

## 7. Удаление невидимых линий и поверхностей.

### 7.1. Задача удаления невидимых линий и поверхностей, её роль и значение в компьютерной графике.

Пусть заданы трехмерный объем и видовые параметры, описывающие тип проекции, проекционную плоскость и т. д., и требуется определить, какие ребра и поверхности объекта видимы, если смотреть из центра проекции (для центральных проекций) или вдоль направления проецирования (для параллельных проекций). Только эти ребра и поверхности мы будем выводить на экран. Для решения этой задачи [2,3,4] требуется много процессорного времени, поэтому оправданы разработки многочисленных и тщательно структурированных алгоритмов.

Анализ существующих на сегодняшний день алгоритмов удаления невидимых поверхностей позволяет выявить существенную разницу между двумя подходами. Эти подходы ориентированы на работу с отображаемыми объектами и работу с формируемым изображением. Обобщённый подход, основанный на анализе пространства объектов, предполагает попарное сравнение положения всех объектов по отношению к наблюдателю. Существует конечное множество вариантов расположения объектов относительно друг друга и наблюдателя (объект А полностью закрывает В, объект В полностью закрывает А, А и В не перекрываются или перекрываются частично). Анализ показывает, что такой объектно – ориентированный подход выгоднее применять для относительно несложных сцен. Грубая оценка сложности для этих методов –  $O(k^2)$ .

Подход, основанный на анализе пространства изображений, похож на моделирование процесса визуализации методом приведённых лучей. Можно рассмотреть луч, исходящий из центра проецирования и проходящий через определённый пиксель плоскости. Этот луч в какой – либо точке пересечёт плоскость многоугольника. Можно вычислить точку пересечения луча с плоскостью и выяснить, лежит ли точка во внутренней области многоугольника. После

этого надо выбрать многоугольник, который первым пересекает луч. Грубая оценка сложности для этих методов –  $O(k)$ .

Упрощение сравнений по глубине: перспективное преобразование. Удаление скрытых поверхностей должно проводиться в трехмерном пространстве до проецирования на плоскость, при котором теряется информация о третьей координате, необходимая для проведения сравнений по глубине. Основное сравнение по глубине можно свести к следующему вопросу: закрывает ли одна из двух заданных точек  $P_1$  и  $P_2$  другую? Этот вопрос эквивалентен следующему: лежат ли точки  $P_1$  и  $P_2$  на одном и том же проекторе? Если да, то, сравнивая  $z_1$  и  $z_2$  можно определить, какая точка ближе к наблюдателю. Если же нет, то ни одна из точек не закрывает другую.

Сравнения по глубине обычно производятся после применения нормирующего преобразования, поэтому при параллельном проецировании проекторы параллельны оси  $z$ , а при центральном проецировании выходят из начала координат. Следовательно, при параллельном проецировании точки лежат на одном проекторе, если  $x_1 = x_2$  и  $y_1 = y_2$ , а при центральном проецировании для определения принадлежности точек проектору приходится выполнять четыре операции деления, чтобы выяснить, имеют ли место равенства  $x_1/z_1 = x_2/z_2$  и  $y_1/z_1 = y_2/z_2$ . Более того, если точка  $P_1$  затем сравнивается с некоторой точкой  $P_3$ , две из этих операций придется повторить. Лишних операций деления можно избежать, если произвести такое преобразование трехмерного объекта, при котором параллельная проекция преобразованного объекта представляла бы то же самое, что и центральная проекция первоначального объекта. Тогда проверка, закрывает ли одна из точек другую, будет такой же, как и для параллельных проекций. Преобразование деформирует объекты и перемещает центр проекции в бесконечность вдоль отрицательного направления оси  $x$ , при этом проекторы оказываются параллельными.

Матрица

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1 - z_{v.\min}} & 1 \\ 0 & 0 & \frac{-z_{v.\min}}{1 - z_{v.\min}} & 0 \end{bmatrix}$$

преобразует нормированный видимый объем при центральной проекции в прямоугольный параллелепипед, который определяется следующими соотношениями:

$$-1 \leq x \leq 1, \quad -1 \leq y \leq 1, \quad 0 \leq z \leq 1,$$

Отсечение, однако, не может проводиться относительно прямоугольного параллелепипеда, описываемого уравнениями, так как деление на  $z$ , выполняемое в матрице  $M$  для формирования параллелепипеда, приводит к потере информации, необходимой для определения знака  $z$ -координаты точки. Отсечение относительно нормированного видимого объема, представляющего собой усеченную пирамиду, можно выполнять до применения матрицы  $M$ , однако после этого результаты отсечения надо умножить на матрицу  $M$ . Более привлекательным решением является объединение матрицы  $M$  с перспективным нормирующим преобразованием  $N_{\text{центр}}$  так, чтобы можно было обойтись лишь одним умножением на матрицу. Отсечение проводилось бы затем в однородных координатах до выполнения деления. Если результаты этого умножения обозначить через  $(X, Y, Z, W)$ , отсекающие пределы можно записать так:

$$-W \leq X \leq W, \quad -W \leq Y \leq W, \quad 0 \leq Z \leq W,$$

После проведения отсечения мы результат делим на  $W$  и получаем  $(x_p, y_p, z_p)$ . Теперь мы можем приступить к удалению скрытых поверхностей. Если задано параллельное проектирование, перспективное преобразование матрицы  $M$  не требуется, поскольку в этом случае при нормирующем преобразовании  $N_{\text{пар}}$  для параллельных проекций проекторы оказываются параллельными оси  $z_v$ .

Исключение сравнений по глубине. Оболочки. В некоторых алгоритмах, чтобы избежать ненужных сравнений между объектами, используются экранные оболочки введенные исключения лишних отсечений. Предположим, что многоугольники были подвергнуты преобразованию, которое описывается матрицей  $M$ . Тогда проецирование на плоскость  $xu$  выполняется тривиально путем установки  $z = 0$  для каждой вершины. Если бы это преобразование не применялось, для построения проекции потребовалось бы деление на  $z$ . Если прямоугольные оболочки перекрываются, имеет место один из двух случаев: либо проекции многоугольников также перекрываются, либо этого не происходит. В обоих случаях для более подробного анализа требуются дополнительные сравнения. Во втором случае с их помощью будет установлено, что на самом деле два многоугольника не перекрываются. Можно воспользоваться оболочками, для окружения самих многоугольников, а не их проекций — в этом случае оболочки становятся пространственными. С другой стороны, их можно применить для указания границ в пределах одного измерения, чтобы выяснить, например, перекрываются ли два многоугольника в направлении  $z$ . Перекрытие по оси  $z$  отсутствует, если

$$z_{\max 1} \leq z_{\min 2} \quad \text{или} \quad z_{\max 2} \leq z_{\min 1}$$

Во всех этих случаях наиболее сложным является нахождение самих оболочек. Их можно определить, просматривая снова список вершин координат и фиксируя наибольшие и наименьшие значения по каждой координате.

