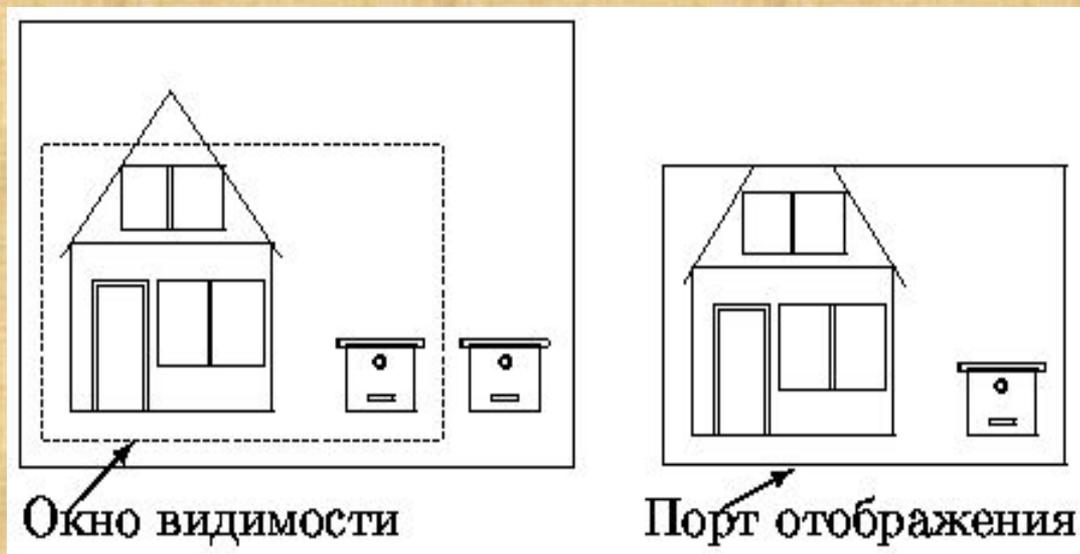


Компьютерная графика

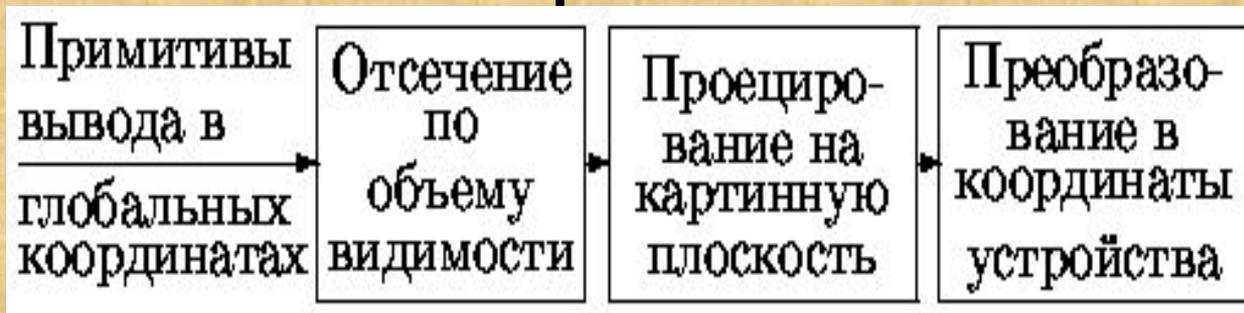
Содержание

1. Координаты и преобразования
 - 1.1 Проекция
 - 1.2 Стереои́зображения
2. Геометрическое моделирование
 - 2.1. Воксельное представление
 - 2.2. Точечное представление
 - 2.3. Конструктивная геометрия
 - 2.4. Каркасное представление

Пример визуализации для двумерных изображений



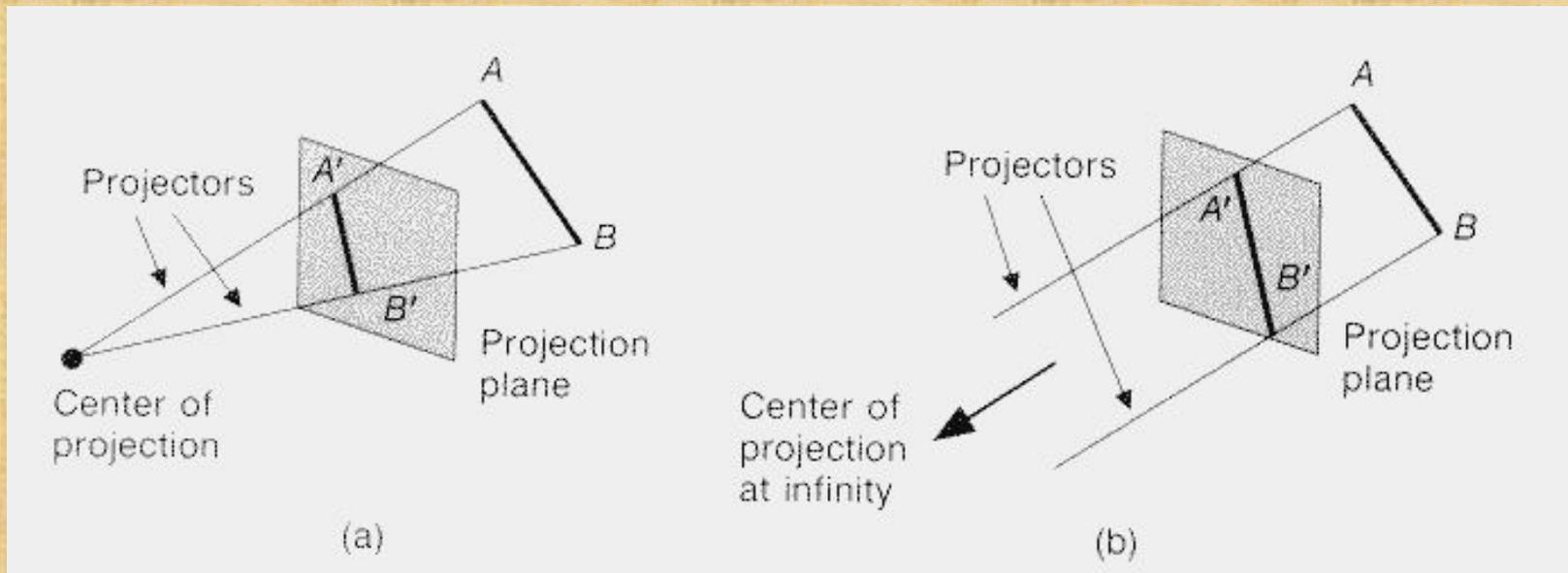
Визуализация трехмерных изображений



Модель процесса визуализации трехмерных изображений

Центральная и параллельная проекции

По расположению центра проекции относительно плоскости проекции различаются центральная и параллельные проекции.



При **параллельной проекции** центр проекции находится на бесконечном расстоянии от плоскости проекции. Проекторы представляют собой пучок параллельных лучей.

При **центральной (перспективной) проекции** расстояние от центра проекции до плоскости проецирования конечно, поэтому проекторы представляют собой пучок лучей, исходящих из центра проекции.

Типы проекций



Рис. 1.4: Классификация проекций

Параллельные проекции

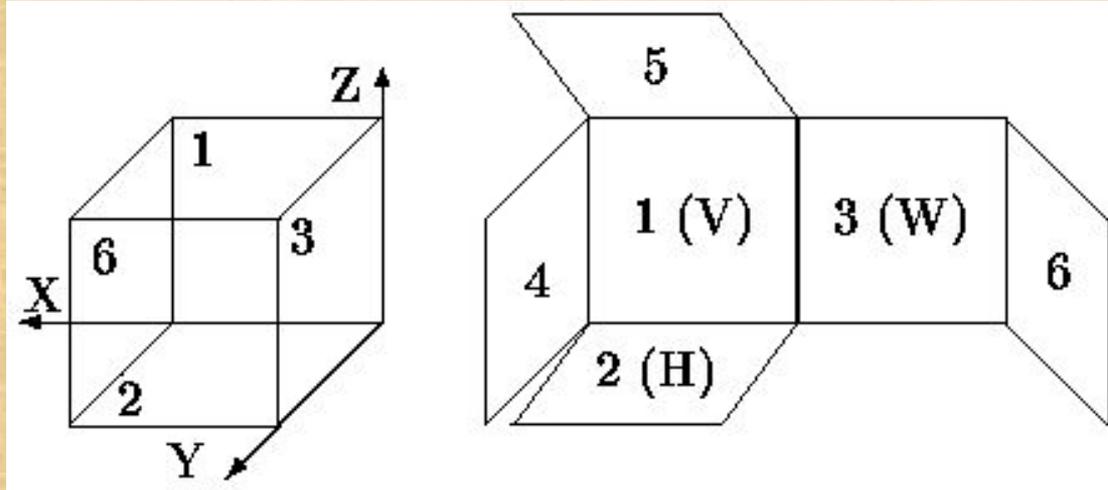


Рис. 1.5: Ортогональные проекции (основные виды) и их расположение на листе чертежа

- Вид спереди, главный вид, фронтальная проекция, (на заднюю грань V),
- Вид сверху, план, горизонтальная проекция, (на нижнюю грань H),
- Вид слева, профильная проекция, (на правую грань W),
- Вид справа (на левую грань),
- Вид снизу (на верхнюю грань),
- Вид сзади (на переднюю грань).

