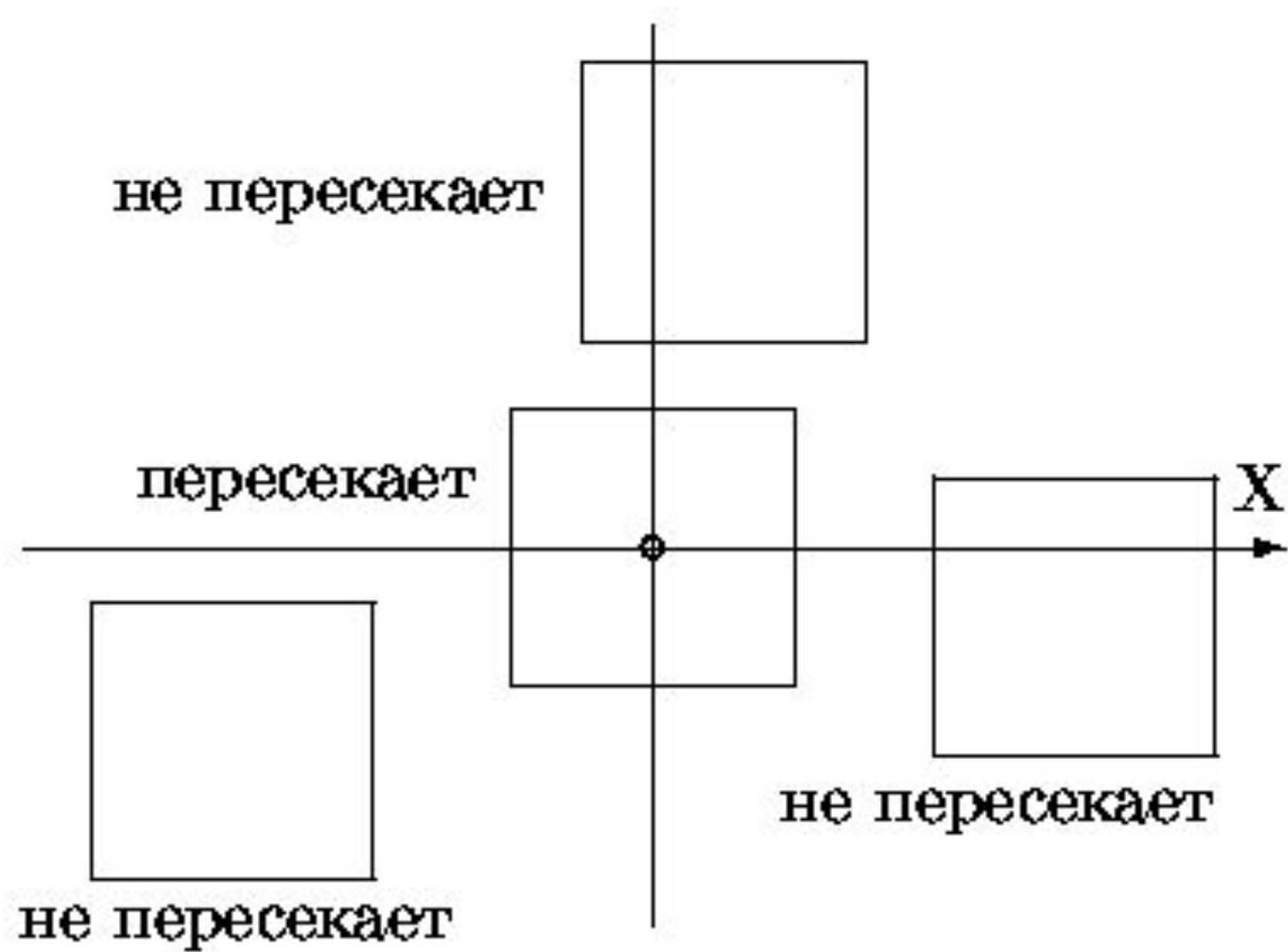
Алгоритм трассировки лучей

Удаление невидимых (скрытых) поверхностей в алгоритме трассировки лучей выполняется следующим образом:

- сцена преобразуется в пространство изображения,
- из точки наблюдения в каждый пиксел экрана проводится луч и определяется какие именно объекты сцены пересекаются с лучом,
- вычисляются и упорядочиваются по Z координаты точек пересечения объектов с лучом. В простейшем случае для непрозрачных поверхностей без отражений и преломлений видимой точкой будет точка с максимальным значением Z -координаты. Для более сложных случаев требуется сортировка точек пересечения вдоль луча.