## Классификация алгоритмов удаления невидимых поверхностей



Рис. 7.1. Классификация алгоритмов удаления невидимых линий и поверхностей.

Эффективность алгоритмов. Удаление скрытых поверхностей можно рассматривать как процесс сортировки, поскольку в этих алгоритмах встречаются многие примеры сортировки и поиска. Выбор эффективного алгоритма сортировки в большой степени влияет на быстродействие алгоритмов удаления скрытых поверхностей. Важно также не злоупотреблять упорядочиванием, так как обычно вполне достаточно использования когерентности. Например, в алгоритме построчного сканирования когерентность сканирующих строк используется для того, чтобы избежать полного упорядочения по  $x$  для каждой сканирующей строки.

В алгоритме сортировки по глубине применяется упорядочение по  $z$ , а затем по  $x$  и  $y$  (с помощью оболочек). В алгоритме построчного сканирования выполняется сортировка по  $y$  (с использованием групповой сортировки), а потом по  $x$  (вначале путем сортировки вставками,

а затем при обработке каждой сканирующей строки с помощью метода «пузырька»), и в заключение производится поиск по  $z$  многоугольника, ближайшего к точке зрения. При разбиении области проводится параллельная сортировка по  $x$  и  $y$ , а затем поиск по  $z$ . В алгоритме, использующем  $z$ -буфер, непосредственное упорядочение отсутствует, производится лишь поиск по  $z$ .

Порядок выполнения сортировки не имеет значения: упорядочение вначале по  $y$  не дает никаких особых преимуществ по сравнению с упорядочением по  $x$  или  $z$  и т. д. Это объясняется тем, что средний объект обладает одинаковой сложностью по всем трем направлениям. Нельзя, однако, сказать, что все алгоритмы одинаково эффективны: они различаются тем, насколько действенно применяется когерентность для исключения сортировки, а также тем, насколько целесообразно используется память и время. В табл. 7.1 приведены результаты сравнения производительности четырех алгоритмов [3]. Эти данные являются оценочными и можно пренебречь небольшими различиями в них, однако при этом свободно можно пользоваться сравнениями по порядку величины между различными алгоритмами, чтобы получить представление об эффективности применяемых методов. Алгоритм сортировки по глубине эффективен при небольшом числе многоугольников, поскольку для определения, может ли многоугольник быть преобразован в растровую форму, практически всегда достаточно простых проверок на перекрытие. С увеличением числа многоугольников требуется применение более сложных тестов и более вероятной станет необходимость в разбиении многоугольника.

Табл. 7.1.

Алгоритм	Число граней и многоугольников в сцене		
	100	2500	60 000

Упорядочение по глубине	$I^1$ )	10	507
Использование Z-буфера	54	54	54
Построчное сканирование	5	21	100
Разбиение области (алгоритм Warnock)	11	64	307

') Данные нормированы так, что этот элемент принят за единицу. производительностью,

Так как с ростом числа многоугольников в сцене число пикселей, покрываемых одним многоугольником, уменьшается. Однако такому алгоритму требуется большой объем памяти. Отдельные проверки и вычисления, применяемые в методе разбиения области, относительно сложны, поэтому в общем случае этот метод является более медленным, чем остальные.

Сравнительный анализ алгоритмов крайне затруднителен. Даже при работе с одной сценой может оказаться, что в разные моменты времени следует использовать различные алгоритмы. Ниже предлагается другой подход к оценке эффективности алгоритмов [4]. Он заключается в использовании тест – изображений (дом Роберта, гавань, большая гавань).

Табл. 7.2

Алгоритмы	Число операций сложения для реализации алгоритма		
	Дом Роберта	Гавань	Большая гавань
Roberts	$2,4 \cdot 10^6$	$510 \cdot 10^6$	$31 \cdot 10^9$
Appel	$1,8 \cdot 10^6$	$590 \cdot 10^6$	$97 \cdot 10^9$
Schumacher	$4,2 \cdot 10^6$	$25 \cdot 10^6$	$170 \cdot 10^6$
Newell	$140 \cdot 10^3$	$1,4 \cdot 10^6$	$71 \cdot 10^6$
Warnock	$1,5 \cdot 10^6$	$9 \cdot 10^6$	$43 \cdot 10^6$
Romney	$770 \cdot 10^3$	$2,9 \cdot 10^6$	$14 \cdot 10^6$
Watkins	$470 \cdot 10^3$	$3 \cdot 10^6$	$64 \cdot 10^6$
Z-буфер	$7,5 \cdot 10^6$	$7,5 \cdot 10^6$	$7,5 \cdot 10^6$

## 7.2.Метод количественной невидимости Аппеля

Алгоритм Аппеля [1,2,3] оперирует с телами, ограниченными плоскими гранями, каждая из которых представляет собой многоугольник. Такие тела называются многогранниками. Вершины многоугольников должны быть упорядочены по направлению обхода, противоположному движению часовой стрелки. Оперирует в **объектном пространстве**. Видимость объекта проверяется по отношению к конечной точке наблюдения, т. е. используется **центральная проекция**. Для определения видимости объектов исследуется видимость ребер их граней.

1. На первом шаге работы алгоритма удаляются все ребра, заслоняемые телами, к которым они принадлежат. Для этого используется тест видимости. С его помощью определяются полностью невидимые и потенциально видимые грани. Ребра, ограничивающие потенциально видимую грань, называются вещественными ребрами; вещественные ребра, принадлежащие одновременно потенциально видимым и невидимым граням, называются контурными ребрами.

При выполнении теста видимости сначала определяется нормаль к грани - направленный от тела вектор  $N$ , перпендикулярный к грани (уравнение нормали получается из уравнения плоскости, содержащей грань).

Для определения нормали к плоскости вычисляют нормали к каждой вершине многоугольника с помощью векторного произведения прилежащих ребер. Например, для треугольника имеем:

$$A(x_1, y_1, z_1), \quad B(x_2, y_2, z_2), \quad C(x_3, y_3, z_3);$$
$$N_A = \begin{bmatrix} i & j & k \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{bmatrix} \quad \text{и т. д.}$$

а затем усредняют результаты.



После этого строится линия визирования  $L$  - прямая, проходящая через точку наблюдения и основание нормали, и вычисляется угол между  $N$  и  $L$ . Грань потенциально видима, если этот угол меньше или равен  $\pi/2$ , и невидима, если угол больше  $\pi/2$ . Так как «критическим углом» является угол  $\pi/2$ , то надо вычислить скалярное произведение  $L \cdot N = |L| \cdot |N| \cdot \cos \phi_{LN}$  и проверить его знак ( $\cos \phi_{LN}$  изменяет знак при  $\phi_{LN} = \pi/2$ ). Для одного выпуклого тела такой способ полностью определяет видимость его граней. Если тело не одно, этот способ можно использовать лишь для выделения невидимых граней. К потенциально видимым граням необходимо применить дополнительные тесты, с помощью которых можно обнаружить взаимное затенение граней, принадлежащих разным телам или одному, но не выпуклому телу.