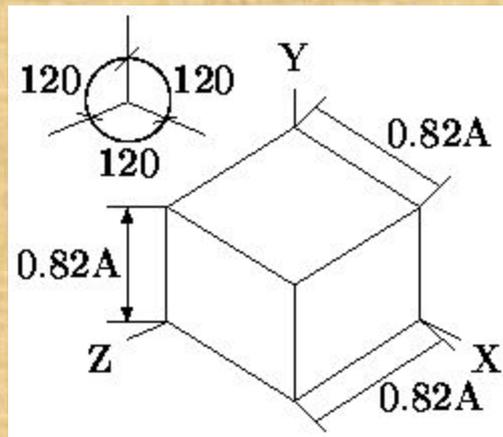


Рис. 1.6: Аксонометрическая прямоугольная изометрическая проекция куба со стороной A

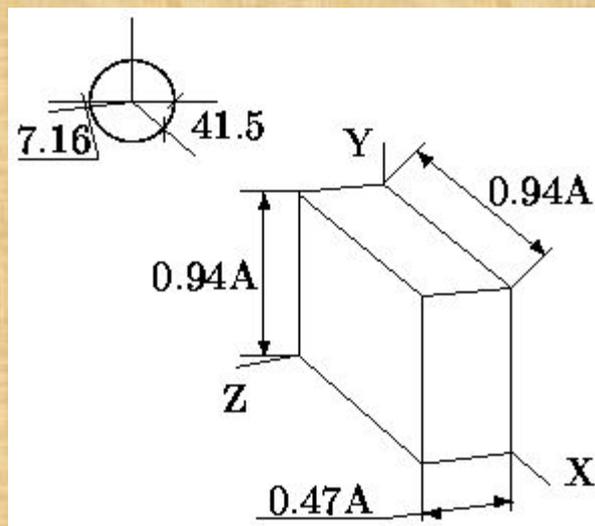


Рис. 0.1.7: Аксонометрическая прямоугольная диметрическая проекция куба со стороной A

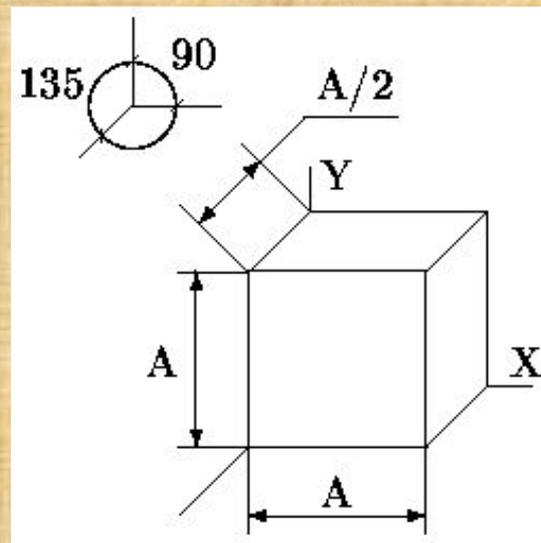


Рис. 1.8: Аксонометрическая косоугольная фронтальная диметрическая проекция куба со стороной A

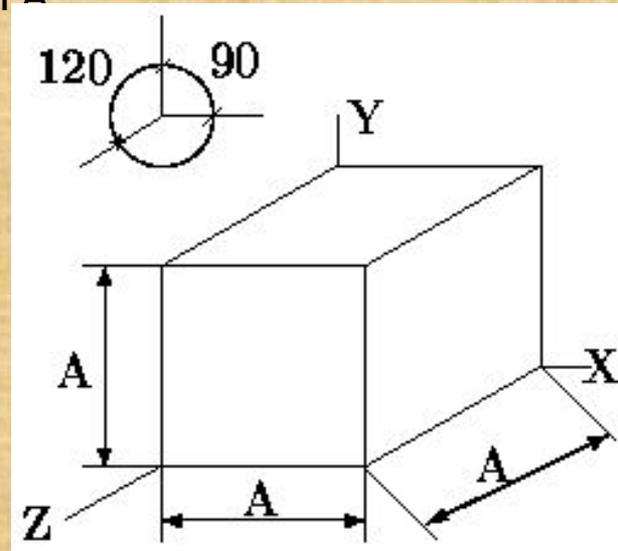


Рис. 1.9: Аксонометрическая косоугольная горизонтальная изометрическая проекция куба со стороной A

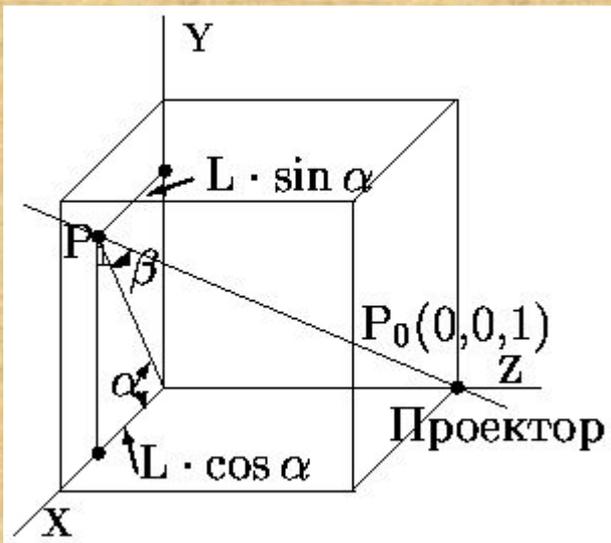
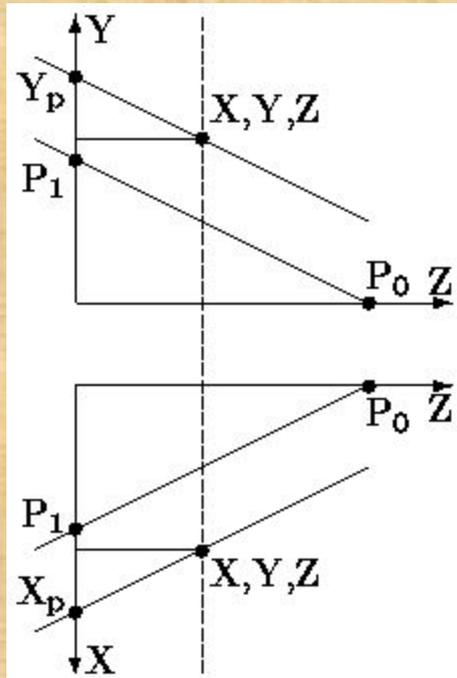


Рис. 1.10: Косоугольная параллельная проекция $P_1(L \cdot \cos \alpha, L \cdot \sin \alpha, 0)$ точки $P_0(0,0,1)$

$$[x_p \ y_p \ z_p \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ L \cdot \cos \alpha & L \cdot \sin \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