Ясно, что наиболее важная часть алгоритма - процедура определения пересечения, которая в принципе выполняется $R_x \times R_y \times N$ раз (здесь R_x, R_y - разрешение дисплея по X и Y, соответственно, а N - количество многоугольников в сцене).

Очевидно, что повышение эффективности может достигаться сокращением времени вычисления пересечений и избавлением от ненужных вычислений. Последнее обеспечивается использованием геометрически простой оболочки, объемлющей объект - если луч не пересекает оболочку, то не нужно вычислять пересечения с ним многоугольников, составляющих исследуемый объект.



При использовании сферической оболочки для определения пересечения луча со сферой достаточно сосчитать расстояние от луча до центра сферы. Если оно больше радиуса, то пересечения нет. Параметрическое уравнение луча, проходящего через две точки $P1(x1, y1, z1)$ и $P2(x2, y2, z2)$, имеет вид:

$$P(t) = P1 + (P2 - P1) \times t$$

Минимальное расстояние от точки центра сферы $P_0(x_0, y_0, z_0)$ до луча равно:

$$d^2 = (x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2$$

Этому соответствует значение t :

$$t = - \frac{(x_2-x_1) \cdot (x_1-x_0) + (y_2-y_1) \cdot (y_1-y_0) + (z_2-z_1) \cdot (z_1-z_0)}{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2}$$

Если $d^2 > R^2$, то луч не пересекает объекты, заключенные в оболочку.

РЕАЛИСТИЧНОЕ ПРЕДСТАВЛЕНИЕ СЦЕН

Где нужен реализм:

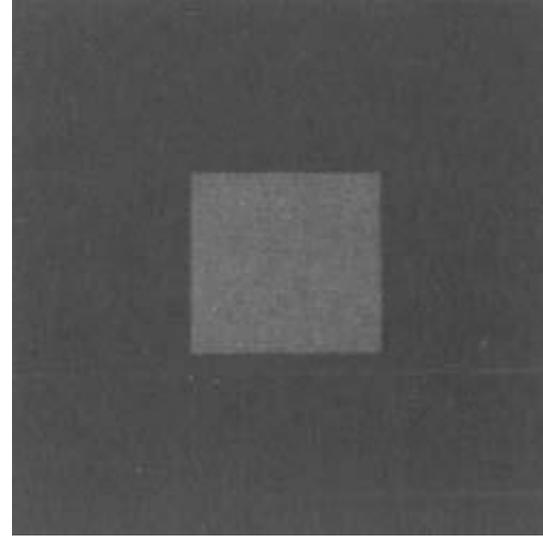
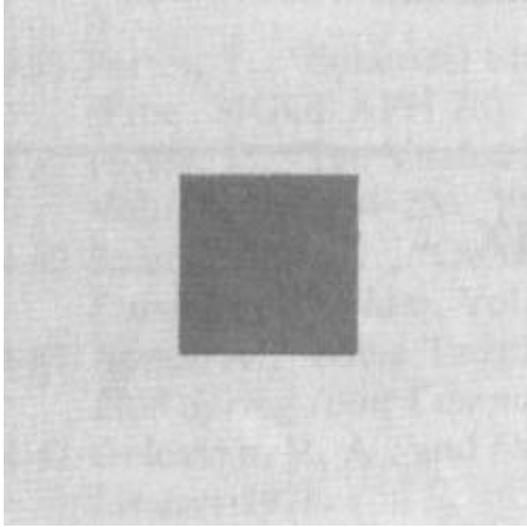
- в конструировании,
- в архитектуре,
- в биологии и медицине,
- в науке (компьютерное моделирование),
- в масс-медиа,
- в тренажерах, играх.

Основные направления:

- синтез реалистичных изображений,
- реалистическое оживление синтезированных объектов.