2. На втором шаге необходимо определить видимость всех потенциально видимых ребер. Вводится в рассмотрение количественная невидимость точек (или коротких отрезков) на потенциально видимом ребре. В данном случае можно не применять неэффективный поточечный тест видимости, так как для данной точки наблюдения количественная невидимость вещественного ребра может измениться только в тех случаях, когда это ребро уходит за контурное ребро или выходит из-за него. В терминах проекций на картинную плоскость эти соображения можно сформулировать следующим образом: количественная невидимость проекции вещественного ребра может измениться только в точке пересечения с каким-либо контурным ребром. В такой точке количественная невидимость увеличивается на 1, если вещественное ребро уходит за контурное, и уменьшается на 1, если вещественное ребро выходит из-за контурного. Таким образом, вещественное ребро разбивается на отрезки с различными значениями количественной невидимости; при этом различие между двумя соседними отрезками равняется 1. Тестовой прямой называется вещественная прямая, проходящая через тело

и точку наблюдения. Количественная невидимость может измениться при пересечении скрытого контурного ребра, образуемого внутренним углом невыпуклого тела.

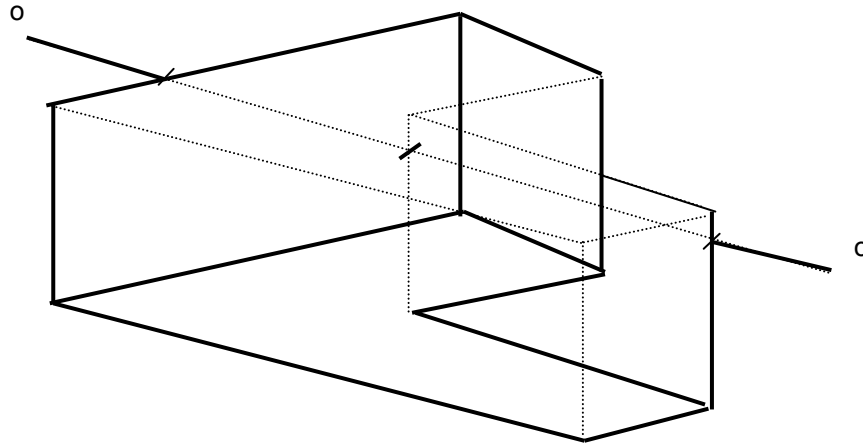


Рис. 7.2.1. Определение количественной невидимости при прохождении тестовой прямой за телом.

Подробнее о вычислении количественной невидимости вещественного ребра. Сначала вычисляется количественная невидимость начальной точки ребра. Для этого через начальную точку и точку наблюдения  $O$  проводится вещественная прямая и находятся пересечения этой прямой со всеми гранями. Возможны следующие варианты.

1) Плоскость, проходящая через точки  $P_i = (x_i, y_i, z_i)$  и  $P_j = (x_j, y_j, z_j)$  и  $P_k = (x_k, y_k, z_k)$ . Точки  $P_i P_j P_k$  не коллинеарны и удовлетворяют уравнению  $Ax + By + Cz + D = 0$ .

2) Прямая, проходящая через точки  $P_1 = (x_1, y_1, z_1)$  и  $P_2 = (x_2, y_2, z_2)$  Прямая может быть представлена уравнением

$$\frac{x - x_1}{\cos \alpha} = \frac{y - y_1}{\cos \beta} = \frac{z - z_1}{\cos \gamma},$$

где :

$$\cos \alpha = \frac{x_2 - x_1}{d}, \quad \cos \beta = \frac{y_2 - y_1}{d}, \quad \cos \gamma = \frac{z_2 - z_1}{d},$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}.$$

Тогда:

а) Если  $A \cos \alpha + B \cos \beta + C \cos \gamma = 0$ , то прямая параллельна плоскости.

б) Если  $A \cos \alpha + B \cos \beta + C \cos \gamma = 0$  и  $Ax_1 + By_1 + Cz_1 = 0$ , то прямая лежит в плоскости. В остальных случаях имеется точка пересечения :

$$x_i = x_1 - t \cos \alpha,$$

$$y_i = y_1 - t \cos \beta,$$

$$z_i = z_1 - t \cos \gamma,$$

$$\text{где : } t = \frac{Ax_1 + By_1 + Cz_1 + D}{A \cos \alpha + B \cos \beta + C \cos \gamma}.$$

Из всех полученных таким образом точек, выделяются точки, лежащие на отрезке вещественной прямой, соединяющем точку наблюдения с начальной точкой. Функция, реализующая тест глубины, применяется к двум перекрывающимся графическим элементам и определяет, какой из элементов заслоняет другой. Сравниваемыми элементами могут быть точка и грань или две грани. «Глубиной» называют расстояние между элементом и картинной плоскостью или между элементом и точкой наблюдения. Таким образом, глубина относится к координате элемента. Следовательно, тест глубины обычно заключается в сравнении координат двух элементов.

Тест применяется к элементам объектного пространства. Он сопоставляет два графических элемента (грань и точку) и определяет, заслоняет ли грань точку. Для этого находится точка, в которой грань пересекается с «линией визирования», т. е. линией, проходящей через проверяемую точку и точку наблюдения. Затем вычисляется расстояние от точки наблюдения до проверяемой точки и точки пересечения. Пусть  $d_i$  - расстояние от точки наблюдения до проверяемой точки, а  $d_p$  - расстояние от точки наблюдения до точки пересечения. Тогда проверяемая точка закрывается гранью, если  $d_i > d_p$ , и не закрывается в противном случае. При этом под-

разумеается наличие конечной точки наблюдения, т. е. используется центральное проектирование.

Для каждой выделенной точки находится соответствующая ей грань. Число найденных граней является количественной невидимостью начальной точки,

Тест принадлежности применяется к объектам, находящимся в картинной плоскости; он проверяет, лежит ли точка  $P$  внутри многоугольника  $F$  (который обычно является проекцией грани объекта). В элементарной геометрии имеются две хорошо известные процедуры, позволяющие выполнить эту проверку.

Вычисление суммы углов. Пусть  $F = (p_1, p_2, \dots, p_n)$  - многоугольник в картинной плоскости с вершинами  $p_i = (x_i, y_i), i = 1, \dots, n, p_n = p_1$ ,  $p_t$  - точка, для которой необходимо определить, принадлежит ли она  $F$ ;  $p_t p_i$  - отрезок, соединяющий  $p_t$  и  $p_i$ ;  $p_t$  соединена такими отрезками со всеми вершинами  $F$ ;  $\alpha_i$  - угол между  $p_t p_i$  и  $p_t p_{i+1}$ . Тогда точка  $p_t$  находится вне  $F$ , если  $\sum_i \alpha_i = 0$  и внутри  $F$ , если  $\sum_i \alpha_i = 2\pi$ .