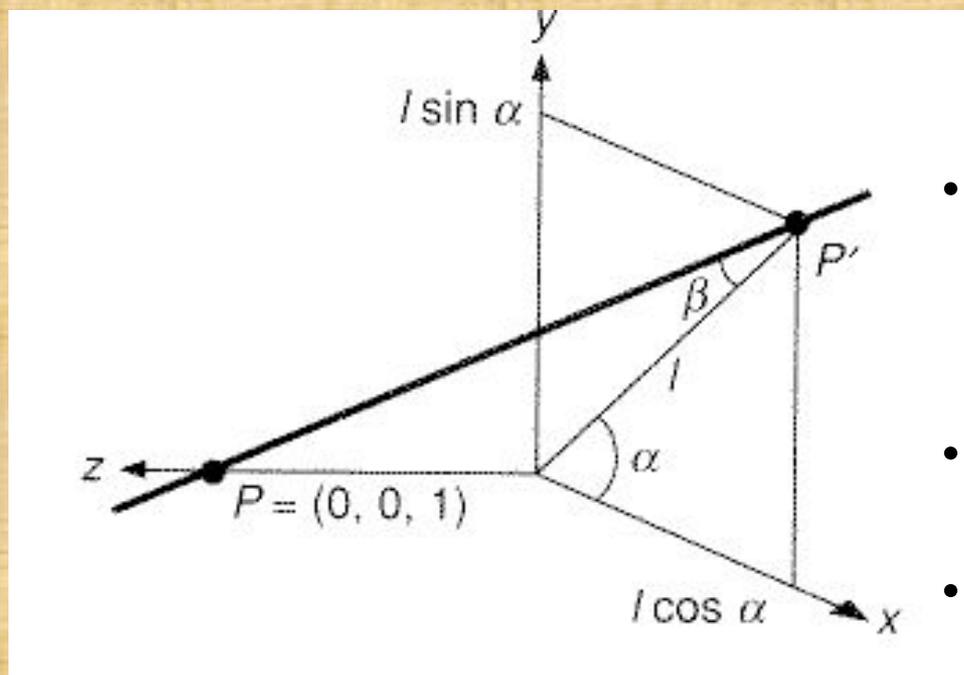
Из подобия треугольников получаем:

$$(X_p - X)/Z = L \cdot \cos \alpha \quad \square \quad X_p = X + Z \cdot L \cdot \cos \alpha$$

$$(Y_p - Y)/Z = L \cdot \sin \alpha \quad \square \quad Y_p = Y + Z \cdot L \cdot \sin \alpha$$

Рис. 1.11: Косоугольная параллельная проекция $(X_p, Y_p, 0)$

Различные варианты параллельных проекций формируются из полученной подстановкой значений L и углов α и β .



- Направление проецирования
 $(l \cos \alpha, l \sin \alpha, 1)$
- Угол между косыми проекторами и плоскостью проекции
 $\beta = \arcsctg(l)$
- Военная проекция
 $\beta = \arcsctg(1) = 45^\circ$
- Кабинетная проекция
 $\beta = \arcsctg(1/2) = 63.435^\circ$

Специальные виды перспективных проекций

Специальные перспективные проекции - проекции на цилиндрические, конические, сферические и др. поверхности с последующим разворачиванием полученной проекции на плоскость.

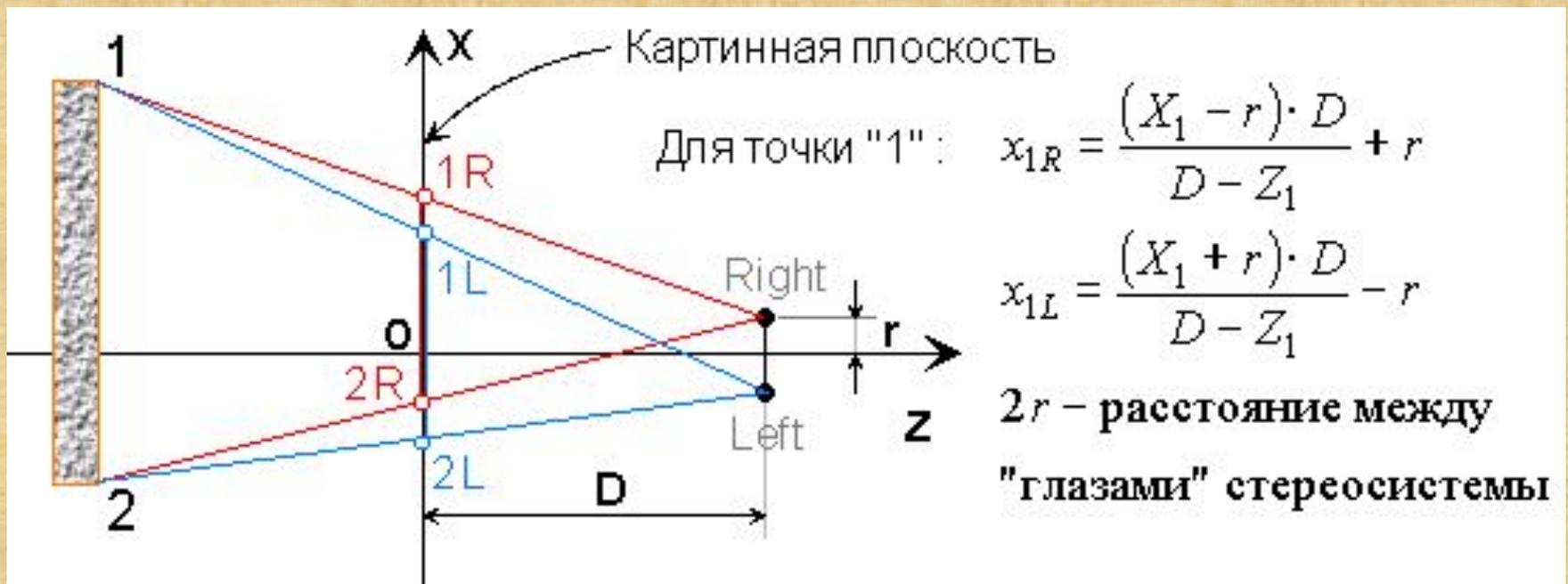
Проекция на **цилиндрическую** поверхность позволяет показывать объекты с очень большими углами зрения - вплоть до круговой панорамы.

Трудности **сферических** проекций прежде всего в том, что сфера на плоскость без разрывов не разворачивается. Если точки пространства проецируются на поверхность сферы лучами, проходящими через некоторую точку внутри сферы, то перепроецировать их отображения на плоскость можно множеством способов, например ортогональным проецированием параллельными лучами, способом развертки меридианов и др.

Среди этих различных способов выделяется *стереографический*; в нем со сферы на плоскость точки перепроецируются прямолинейными лучами, проходящими через полюс сферы, диаметрально противоположный тому, в котором сфера касается плоскости. При этом способе углы между пересекающимися линиями на сфере и их отображениями на плоскости равны.

Стереопроекция

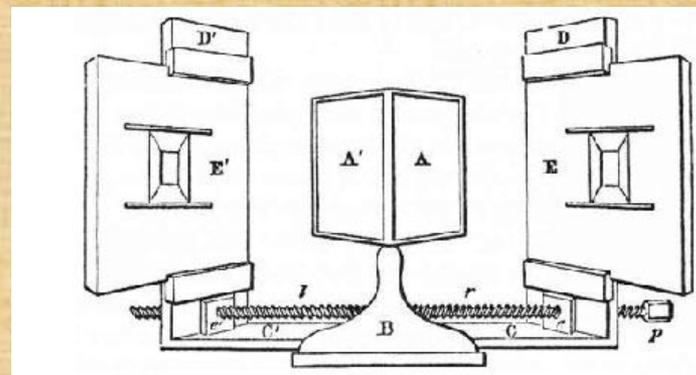
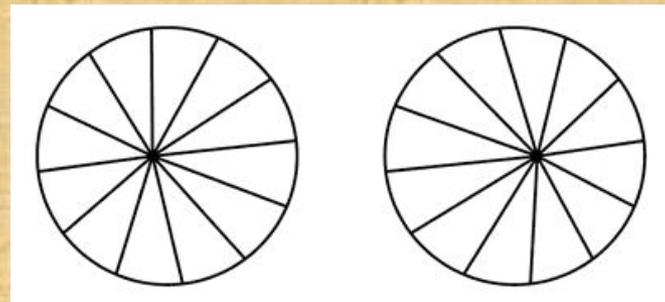
Еще один вид специальных проекций - **стереоскопические**. Простейший вид стереоизображения образуется с помощью стереопары - двух перспективных проекций, построенных каждая для своего глаза. Принцип создания и расчета таких изображений показан на рисунке.



