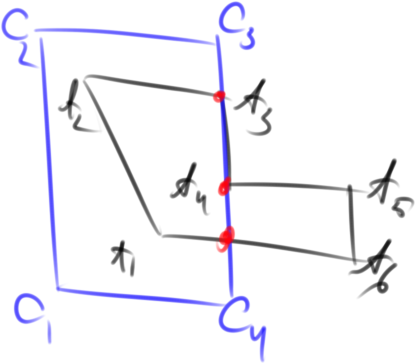
14.05.14-----------------------------------------------------------------------

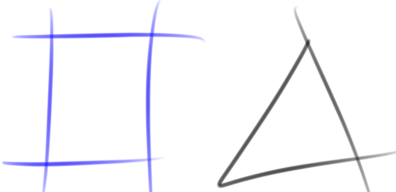
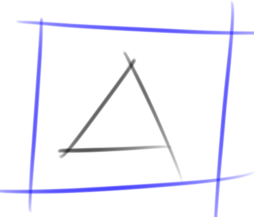
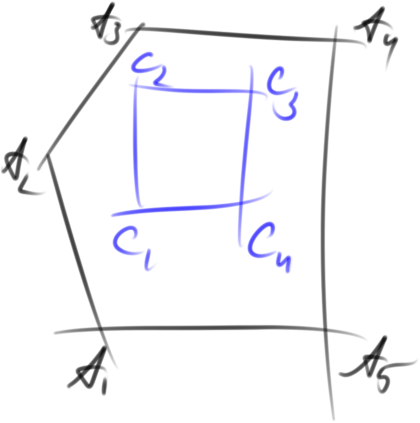
Частные случаи:



А3 – точка касания, не учитывается.

Мы ищем пересечения с границей отсекателя – однако оная относится к видимой части. Истиная граница ^ проходит на пиксель правее от ребра С3С4 – в А3 пересечения не будет.

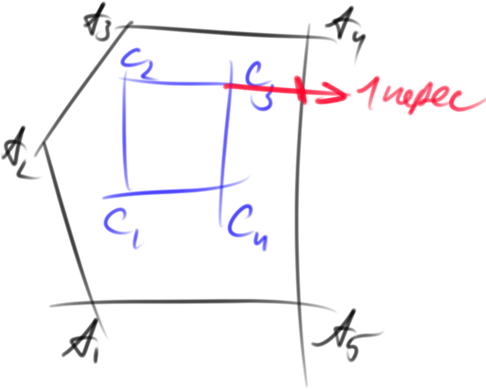
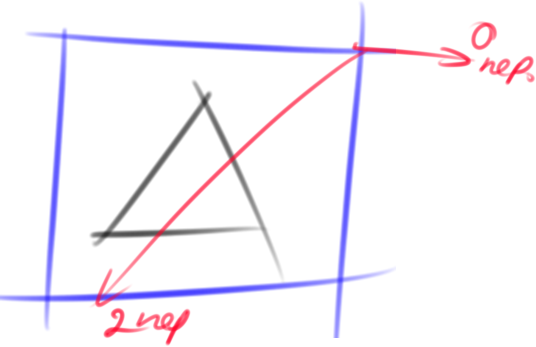
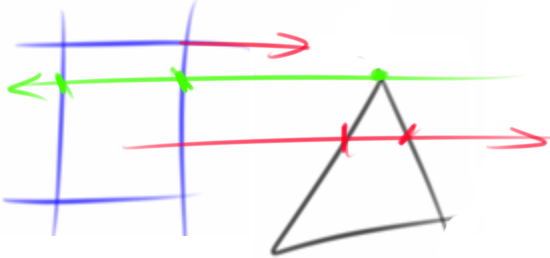
Точки пересечения делятся на входы и выходы с помощью векторного произведения вектора стороны отсекаемого на вектор стороны отсекателя. Положительное – вход, отрицательное – выход.



Точек пересечения может не быть.

Тест с лучом:

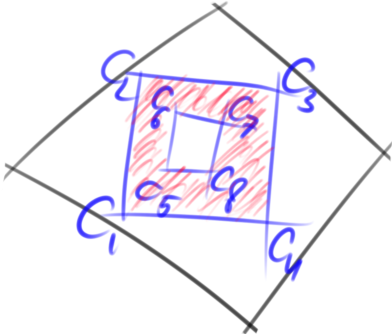
Из произвольной точки отсекателя испускается луч (например из вершины). Основной критерий – чётное или нечётное количество пересечений.

1)  2)  3) 

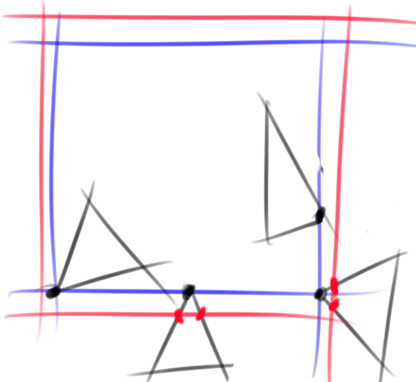
Если нечётное – отсекатель является обхватываемым. Если чётное – не обязательно обхватывающий. Нужно выпустить дополнительный луч из многоугольника в противоположную сторону.  
Если второе число пересечений будет чётным – то многоугольники внешние друг к другу; в результате отсечения видимых многоугольников нет.  
Если второе число пересечений нечётное – то отсекатель целиком охватывает многоугольник; в результате отсечения многоугольник полностью видим.

Если же луч проходит через вершину (касается) или же проходит сквозь всю сторону – ситуация неопределена, нужно взять другой луч (смещённый вверх или вниз).