Подсчет числа пересечений. Обозначим через  $R$  луч, начинающийся в точке  $p_t$  и проходящий через многоугольник  $F$ , но не содержащий ни одной вершины многоугольника. Тогда точка  $p_t$  находится внутри многоугольника  $F$ , если число пересечений  $R$  и  $F$  нечетно, и  $p_t$  находится вне  $F$ , если число пересечений четно.

Аппелем предлагается метод нахождения пересечений вещественных и контурных ребер

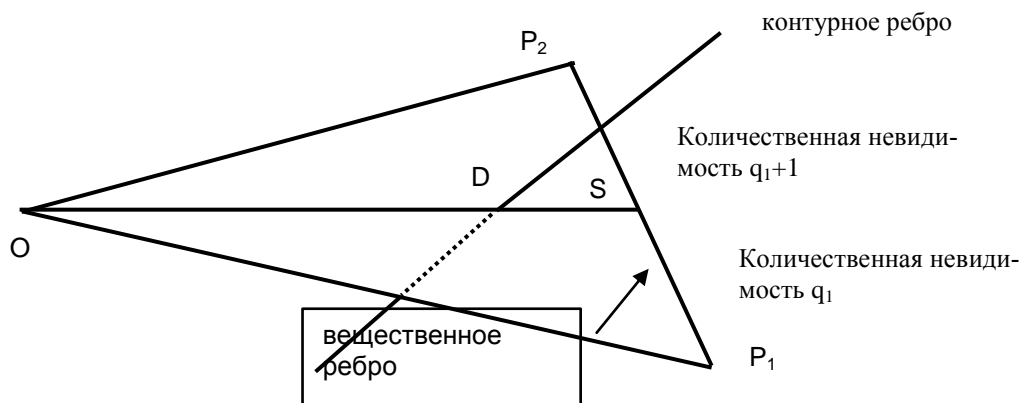


Рис. 7.2.2. Вычисление количественной невидимости вещественного ребра.

бер. Для этого образуется треугольник, вершинами которого являются точка наблюдения и концы вещественного ребра.

Контурное ребро изменяет количественную невидимость вещественного ребра только в том случае, когда оно протыкает этот треугольник, т. е. когда точка  $D$ , которая представляет собой точку пересечения контурного ребра с плоскостью, содержащей треугольник, лежит внутри треугольника. Если такое пересечение существует, то количественная невидимость увеличивается на 1 при положительном знаке векторного произведения контурного и вещественного ребер и уменьшается на 1 при отрицательном знаке. Точка изменения невидимости определяется как точка пересечения прямой  $OD$  с вещественным ребром (рис. 7.2.1). Таким образом, если  $q_1$  представляет собой количественную невидимость точки  $P_1$  вещественного ребра, то весь отрезок  $P_1S$  имеет это же значение количественной невидимости, в то время как соответствующее значение для отрезка  $SP_2$  равняется  $q_1 + 1$ .

### 7.3. Метод приоритетов Энкарнако.

Метод приоритетов, в том виде как он был первоначально разработан Энкарнако, Гросскопфом и др., применяется к многогранникам. Оперирует в **картинной плоскости**. Он применим и к невыпуклым объектам, т. е. объектам с вогнутыми гранями, отверстиями, разрезами и т. д. Такая возможность обеспечивается введением в алгоритм этапа предварительной обработки, на котором выполняется «триангуляция», т. е. декомпозиция всех граней на треугольники. Во время декомпозиции вводится некоторое число дополнительных ребер, помечаемых как вспомогательные. Таким образом, собственно алгоритм удаления невидимых линий оперирует с множеством треугольников, т. е. с множеством плоских, выпуклых граней самого простого типа из всех возможных. Треугольники определяются своими вершинами (тем самым неявно задаются и их ребра). Упрощение объекта с помощью триангуляции по-

могает минимизировать время выполнения алгоритма и требуемую память; дополнительным выигрышем является возможность обработки невыпуклых тел.

Для случая прямоугольной области приемы разбиения ее на элементарные прямоугольники и прямоугольные треугольники являются наиболее естественными и приводят к регулярным сеткам. Если получаемая точность недостаточна, то можно детализировать сетку, разбивая каждый прямоугольный треугольник еще на четыре (рис.1.).

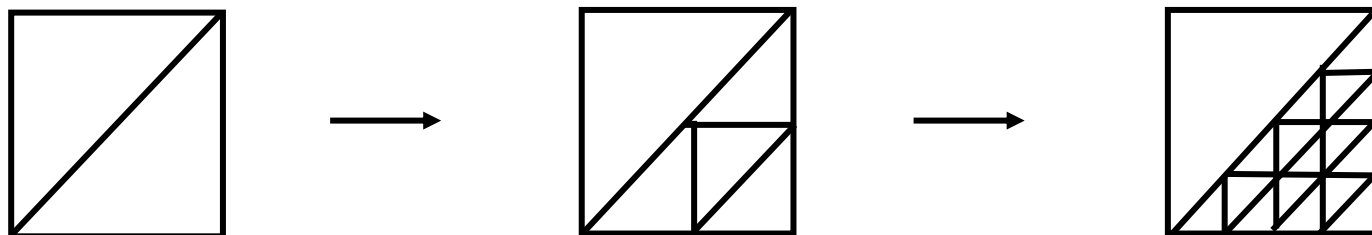
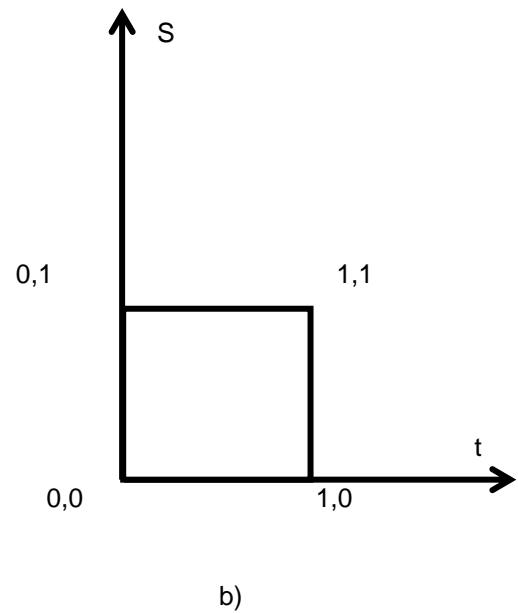
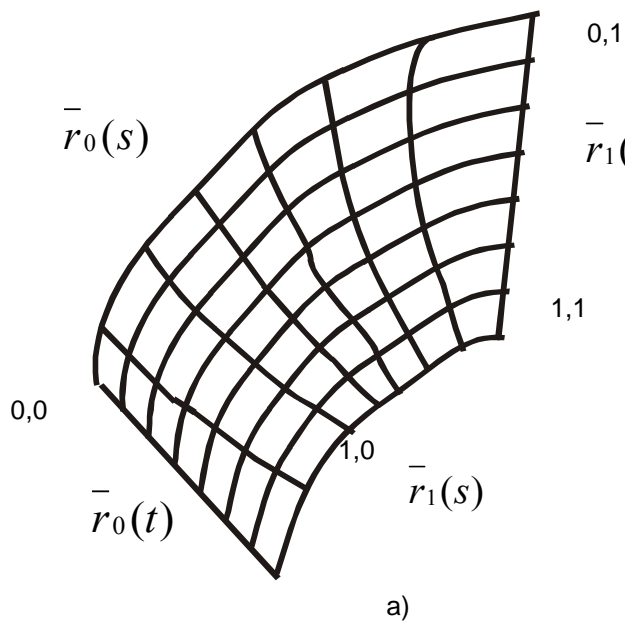


Рис.7.3.1.