Виды стереоизображений

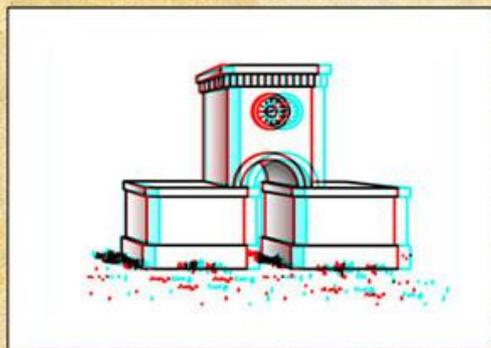
- цветное разделение (*построение анаглифов*), когда, например, изображение для левого глаза строится красным цветом, а для правого - синим и для просмотра используются цветные очки;
- пространственное-временное разделение (глаз видит изображение, предназначенное именно для этого глаза, в то время как картинки для обоих глаз демонстрируются на одной и той же поверхности)
- временное разделение («Палфрич» феномен), когда для левого глаза используется одна страница видеопамяти, а для правого - вторая и происходит их переключение с достаточно большой частотой кадровой развертки.
- *автостереограммы* (autostereogram).

Анаглифы



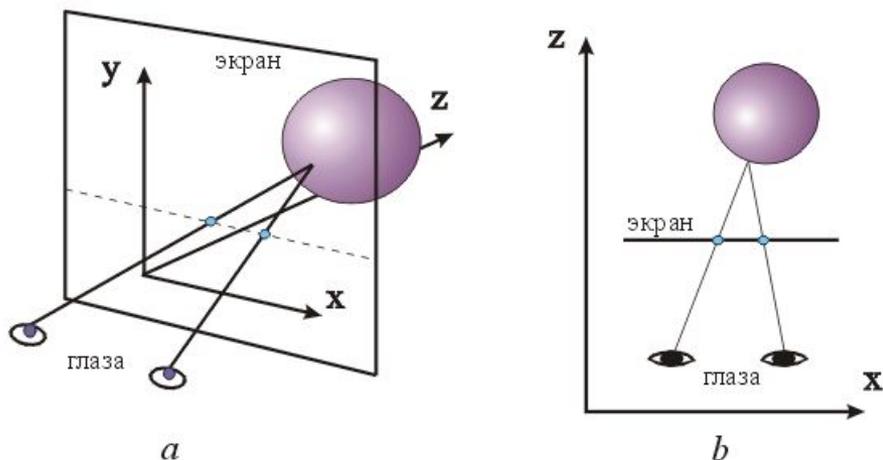
Сэр Чарльз Уитстоун
1838г.

Fig. 20



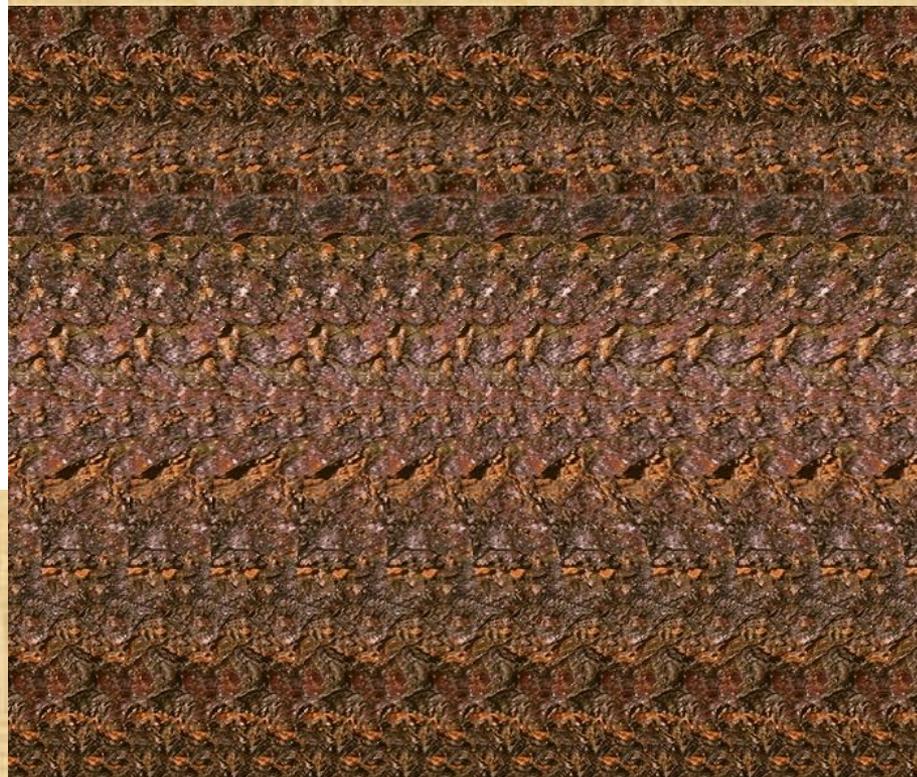
Sir Charles Wheatstone, 1838

Автостереограммы



Проецирование трёхмерного объекта на экран.

Основной принцип построения автостереограммы: две точки одного цвета генерируются на экране для формирования образа соответствующей точки поверхности объекта. Таким образом, когда наблюдатель разводит глаза так, чтобы смотреть за плоскость экрана, мозг принимает две точки одного цвета за одну, находящуюся на большей глубине.



Пример автостереограммы для функции

$$z = \frac{\cos(x^2 + y^2)}{x^2 + y^2 + 1};$$

Геометрическое моделирование

Понятие о моделировании

- Модель – это абстрактное представление сущности реального мира
 - Математическое моделирование физических, химических процессов и др.
- Компьютерное моделирование
 - Данные о физических объектах не могут быть целиком введены в компьютер
 - Необходимо априори ограничить объем хранимой информации об объекте
- Задача моделирования
 - найти вид модели, наилучшим образом отвечающий решаемой задаче

Геометрическое моделирование

- В компьютерной графике используется геометрическое моделирование
 - моделирование объектов различной природы с помощью геометрических типов данных
- Выбор модели
 - максимально использовать возможности графической системы
 - учесть задачи обработки и редактирования модели

Существует огромное количество геометрических моделей и их ВИДОВ

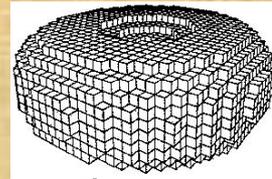
- Воксельное
- Точечное представление
- Конструктивная геометрия
- Каркасное представление
- Граничное представление первого порядка
(Векторная полигональная модель)
- Граничные представление высших
порядков
- На основе изображений
- Гибридные модели

Представления объектов

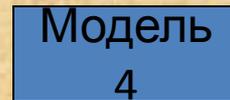
- Представление объекта почти = модель объекта
- Один и тот же объект может иметь несколько представлений (моделей)
 - Представления могут быть получены как из исходного объекта, так и путем преобразования другого представления объекта



объект 1



объект 2



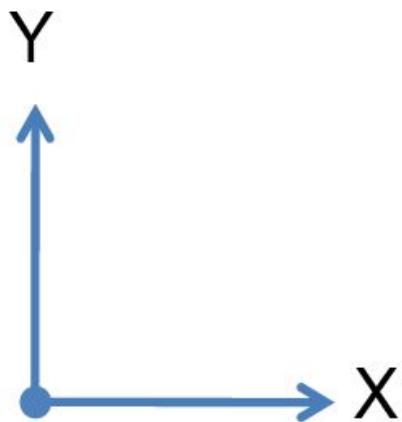
«Представления объекта 1»

«Представления объекта 2»

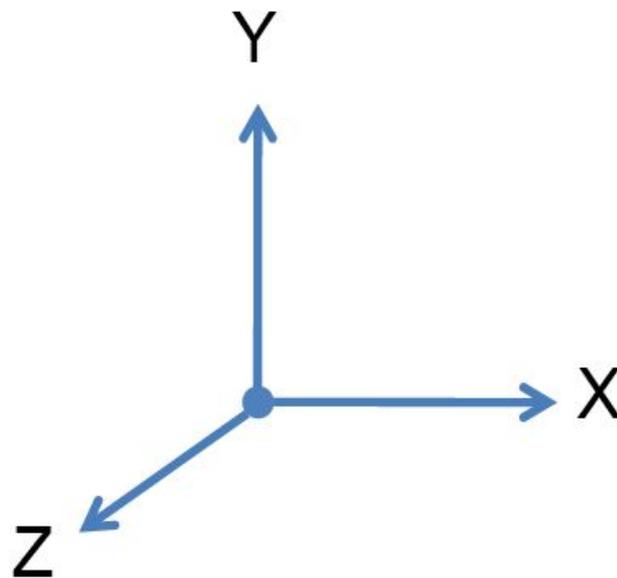
Рассматриваем **сплошные тела**

- Характеристики модели необходимые для правильного выбора типа модели:
 - структура данных;
 - алгоритм построения;
 - количество памяти, необходимое для хранения модели;
 - типичные свойства представления (алгоритмы);
 - область применения моделей в данном представлении.
- Каждую модель можно охарактеризовать по набору параметров
 - Объем/Поверхность
 - Какие свойства трехмерного объекта описывает модель?
 - Дискретное/Непрерывное
 - Содержится ли в модели информация о дополнении дискретных данных до непрерывных?
 - Явное/Параметрическое
 - Способ получения трехмерных координат точек, принадлежащих модели