^ в 1м случае, отсекатель расположен внутри многоугольника. У внутреннего многоугольника внешние границы совпадают с границами отсекателя. У внешнего многоугольника внешня граница совпадает с границей многоугольника; внутренняя граница совпадает с границей отсекателя; отсекатель «проделывает отверстие» в многоугольнике.



Внешняя граница отсекателя, попавшая внутрь многоугольника, является внешней границей внутреннего многоугольника, и внутренней границей внешнего многоугольника. Если граница отверстия попала внутрь многоугольника, то она проделывает отверстие во внутреннем многоугольнике.



В частном случае, когда происходит пересечение только в одной точке, отсекатель нужно расширить или уменьшить на один пиксель – точек пересечения станет несколько.

**Трёхмерная графика**

Используется модель трёхмерного объекта. Применяется машинное представление, дающее информацию о форме и размерах объекта: каркасное, поверхностное или объёмное.

Каркасная: проста при представлении, но не даёт полной информации о форме объекта.

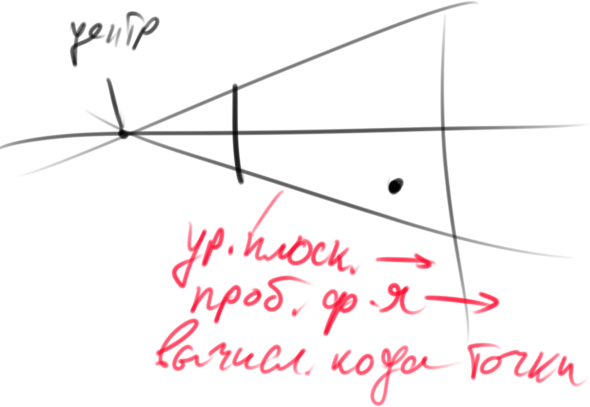
Поверхностные: не содержат информации о том, где находится материал, а где – пустота (по какую сторону поверхности).

Объёмные: добавляют к поверхности модели информацию о раположении материала. Материал обычно указывается направлением внутренней нормали.

Требования, предъявляеются к математической и к программистской модели.

Преобразования в трёхмерном пространстве требуют матрицы 4х4, чтобы была возможной реализация матрицы переноса.

Трехмерное отсечение: прямоугольным параллелепипедом или усеченной 4гранной пирамидой.



Определение факта выпуклости трёхмерного тела (основанный на переносах и поворотах.

**Задача удаления невидимых линий и плоскостей (граней)**

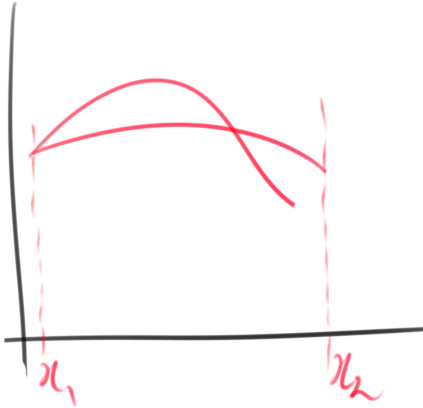
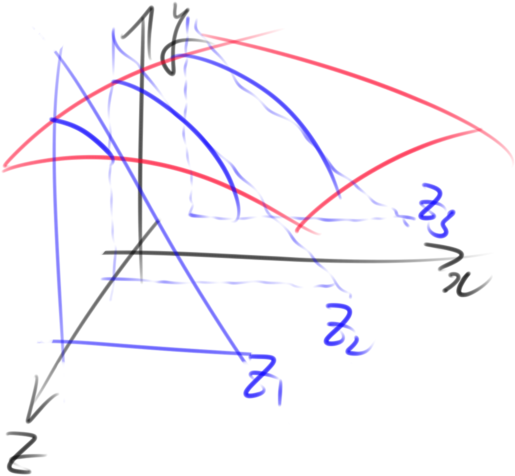
Осложняется тем, что необходимо учитывать положение наблюдателя.

В силу сложности, а так же из-за разных задач (статика\динамика), одного единого метода для всех задач не существует. Рассматривают пространство объектов (в мировой системе координат) и пространство изображения (экранная система координат).

В основу многих алгоритмов заложена сортировка объектов по «глубине» - расстоянию от наблюдателя. Трудоёмкость алгоритмов в объектном пространстве (N объектов) – N^2. Задача же в пространстве изображения (N объектов и M пикселей) – трудоёмкость N\*M. Таким образом, приоритет определяется тем, чего больше – пикселей или объектов.

**Алгоритм плавающего горизонта**

Предназначен для изображения поверхностей, задаваемых неявным уравнением F(x,y,z)=0. В общем случае считается, что наблюдатель расположен на оси З в бесконечности. Поверхность рассекается плоскостями, параллельными ОХУ



Ближайшая к наблюдателю кривая видима всегда. Далее в каждом сечении имеем уравнение y = f(x, z=const). Для очередной кривой вычисляем Yj(x\_i) – она будет видима, если она расположена выше самой верхней точки кривой j-1, или ниже самой нижней её точки. . Таким образом, нижний горизонт – массив минимальных значений функции, верхний – массив максимальных значений.

Если мы можем вычислить значение функции в каждой точке, то алгоритм достаточно просто выполняется. Очередная точка текущей кривой является видимой, если её ордината больше текущего максимума (по предыдущим кривым) или меньше текущего минимума.

Вычисления могут производиться не с шагом=1 по оХ. Если очередная точка является невидимой, то она не соединяется с предыдущей. Наоборот, если видима, то соединяется. Для упрощения вычислений можно использовать линейную аппроксимации.