В ряде случаев точность повышается при смене диагоналей, разбивающих прямоугольник на два треугольника. Сгущая сетку, руководствуются правилом: дальнейшее разбиение данного участка области производится только в том случае, если предшествующее привело к заметному повышению точности. В результате, начиная с некоторого момента, уточняется триангуляция только некоторых участков области. Теряется регулярность сетки, но зато существенно сокращается общий объем вычислений.

Если область отличается от прямоугольника, то указанное разбиение также может быть использовано, но при этом, как правило, для предотвращения потери точности вблизи границы требуется дополнительная триангуляция, учитывающая особенности последней.

Употребителен и другой подход. Данная область с криволинейной границей (или отдельные ее части) с помощью подходящего преобразования отображается в единичный квадрат, который разбивается на элементарные прямоугольники или прямоугольные треугольники. Производя обратное преобразование, получают на исходной области криволинейную сетку, хорошо учитывающую геометрию области. Например, если задана область, изображенная на рис. 2а, то одно из возможных преобразований, переводящих единичный квадрат в область  $D$ , может быть записано следующим образом.



Пусть вектор  $\varphi(t) = (-1, \varphi_0(t), \varphi_1(t))$ ,  $\varphi^T(s)$  — транспонированный вектор

$\varphi(s)$ ,  $\varphi_0(t) = 1 - t$ ,  $\varphi_1(t) = t$ . Через  $\bar{r}(t, s)$  обозначен радиус-вектор произвольной точки в области  $D$

относительно прямоугольной системы координат, и пусть точки  $\bar{r}_{ij}$ ,  $i, j = 0, 1$ , — вершины криволинейного четырехугольника  $D$ , отвечающие соответственно вершинам единичного квадрата  $(i, j)$ ,  $i, j = 0, 1$ . Нужное преобразование будет иметь вид

$$\bar{r}(t, s) = \varphi(t)F(t, s)\varphi^T(s),$$

где  $F(t, s)$  — матрица:

$$F(t, s) = \begin{pmatrix} 0 & \bar{r}_0(t) & \bar{r}_1(t) \\ \bar{r}_0(s) & \bar{r}_{00} & \bar{r}_{01} \\ \bar{r}_1(s) & \bar{r}_{10} & \bar{r}_{11} \end{pmatrix}$$

Основной алгоритм удаления невидимых поверхностей разбивается на три этапа:

- 1) предварительное упорядочение,
- 2) приписывание приоритетов и
- 3) определение видимости.

Предварительное упорядочение. На этом этапе выделяются все треугольники, перпендикулярные плоскости ху. Они не будут рассматриваться на следующем этапе, поскольку не могут заслонять другие грани. Грань перпендикулярна картинной плоскости, если проекции ее вершин коллинеарны.

Приписывание приоритетов. Пусть задан пространственный объект, расчлененный на множество треугольных граней,  $S = \{s_1, K, s_m\}$ . Пусть  $pr(s_i)$  - проекция грани  $s_i$  полученная с помощью ортогонального проецирования  $\bar{\Lambda} : S \rightarrow T, T = \{t_i / t_i = pr(s_i)\}$ . Предположим, что треугольные грани в объектном пространстве, так же как и соответствующие им проекции в картинной плоскости, определяются координатами своих вершин и для облегчения идентификации пронумерованы. На первом шаге метода приоритетов формируется множество упорядоченных пар:

$$(t_i, T_i), \quad i = 1, K, M,$$

где  $t_i \in T$  - треугольник (проекция треугольной грани  $s_i$ ) и  $T_i \subseteq T$  - множество треугольников, имеющих больший приоритет, чем  $t_i$ :

$$T_i = \{t_j \in T / t_i \leq t_j\}.$$

Каждая пара  $(t_i, T_i)$ ,  $i = 1, K, M$ , представляет собой упорядоченное по неубыванию множества пар треугольников. Первой компонентой каждой пары является  $t_i$ , а вторыми компонентами - элементы  $T_j$ . Пары  $(t_i, T_i)$  формируются с помощью теста глубины. Для этого вначале применяется минимаксный тест, с помощью которого определяется, какой из объектов  $t_i$ , может пересекаться с другими объектами  $t_j$ . «Изолированные» объекты  $t_i$  исключаются из рассмотрения, поскольку они встречаются только в парах  $(t_i, \emptyset)$  ( $\emptyset$  - обозначает пустое множество). Для остальных  $L \leq M$  объектов должен быть определен приоритет, что требует  $L(L-1)/2$  повторений теста глубины.



Проверка перекрытия двух выпуклых многоугольников в плоскости изображения. Заданы два выпуклых многоугольника в плоскости изображения (они могут быть проекциями двух граней трехмерных объектов). Каждый многоугольник определяется множеством своих вершин, т. е. если мы обозначим многоугольники как

$$F_1 = P_{1,1}, P_{1,2}, \dots, P_{1,m} \text{ и } F_2 = P_{2,1}, P_{2,2}, \dots, P_{2,m}$$

где  $P_{i,j} = (x_{i,j}, y_{i,j})$ .

Многоугольники  $F_1$  и  $F_2$  не перекрываются друг с другом, если выполняется следующее минимаксное условие:

$$j \in [1, m] \quad i \in [1, n]: \max(x_{1,j}) < \min(x_{2,k}) \text{ или } \max(x_{2,k}) < \min(x_{1,j}) \text{ или} \\ \max(y_{1,j}) < \min(y_{2,k}) \text{ или } \max(y_{2,k}) < \min(y_{1,j}).$$

Обычно задача не сводится только к определению факта перекрытия двух многоугольников, но требуется также найти точки пересечения этих двух граней. Данную задачу можно решить, если попытаться найти точку пересечения ребер многоугольников или проверить, не содержатся ли вершины одного многоугольника внутри другого, и наоборот. Ясно, что существует по крайней мере две точки пересечения двух перекрывающихся многоугольников. В противном случае либо один из них целиком содержится в другом, либо они не перекрываются. Как правило, предпочтительнее начинать с теста пересечения, а не с теста принадлежности. Это объясняется следующими причинами:

1. Вероятность пересечения ребер гораздо больше, чем вероятность полной вложенности или отсутствия общих точек.