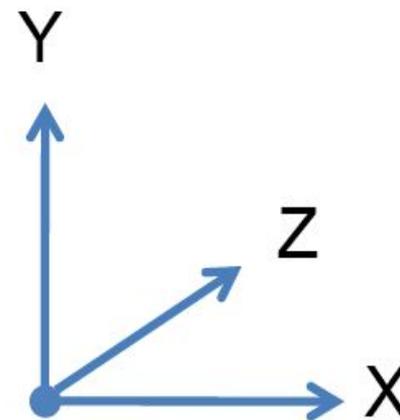
Системы координат для моделей



Двухмерная



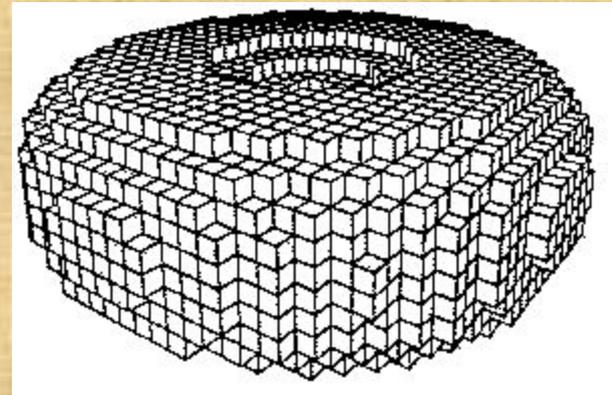
Правосторонняя



Левосторонняя

Воксельное представление

- Структура
 - Равномерная сетка, каждый элемент которой показывает, если в нем часть объекта
 - Ячейка называется воксель (voxel = volume element)
 - Каждый воксель принимает значение 0 или 1
 - Может также задавать плотность (0-1)
- Способ получения
 - Дискретизация трехмерных данных на равномерной сетке



```
bool Voxels[X][Y][Z];
```

```
BYTE Voxels[X][Y][Z];
```

Воксельное представление: свойства

+

- Каждый воксел имеет свой цвет и прозрачность.
- Дискретное представление: приближение реального объекта (чем больше вокселей и чем их размер меньше, тем точнее)!
- Простая процедура отображения объемных сцен
- Простое выполнение топологических операций над отдельными объектами и сценой в целом (разрез – прозрачные воксели)

-

- Плохо описываются части объекта, не параллельные сторонам воксельного куба
- Возникают проблемы при масштабировании изображения (ухудшение разрешения)
- Размер данных пропорционален кубу разрешения сетки
 - 1 байт на точку:
 $2000 \times 2000 \times 2000 = 7,45 \text{ Гб} !$
 - скорость создания отображений

Воксельное представление:

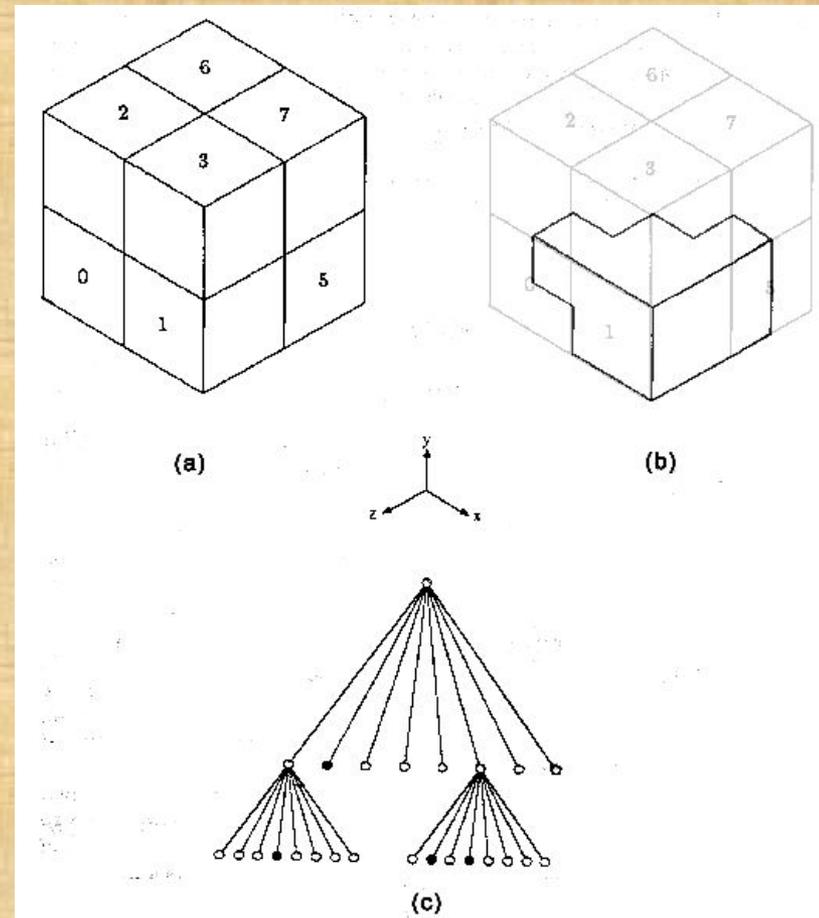
Типичные алгоритмы - пространственные алгоритмы:

- вычисление объема объекта
- нахождение центра масс
- булевы операции (пересечение, объединение)

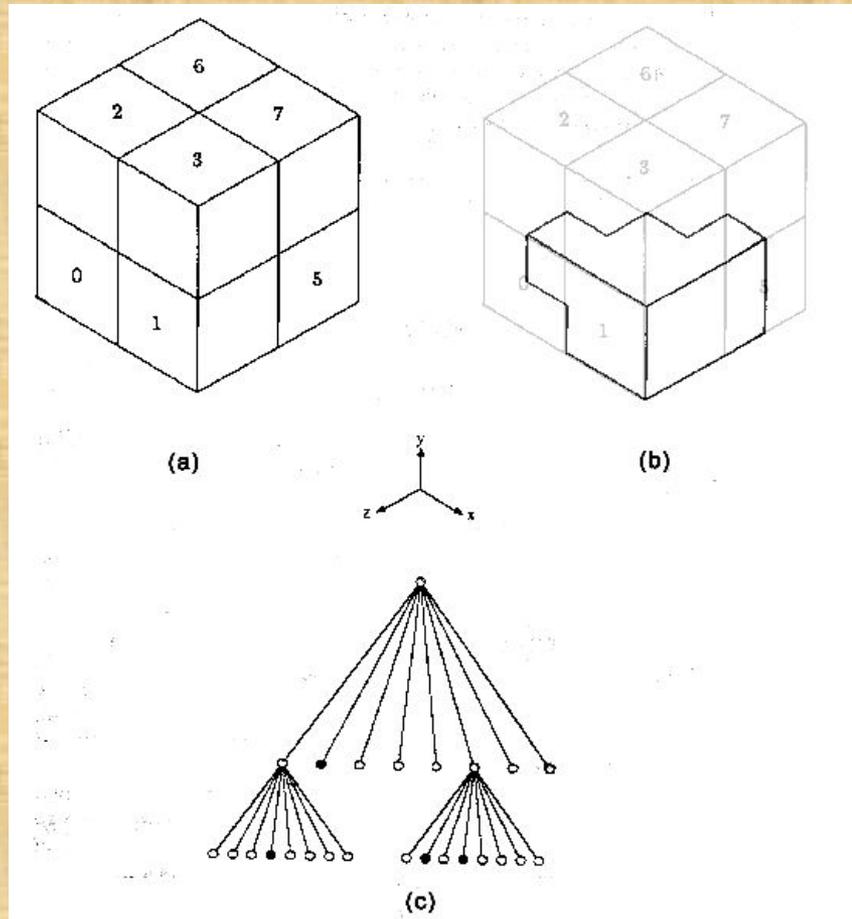
Плохо работают алгоритмы, требующие понятия поверхности!