С точки зрения приложений в науке и промышленности наиболее важно первое направление. Ключевая проблема - реалистическое представление освещенности:

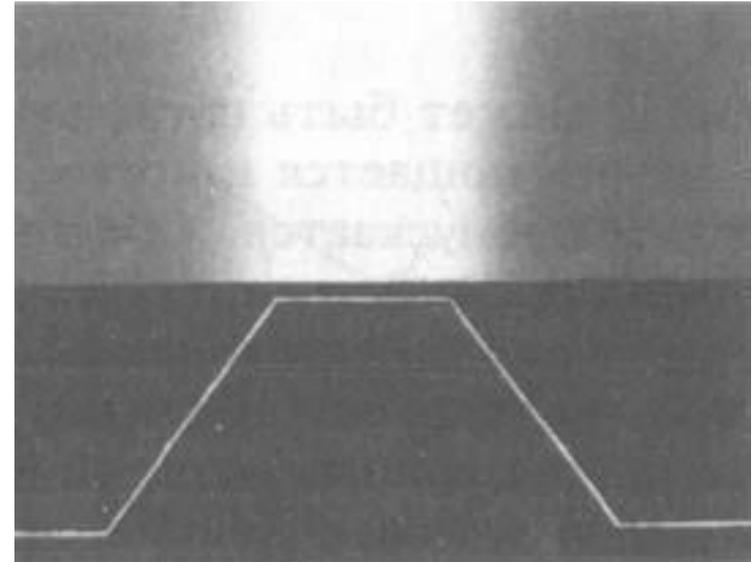
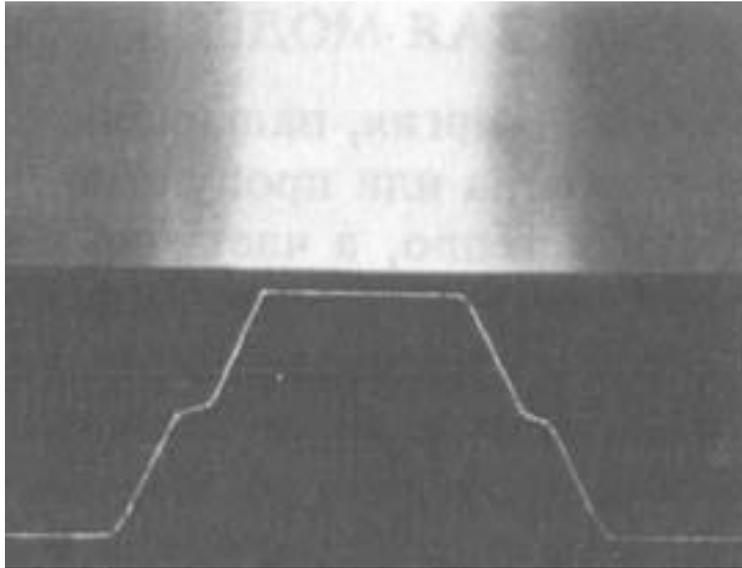
- модели освещения, прозрачность, тени, фактура,
- глобальная модель освещения с трассировкой лучей,
- излучательность.



a

b

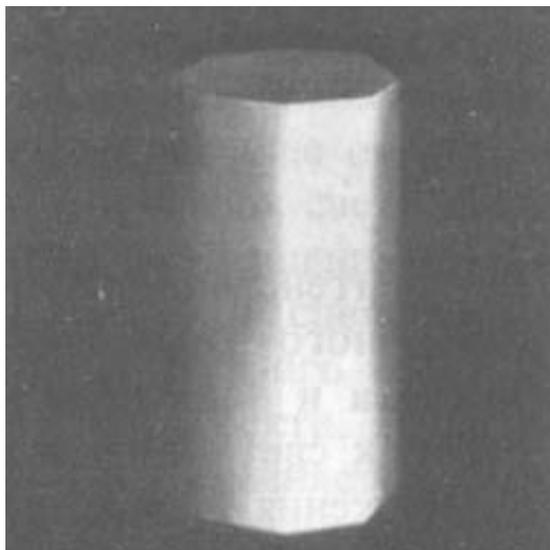
Рисунок Одновременный контраст.