2. Отношение инцидентности симметрично, а отношение принадлежности антисимметрично. Поэтому тест принадлежности должен быть выполнен для обоих многоугольников, а пересечение ребер достаточно проверить для одного из них.

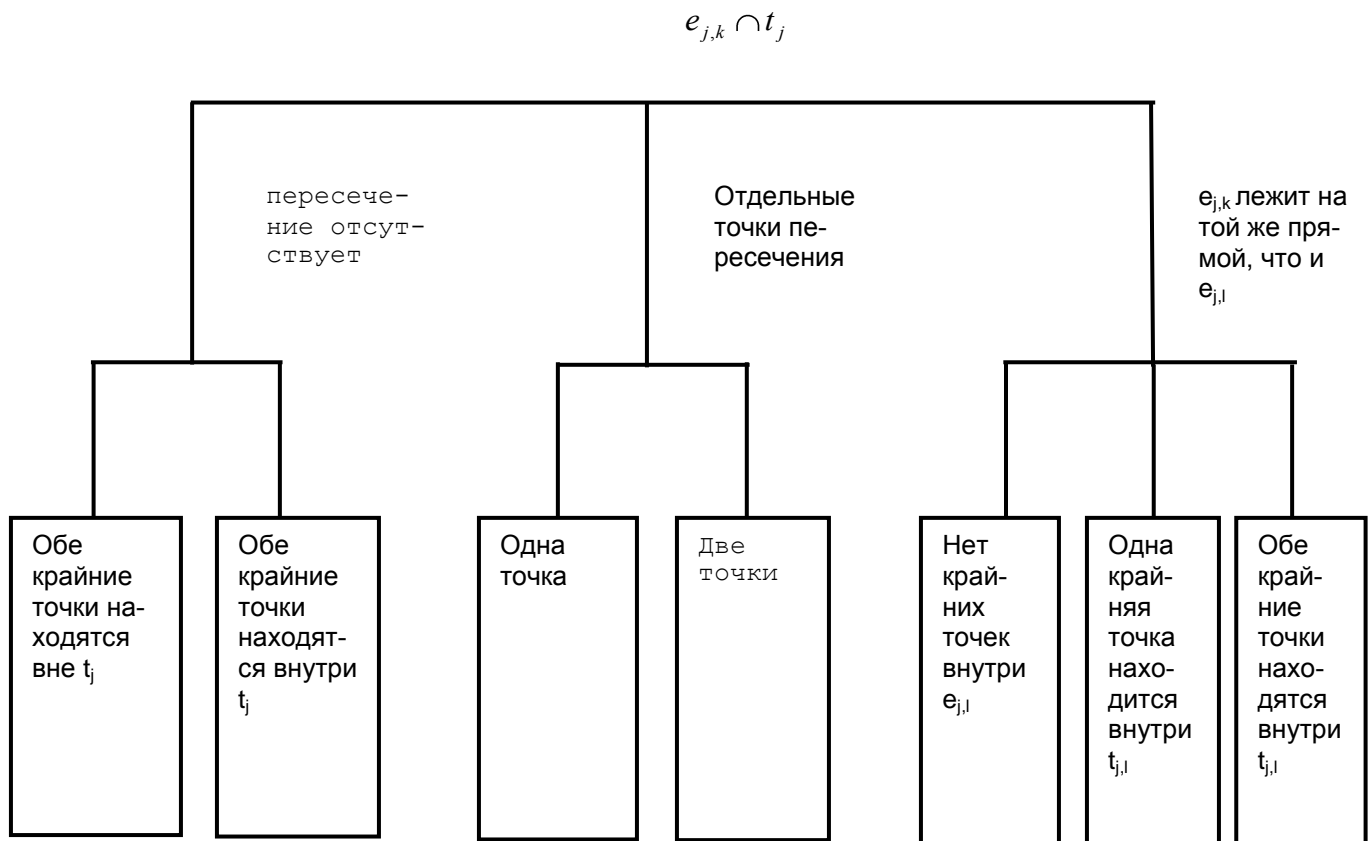
3. Тест принадлежности включает вычисление пересечений. Если ни один из тестов не дает точку пересечения, то это означает, что пересечение пусто. Желательно, однако еще до выполнения тестов проверить, возможно ли вообще пересечение двух конкретных многоугольников. Для предварительной проверки можно использовать минимаксное условие, описанное выше. Если вероятность того, что пересечение окажется пусто, гораздо больше вероятности непустого пересечения, то предварительная проверка поможет уменьшить время выполнения программы. Но предварительная проверка может быть оправдана даже в тех случаях, когда вероятность пустого пересечения намного меньше вероятности непустого пересечения, поскольку простое минимаксное условие требует выполнения лишь немногих логических операций и операций отношения, в то время как тесты пересечения и принадлежности требуют выполнения большого числа громоздких арифметических вычислений. Время их выполнения может превышать время проверки минимаксного условия на несколько порядков.

Для нахождения тестовых точек (точек пересечения двух треугольников), необходимых для определения приоритетов, сначала отыскиваются точки пересечения ребер, а если такие точки отсутствуют, проверяется, нет ли вложенности одного треугольника в другой. Вследствие триангуляции циклическое перекрытие невозможно, однако треугольные грани могут проникать друг в друга. Такие случаи требуют специального рассмотрения. Проникание может иметь место только тогда, когда проекции двух граней имеют непустое пересечение. Обнаружить проникание можно, выполнив приоритетный тест для всех точек пересечения ребер проекций. Если результат теста постоянен для всех точек, то проникания нет.

Определение видимости. В каждой паре  $(t_i, T_i)$   $T_i$  представляет множество треугольников, покрывающих, хотя бы частично, треугольник  $t_i$ . Если  $T_i$  пусто, то ребро  $t_i$  полностью видимо (за исключением вспомогательных ребер). Для непустого  $T_i$  необходимо опре-

делить степень покрытия  $t_i$  треугольниками из  $T_i$ . Следовательно, достаточно описать, как исследуется видимость некоторого ребра.

Пусть  $e_{i,k}$   $k \in 1, 2, 3$  - ребра  $t_i$ . Видимость  $e_{i,k}$  по отношению к треугольнику  $t_j \cdot > t_i$  можно определить, найдя точки пересечения  $e_{i,k}$  с  $t_j$  и проверив, где находятся конечные точки  $e_{i,k}$  - внутри или вне  $t_j$ . Отрезок  $e_{i,k}$  пересекающий  $t_j$ , невидим; отрезок  $e_{i,k}$ , не пересекающийся с  $t_j$  потенциально видим. На рис. 7.3.3 иллюстрируются возможные случаи.



Тест принадлежности применяется к объектам, находящимся в картинной плоскости; он проверяет, лежит ли точка  $P$  внутри многоугольника  $F$  (который обычно является проекцией грани объекта). В элементарной геометрии имеются две хорошо известные процедуры, позволяющие выполнить эту проверку.