Воксельное представление - улучшение: Октодерево

- Разбиение пространства на восемь октант, которое представляется деревом.
 - Упорядоченно!
- Располагается вокруг начала его локальной системы координат
- Октанты первого уровня совпадают с октантами системы координат.
- Ветвь дерева:
 - Код
 - Код = «черный»
 - Пространство заполнено, лист
 - Код = "белый«
 - Пространство пустое, лист
 - Код = "серый"
 - область пространства частично пуста и частично заполнена.
 - 8 ссылок указывают на подразбиение данной области.
 - Восемь указателей на восемь потомков, пронумерованных от 0 до 7



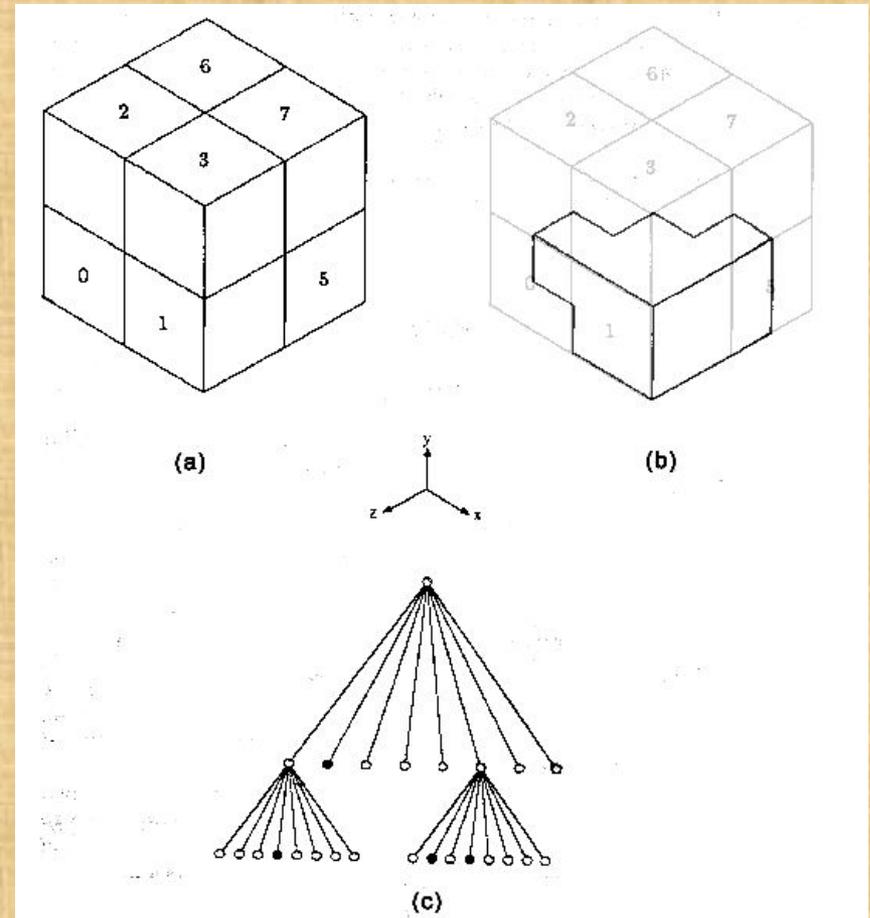
Октодерево: свойства



- Применяется для оптимизации воксельного представления
- Свойства:
 - Позволяет хранить информацию только о блоках, относящихся к объекту
 - Число элементов пропорционально площади поверхности объекта, т.е. квадрату разрешения
 - Для разреженных моделей позволяет уменьшить размер в тысячи раз!
- Способ получения
 - Из воксельного представления или напрямую, через дискретизацию

Линейная запись октодерева 1

- Октанты дерева пронумерованы от 0 до 7
- Конструирование адреса каждой ветви дерева, кроме корня.
- Адрес ветви уровня i – последовательности i чисел от 0 до 7 – путь от корня к этой ветви
- Символ X: если в последовательности чисел меньше, чем максимальное разрешение
- Линейная запись дерева есть просто сортированный массив адресов ветвей с кодом "черный"
- Пример: {03,1X,51,53}



Октодерево: типичные алгоритмы

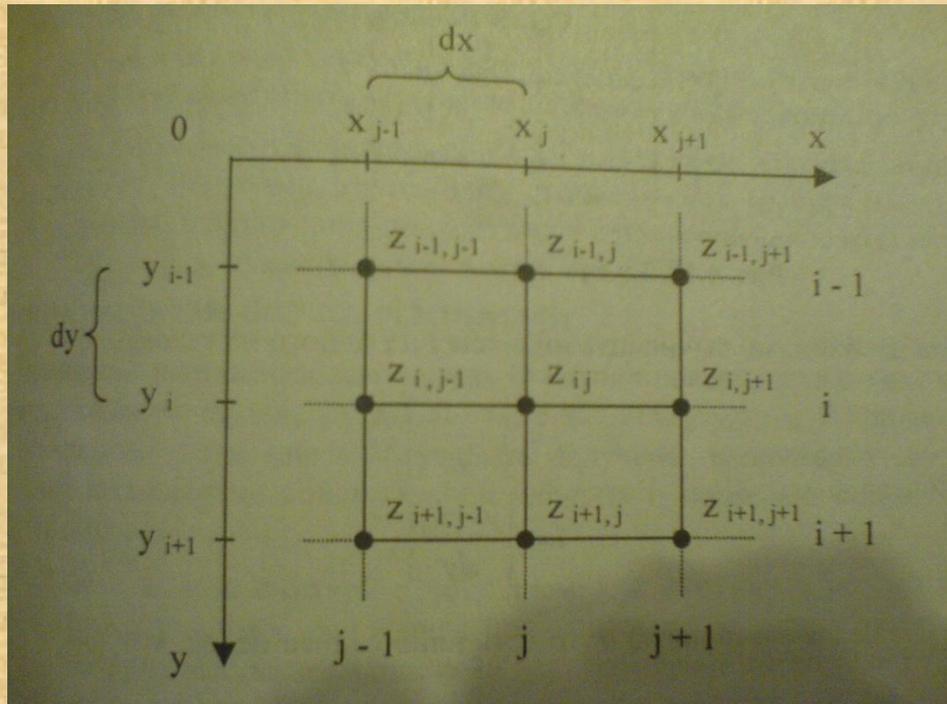
- Удобно для синтеза:
 - Переменный уровень детализации
 - Вывод back-to-front
- Усложняются операции, требующие информации о смежных ячейках
- Проблемы с анимацией – надо постоянно перестраивать дерево

Точечное представление

- Набор неструктурированных точек
- Количество памяти, необходимое для хранения модели пропорционально количеству точек (дискретов, *sample*)
- Дискретное, явное представление



Точечное представление: структура



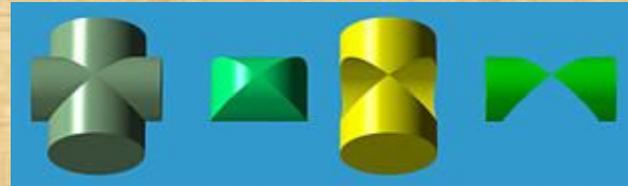
- Массив точек с атрибутами
 - Атрибуты: положение (x_i, y_j) , цвет, нормаль (z_{ij}) , размер (dx, dy)
- Описывает только принадлежащие объекту части пространства
- Явное хранение координат => возможное увеличение размера (для «плотных» моделей)

Точечное представление: свойства, типичные алгоритмы

- Эффективно аппроксимирует форму объекта
- По объему памяти обычно более эффективно чем воксельное, но менее эффективно, чем октодерево (или аналогичная структура)
- Нет связанности, инцидентности => для выполнения преобразований обычно строятся дополнительные структуры данных (октодерево)