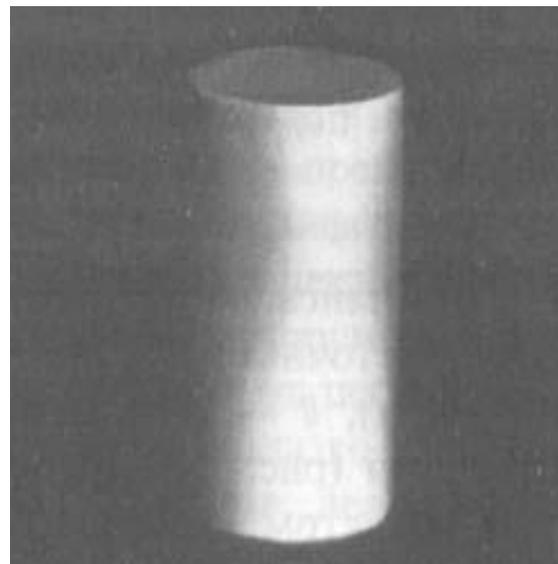
a

b

Рисунок Эффект полос Маха: (a) кусочно-линейная функция интенсивности, (b) функция интенсивности с непрерывной первой производной.



а



б

Рисунок Эффект полос Маха на поверхностях, образованных плоскими полигональными гранями: (а) модель с 8 гранями, (б) модель с 32 гранями.

Простая модель освещенности

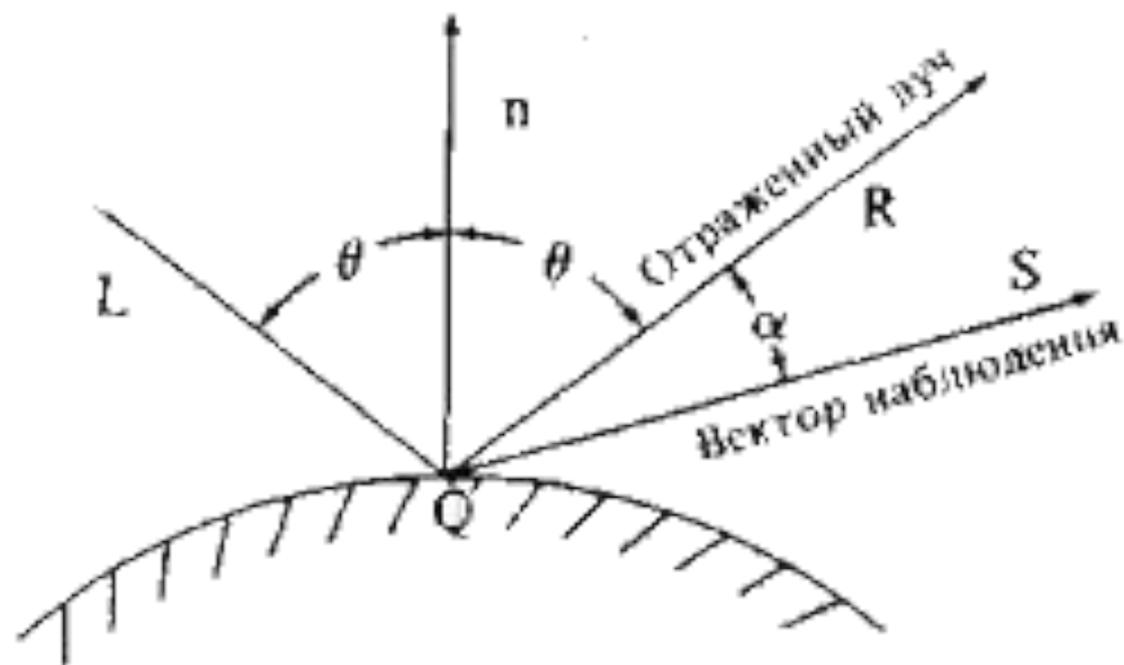
Свет точечного источника отражается от идеального рассеивателя по закону косинусов Ламберта: интенсивность отраженного света пропорциональна косинусу угла между направлением света и нормалью к поверхности, т. е.

$$I = I_t k_d \cos \theta \quad 0 \leq \theta \leq \pi / 2$$

Поверхность предметов, изображенных при помощи простой модели освещения с ламбертовым диффузным отражением, выглядит блеклой и матовой. Предполагается, что источник точечный, поэтому объекты, на которые не падает прямой свет, кажутся черными. Однако на объекты реальных сцен падает еще и рассеянный свет, отраженный от окружающей обстановки, например от стен комнаты. Рассеянному свету соответствует распределенный источник. Поскольку для расчета таких источников требуются большие вычислительные затраты, в машинной графике они заменяются на коэффициент рассеяния - константу, которая входит в формулу в линейной комбинации с членом Ламберта:

$$I = I_a k_a + I_t k_a \cos \theta \quad 0 \leq \theta \leq \pi / 2$$

$$I = I_2 K_2 + \frac{I_1 K_2 \cos \theta}{d + K}$$



A

Механизм диффузного и зеркального отражения света

Диффузное отражение света точечного источника от идеального рассеивателя определяется по закону Ламберта, согласно которому падающий свет рассеивается во все стороны с одинаковой интенсивностью. В этом случае освещенность точки пропорциональна доле ее площади, видимой от источника.