Вычисление суммы углов. Пусть  $F = (p_1, p_2, \dots, p_n)$  - многоугольник в картинной плоскости с вершинами  $p_i = (x_i, y_i), i = 1, \dots, n, p_n = p_1$ ,  $p_t$  - точка, для которой необходимо определить, принадлежит ли она  $F$ ;  $p_t p_i$  - отрезок, соединяющий  $p_t$  и  $p_i$ ;  $p_t$  соединена такими отрезками со всеми вершинами  $F$ ;  $\alpha_i$  - угол между  $p_t p_i$  и  $p_t p_{i+1}$ . Тогда точка  $p_t$  находится вне  $F$ , если  $\sum_i \alpha_i = 0$  и внутри  $F$ , если  $\sum_i \alpha_i = 2\pi$ .

Подсчет числа пересечений. Обозначим через  $R$  луч, начинающийся в точке  $p_t$  и проходящий через многоугольник  $F$ , но не содержащий ни одной вершины многоугольника. Тогда точка  $p_t$  находится внутри многоугольника  $F$ , если число пересечений  $R$  и  $F$  нечетно, и  $p_t$  находится вне  $F$ , если число пересечений четно.

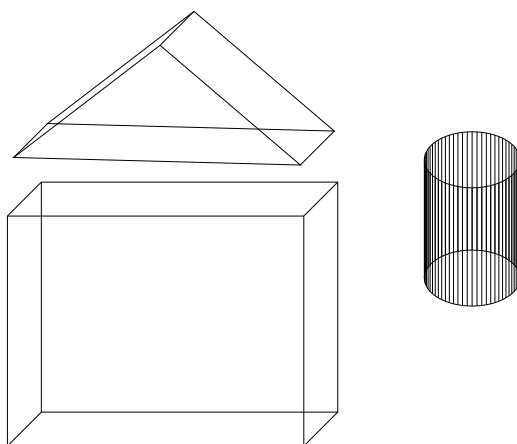
После применения тестов видимости мы получаем в качестве видимых объектов не треугольники (т. е. замкнутые ломаные), а отдельные ребра. Следовательно, на входе процедуры имеется список треугольных граней, каждая из которых представляется тремя тройками координат, а на выходе - список видимых отрезков прямых, каждый из которых определяется двумя парами координат.

#### 7.4.Алгоритм Робертса

Алгоритм работает в **объектном пространстве**. Модификации алгоритма, использующие предварительную пространственную сортировку вдоль оси  $Z$  и габаритные или минимаксные тесты, позволяют добиться почти линейной зависимости увеличения объема вычислений от роста числа объектов.

Задача удаления невидимых линий решается, в данном случае, для объектов, составленных из выпуклых многогранников. Любой замкнутый объект с плоскими гранями можно представить в виде набора выпуклых многогранников (рис. I), т.е. в виде наборов плоскостей, образующих грани данных многогранников. Объем вычислений в алгоритмах удаления невидимых линий и поверхностей увеличивается с ростом числа многоугольников, поэтому для описания поверхностей объектов необходимо использовать многоугольники с более чем тремя сторонами.

Рис. 7.4.I. Представление объекта в виде набора выпуклых многогранников.



Для реализации алгоритма вначале необходимо представить объекты визуализации в виде наборов многогранников, каждый из которых задан уравнениями плоскостей своих граней. Каждая плоскость многогранника описывается уравнением

$$aX + bY + cZ + d = 0$$

где  $X$ ,  $Y$  и  $Z$  - мировые координаты. В матричной форме это уравнение имеет вид

$$\begin{bmatrix} X & Y & Z & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}^T = 0$$

где  $P^T = \begin{bmatrix} a & b & c & d \end{bmatrix}$  - транспонированный вектор коэффициентов плоскостей. Векторы плоско-

стей составляют матрицу порядка  $(4-n)$ , называемую обобщенной матрицей объекта:

$$T = \begin{bmatrix} a_1 & a_2 & K & a_n \\ b_1 & b_2 & K & b_n \\ c_1 & c_2 & K & c_n \\ d_1 & d_2 & K & d_n \end{bmatrix},$$

где каждый столбец содержит коэффициенты одной плоскости.

Любая точка на плоскости однозначно определяется двумя координатами, а точка в пространстве в однородных координатах описывается вектором

$$[S] = [X \quad Y \quad Z \quad 1]$$

Кроме того, если точка лежит на плоскости, то скалярное произведение

$$[S][P]^T = 0$$

Если точка не лежит на плоскости, то по знаку этого скалярного произведения можно судить, по какую сторону от плоскости расположена точка. При разработке алгоритма предполагалось, что точки, лежащие внутри тела, дают положительное скалярное произведение, т.е.

$$SV \geq 0$$

Для формирования матрицы объекта, состоящей из коэффициентов уравнений плоскостей используется общеизвестный из аналитической геометрии принцип: плоскость можно определить по трем неколлинеарным точкам (хотя уравнение плоскости содержит четыре неизвестных коэффициента, его можно нормировать так, чтобы  $d=1$ ). Другой способ основан на вычислении вектора нормали к плоскости:

$$n = ai + bj + ik$$

где  $i, j, k$  - единичные векторы осей  $X, Y$  и  $Z$  соответственно. Тогда уравнение плоскости описывается формулой  $aX + bY + cZ + d = 0$ , где коэффициент  $d$  вычисляют с использованием произвольной точки на плоскости. Например, если эта точка имеет координаты  $(x_1, y_1, z_1)$ , то

$$d = -(ax_1 + by_1 + cz_1)$$

Для определения коэффициентов уравнений плоскостей многогранников синтезируемого объекта поступают следующим образом. Определяют нормали к каждой вершине многоугольника с помощью векторного произведения прилежащих ребер, а затем усреднении

результатов. Для определения нормали к плоскости вычисляют нормали к каждой вершине многоугольника с помощью векторного произведения прилежащих ребер. Например, для треугольника получают:

$$A(x_1, y_1, z_1), \quad B(x_2, y_2, z_2), \quad C(x_3, y_3, z_3);$$

$$n_A = \begin{bmatrix} i & j & k \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{bmatrix} \quad \text{и т. д.}$$

а затем усредняют результаты.

Таким образом, коэффициенты плоскости вычисляют по следующей системе уравнений:

$$a = \sum_{i=1}^3 (y_i - y_j)(z_i + z_j);$$

$$b = \sum_{i=1}^3 (z_i - z_j)(x_i + x_j);$$

$$c = \sum_{i=1}^3 (x_i - x_j)(y_i + y_j).$$

где  $j=i+1$ , если  $i=3$ , то  $j=1$ , а коэффициент  $d$  определяют с использованием любой точки на плоскости.

Перед реализацией алгоритма удаления невидимых линий и поверхностей для получения нужного положения синтезируемой сцены обычно проводят трехмерное видовое преобразование и уравнения плоскостей преобразуют из мировых координат в координаты устройства отображения (например, экранную систему координат). Новые матрицы объектов сцены вычисляют с помощью исходных матриц объектов визуализации и преобразованных вершин или других точек объектов.