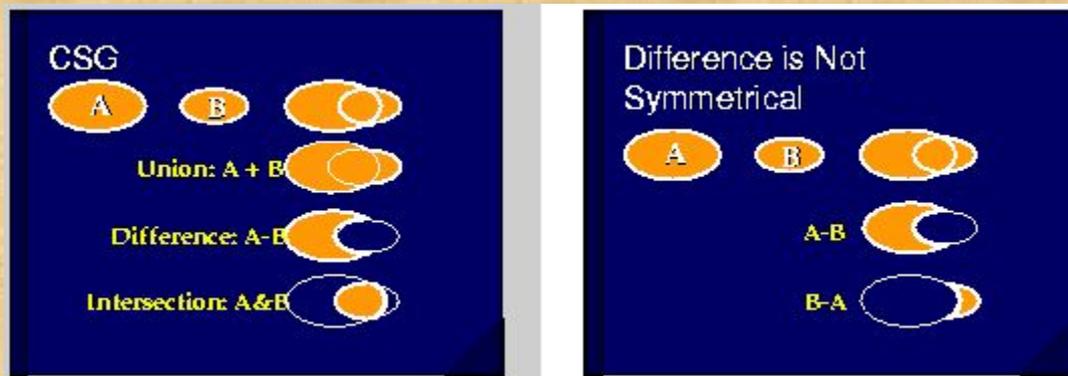
Конструктивная геометрия

- Структура
 - Набор базовых примитивов
 - сфера, куб, цилиндр...
 - Операции по их комбинированию
- Способ получения
 - Ручное моделирование
- Свойства
 - Описывает объем и поверхность (!)
 - Непрерывное представление
 - Явное представление



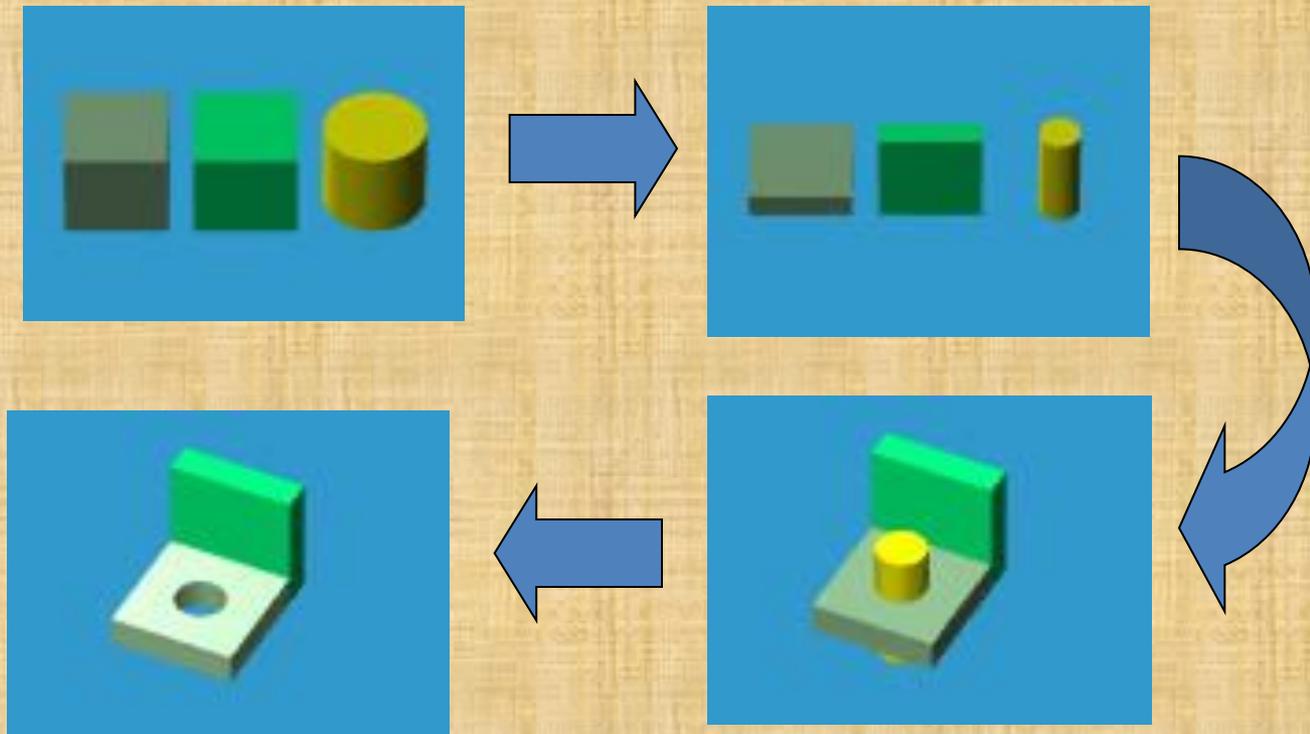
Конструктивная геометрия: операции с телами

- Операции
 - Перенос/поворот/масштабирование
 - Теоретико-множественные:
 - Объединение
 - Разность
 - Пересечение



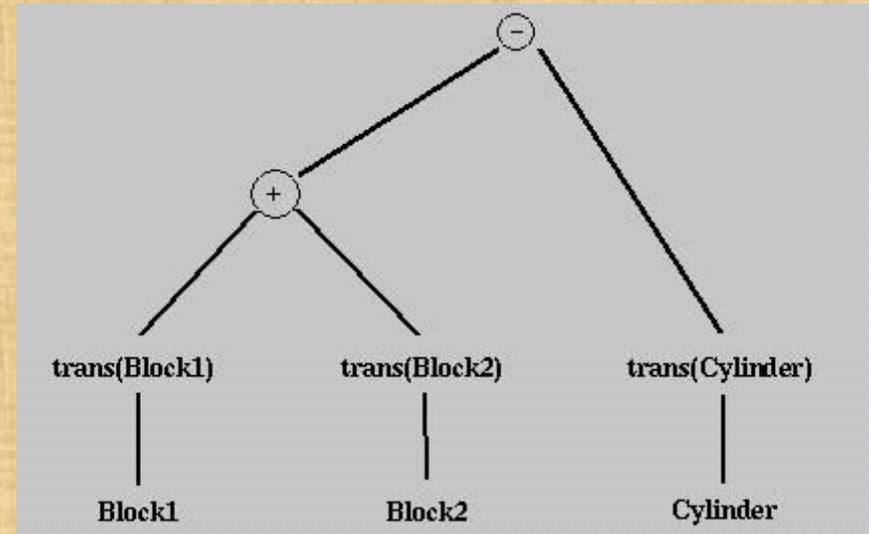
Конструктивная геометрия: операции с телами 2

- $\text{diff}(\text{union}(\text{trans1}(\text{Block1}), \text{trans2}(\text{Block2})), \text{trans3}(\text{Cylinder}))$



Конструктивная геометрия: структура данных

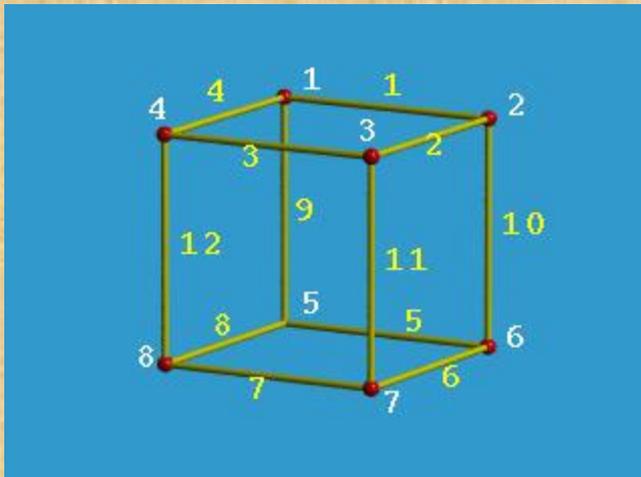
- Дерево из операций и базовых объектов
- Корень – результирующий объект
- Листья – базовые примитивы
- Число потомков равно числу операндов операции
- Из-за повторного использования превращается в направленный ациклический граф.



Конструктивная геометрия: типичные алгоритмы

- Пространственные алгоритмы
 - вычисление объема объекта
 - нахождение центра масс
- Есть понятие поверхности!