$I_r = I_p \cdot \rho_d \cdot \cos(\theta)$, где I_r - интенсивность отраженного света, I_p - интенсивность точечного источника, ρ_d - коэффициент диффузного отражения, зависящий от материала поверхности и длины волны, θ - угол между направлением света и нормалью к поверхности.

В реальных сценах, кроме света от точечных источников, присутствует и рассеянный свет, который упрощенно учитывается с помощью коэффициента рассеяния:

$I = I_r \cdot P_r + I_p \cdot P_d \cdot \cos(f)$, где I_r - интенсивность рассеянного света, $0 \leq P_r \leq 1$ - коэффициент диффузного отражения рассеянного света.

Субъективно достаточно реалистичный учет расстояния от центра проекции до объекта обеспечивается линейным затуханием:

$$I = I_r \cdot P_r + I_p \cdot P_d \cdot \cos(f) \cdot d + K$$

, где d - расстояние от центра проекции до объекта, а K - произвольная константа.

Свет, отраженный от идеального зеркала, виден только если угол между направлениями наблюдения и отражения равен нулю. Для неидеальных отражающих поверхностей используется модель Фонга:

$I_s = I_p \cdot W(l, f) \cdot \cos^n(a)$, где $W(l, f)$ - кривая отражения, зависящая от длины волны l света источника и угла падения f , $0 \leq a \leq \pi/2$ - угол между направлениями наблюдения и отражения, $1 \leq n \leq 200$ - показатель степени, определяющий убывание интенсивности при изменении угла.

Часто $W(l, f)$ заменяется константой K_s , такой чтобы полученная картина была субъективно приемлема.

Суммарная модель освещения имеет вид:

$$I = I_r \cdot P_r + I_p d + K(P_d \cdot \cos(f) + W(l, f) \cdot \cos^n(a)).$$

Или при использовании вместо $W(l, f)$ константы K_s :

$I = I_r \cdot P_r + I_p d + K(P_d \cdot \cos(f) + K_s \cdot \cos^n(a))$. Если использовать нормированные вектора направлений падения \mathbf{L} , нормали \mathbf{N} , отражения \mathbf{R} и наблюдения \mathbf{V} , то модель освещения для одного источника принимает вид:

$$I = I_r \cdot P_r + I_p d + K(P_d \cdot \mathbf{L} \cdot \mathbf{N} + K_s \cdot (\mathbf{R} \cdot \mathbf{V})^n).$$

Модели закраски

Существует три основных способа закраски многоугольников: однотонная закраска, закраска с интерполяцией интенсивности и закраска с интерполяцией векторов нормали.

При однотонной закраске предполагается, что и источник света и наблюдатель находятся в бесконечности, поэтому произведения $L \cdot N$ и $R \cdot V$ постоянны. На изображении могут быть хорошо заметны резкие перепады интенсивности между различно покрашенными многоугольниками. Если многоугольники представляют собой результат аппроксимации криволинейной поверхности, то изображение недостаточно реалистично.

В методе закраски с интерполяцией интенсивности (метод Гуро) нормали в вершинах многоугольников вычисляются как результат усреднения нормалей ко всем полигональным граням, которым принадлежит данная вершина. Используя значения нормалей, вычисляют интенсивности в вершинах по той или иной модели освещения. Эти значения затем используются для билинейной интерполяции: для данной строки сканирования вначале находят значения интенсивностей на ребрах, а затем линейно интерполируют между ними при закраске вдоль строки.

В методе закраски с интерполяцией нормали (метод Фонга) значение нормали вдоль строки интерполируется между значениями нормалей на ребрах для данной строки. Значения нормалей на ребрах получается как результат интерполирования между вершинами. Значения же нормалей в вершинах являются результатом усреднения, как и выше рассмотренном методе. Значение нормали для каждого из пикселей строки используется для вычислений по той или иной модели освещения.

Прозрачность

Суммарная закрашка определяется следующим образом:

$I = k \cdot I_b + (1-k) \cdot I_d$, где $0 \leq k \leq 1$ - характеризует прозрачность ближнего многоугольника. Если $k = 1$, то он непрозрачен. Если же $k = 0$, то ближний многоугольник полностью прозрачен; I_b - интенсивность для пиксела ближнего многоугольника, I_d - дальнего.

Тени

Простой способ определения объектов, попавших в тень и, следовательно, неосвещенных, аналогичен алгоритму удаления невидимых поверхностей: те объекты, которые невидимы из источника освещения, но видимы из точки зрения находятся в тени. На первом шаге в алгоритме с учетом тени определяются все многоугольники, видимые из точки освещения. Затем выполняется удаление поверхностей невидимых из точки зрения. При выполнении закрашки многоугольника проверяется не закрыт ли он многоугольником, видимым из источника освещения. Если да, то в модели освещения учитываются (если надо) все три компоненты - диффузное и зеркальное отражения и рассеянный свет. Если же перекрытия нет, то закрашиваемый многоугольник находится в тени и надо учитывать только рассеянный свет.

Фактура

Решение в лоб - представление в виде соответствующего (очень большого) количества многоугольников мало приемлемо. Более практичное решение - "натягивание" массива узора, полученного в результате оцифровки изображения реальной поверхности на раскрашиваемую. При этом значения из массива узора используются для масштабирования диффузной компоненты в модели освещения.

Для устранения лестничного эффекта должны учитываться все элементы узора, затрагивающие обрабатываемый пиксел изображения.

Такой метод влияет на раскраску поверхности, но оставляет ее гладкой. Неровности могут моделироваться возмущениями нормали поверхности. Другой способ, используемый при

Трассировка лучей

Метод трассировки лучей используется не только для удаления невидимых частей, но, в основном, для получения высокореалистичных изображений с учетом отражений и преломлений света.

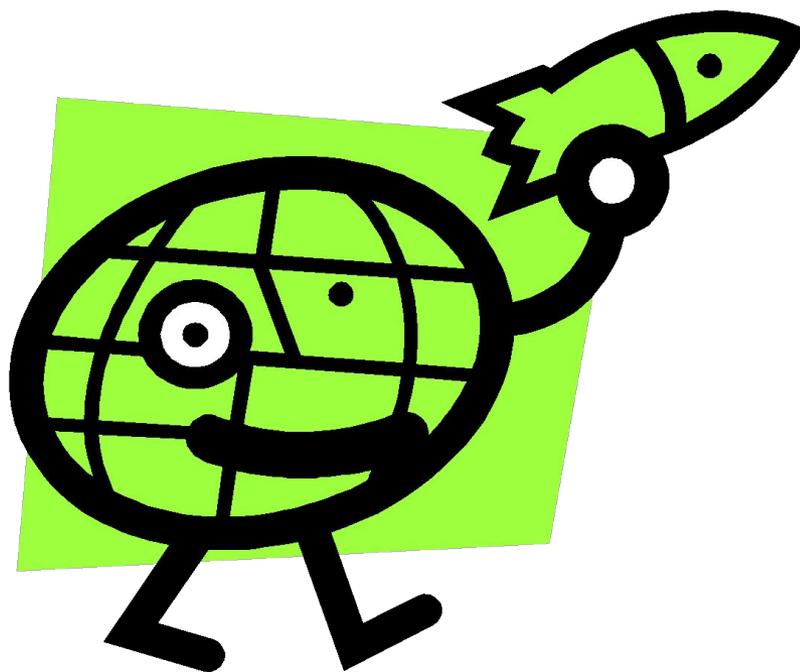
Тест

1. Что такое компьютерная графика?
2. Какие цветовые модели Вам известны?
3. Каково условие пересечения двух прямых?
4. Какие преобразования на плоскости Вам известны?
5. Что такое кривые Безье?
6. Какие методы закраски Вам известны?
7. Выпишите три закона Ома
8. Обоснуйте почему именно Вам нужно поставить автомат?

Вопросы?
Предложения?



Спасибо за внимание!



(с) КИС, ФРФикТ, БГУ,
2011