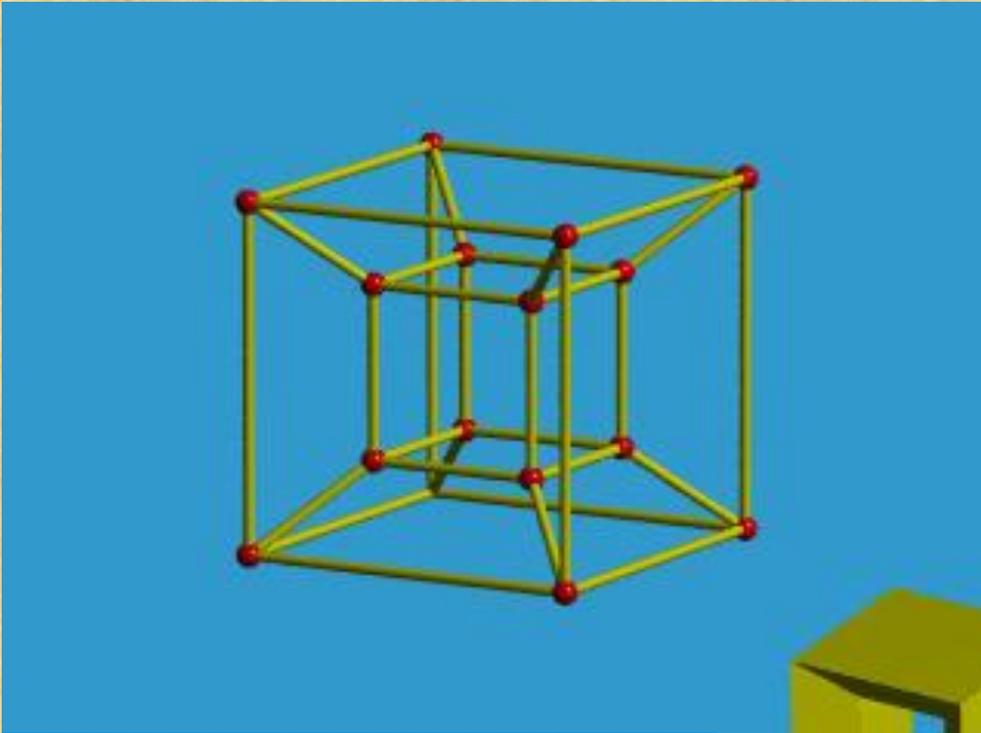
Каркасное представление



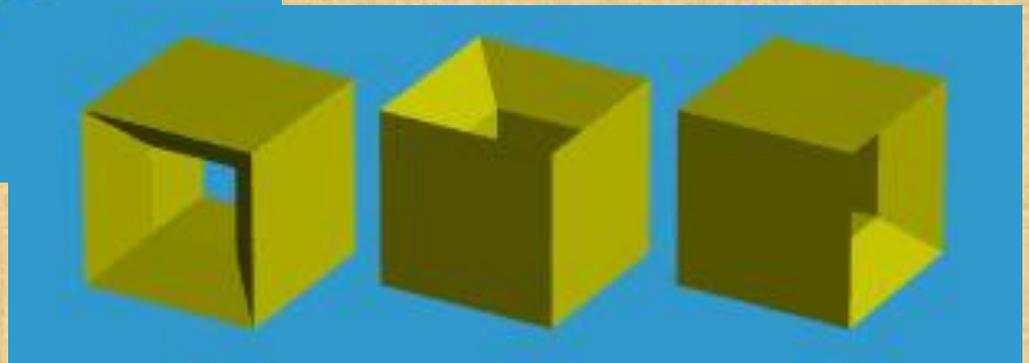
i	X	Y	Z
1	1	1	1
2	1	-1	1
3	-1	-1	1
4	-1	1	1
5	1	1	-1
6	1	-1	-1
7	-1	-1	-1
8	-1	1	-1

1	(1,2)	2	(2,3)	3	(3,4)
4	(5,6)	5	(5,6)	6	(6,7)
7	(7,8)	8	(8,5)	9	(1,5)
10	(2,6)	11	(3,7)	12	(4,8)

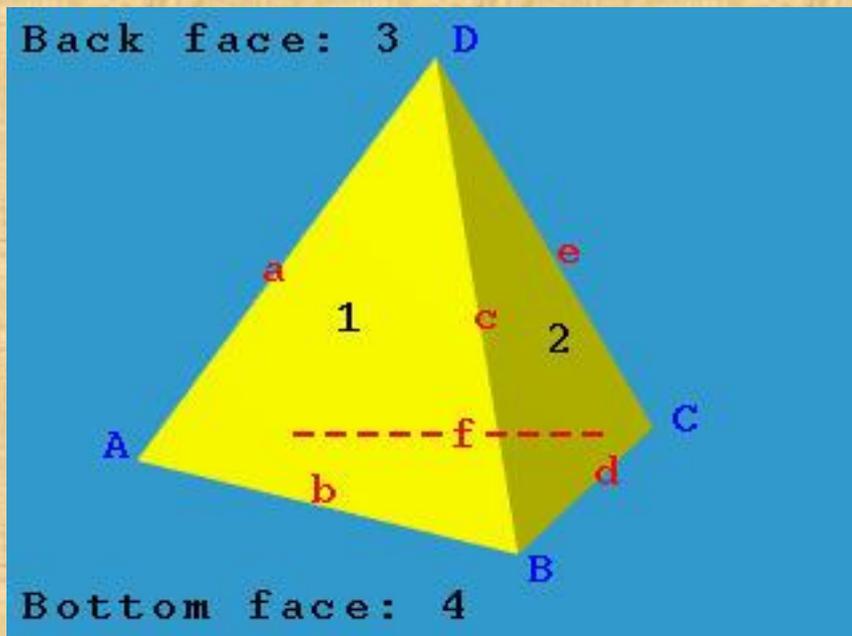
Каркасное представление: неоднозначная интерпретация



Нужна дополнительная информация!

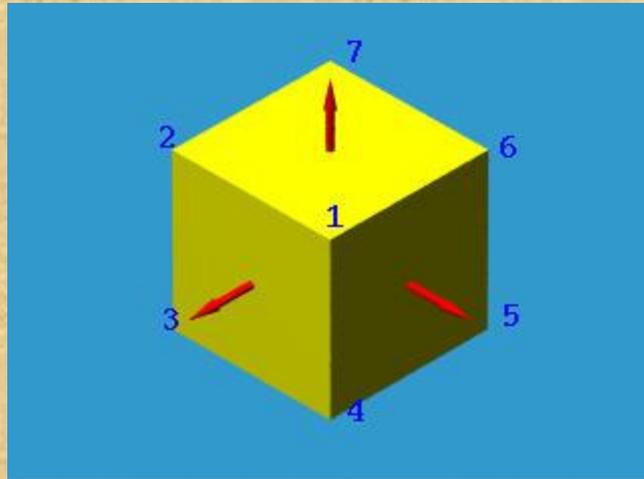


Грани и ребра

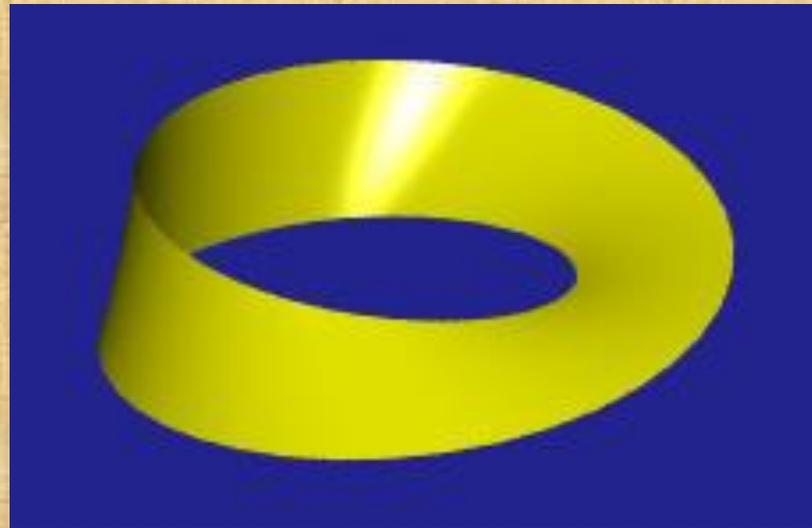


- Грани (Faces)
 - 1 {A,B,D}
 - 2 {B,C,D}
 - 3 {A,C,D}
 - 4 {A,B,C}
- Ребра (Edges)
 - a (A,D)
 - b (A,B)
 -

Поверхность



Две поверхности



Одна поверхность