Пусть  $[PR]$  - матрица размером  $4 \times 4$  преобразования систем координат и перспективного преобразования (матрица видового преобразования в однородных координатах), а  $[CB]$  - матрица однородных координат, определяющая вершины объекта в исходном положении.

Тогда координаты преобразованных вершин можно описать с помощью соотношения

$$[CBPR] = [CB][PR]$$

Уравнения исходных плоскостей, ограничивающих объект, получают путем подстановки трех неколлинеарных точек каждой грани многогранников объекта в соотношение  $[X \ Y \ Z \ 1][P]^T = 0$ :

$$[CB][T] = [N]$$

где  $[N]$  - нулевая матрица.

Уравнения преобразованных плоскостей объекта определяют аналогичным образом:

$$[CBPR][TE] = [N],$$

где  $[TE]$  - преобразованная матрица объекта. Приравнявая левые части двух последних соотношений, получаем

$$[CBPR][TE] = [CB][T].$$

Затем приводим это уравнение к виду

$$[TE] = [PR]^{-1}[T]$$

Таким образом, преобразованную матрицу объекта можно получить умножением исходной матрицы слева на обратную матрицу видового преобразования.

После формирования матрицы объекта и видового преобразования определяют невидимые грани и ребра объекта. Предполагается, что наблюдатель находится в бесконечности на положительной полуоси  $Z$  и смотрит в начало координат. Вектор такого направления в однородных координатах запишем следующим образом:

$$[V] = [0 \ 0 \ 1 \ 0].$$

С помощью вектора  $[V]$  можно также описать любую точку, лежащую на плоскости  $Z = \infty$ , т.е. любую точку типа  $(X, Y, -\infty)$ . Тогда, если скалярное произведение  $[V]$  на столбец матрицы объекта соответствующий какой-либо плоскости, отрицательно, то вектор  $[V]$



расположен с отрицательной стороны от этой плоскости. И, следовательно, эта плоскость невидима из любой точки наблюдения, лежащей в плоскости  $Z = \infty$ . Такие плоскости называют нелицевыми.

Таким образом, если скалярное произведение  $VT < 0$ , то плоскости являются нелицевыми. После отбрасывания нелицевых плоскостей число граней для объектов описанных выпуклыми многогранниками, сокращается примерно вдвое.

Удаляют нелицевые плоскости, вычисляя нормали к поверхности для каждого многоугольника. Если нормаль отрицательна, то она направлена в сторону от наблюдателя, и, следовательно, данный многоугольник не виден.

Затем определяют и удаляют из списка ребер нелицевые ребра, которые образуются в результате пересечения пары нелицевых плоскостей. Если одна или обе грани, образующие ребро, обращены своей внешней поверхностью к наблюдателю, то это ребро считается видимым.

После удаления нелицевых граней и ребер для каждого объекта выполняют самую трудоемкую часть алгоритма: проверяют каждое оставшееся ребро на экранирование другими объектами сцены. Вначале проверки рекомендуется сортировать объекты по  $Z$ -координатам их вершин и габаритным тестам прямоугольной объемлющей оболочки. Это позволяет удалять группы невидимых ребер и объектов. Действительно, если все тела в сцене отсортированы по значениям  $Z$  ближайших вершин и расстоянию до наблюдателя, то никакой объект, ближайшая точка которого расположена дальше от наблюдателя, чем самая удаленная из концевых точек ребра (отрезка), не может закрывать его. Кроме того, ни один из объектов, прямоугольная оболочка которых располагается полностью сверху, снизу, справа или слева от отрезка, и может также закрывать его. После выполнения  $Z$ -сортировки и габаритного теста число объектов, с которыми нужно сравнивать каждый отрезок, существенно уменьшается.

Пусть отрезок  $(R_1 R_2)$  задан в параметрической форме:

$$R(t) = R_1 + (R_2 - R_1)t, \quad 0 < t < 1,$$

или

$$\mathbf{0} = \mathbf{S} + \mathbf{d}t,$$

где  $\mathbf{0}$  - вектор точки на отрезке»  $\mathbf{S}$  - вектор координат начальной точки;  $\mathbf{d}$  - вектор направления отрезка  $R_2 - R_1$ ,  $t$  - параметр.

Требуется определить: существуют ли значения  $t$ , при которых данный отрезок невидим?

Для этого соединим произвольную точку  $t$ , заданную текущим значением параметра, с точкой наблюдения. Если эта прямая не пересекает ни одного многогранника, ни одного объекта, то точка видима. Если же прямая пересекает какой - либо многогранник, то исследуемая точка экранируется этим многогранником. Прямую от точки  $R(t)$  до точки наблюдения представим также в параметрической форме:

$$\mathbf{0}(f, t) = \mathbf{U} = \mathbf{0} + \mathbf{V}f = \mathbf{S} + \mathbf{d}t + \mathbf{V}f, \quad 0 \leq t \leq 1, \quad f \geq 0,$$

где значение  $t$  задает точку на отрезке  $R(t)$ , а  $f$  точку на отрезке, проведенном от точки  $R(t)$  до точки наблюдения. Значение  $f$  всегда положительно, так как объекты, экранирующие  $R(t)$  могут находиться, только в той части данной плоскости, которая заключена между исследуемым отрезком  $R(t)$  и точкой наблюдения.

Скалярное произведение любой точки, расположенной внутри объекта и матрицы объекта, положительно (это утверждение справедливо и для преобразованной матрицы объекта). Точка, находящаяся внутри объекта, невидима. Следовательно, для проверки на экранирование вектор текущей точки отрезка умножают поочередно на матрицу каждого объекта и определяют положительное решение, соответствующее прохождению отрезка внутри объекта:

$$\mathbf{H} = \mathbf{U}[TE] = \mathbf{S}[TE] + \mathbf{d}t[TE] + \mathbf{V}f[TE] > 0$$

Значения  $t$  и  $f$ , для которых все значения вектора  $\mathbf{H}$  положительны, соответствуют невидимой части отрезка.

Пусть

$$\mathbf{p} = \mathbf{S}[TE], \quad \mathbf{q} = \mathbf{d}[TE], \quad \mathbf{w} = \mathbf{V}[TE],$$

Тогда условие  $\mathbf{H} = \mathbf{U}[TE] = \mathbf{S}[TE] + \mathbf{d}t[TE] + \mathbf{V}f[TE] > 0$  будет иметь вид

$$\mathbf{H}_j = \mathbf{p}_j + t\mathbf{q}_j + f\mathbf{w}_j > 0,$$

где  $j$  - номер столбца в матрице объекта;  $\mathbf{w}$  - вектор третьей строки матрицы объекта.

Последние условия должны выполняться для всех плоскостей, ограничивающих объект, т.е. для всех значений  $j$ . Граница между видимой и невидимой частями отрезка определяется условием  $\mathbf{H}_j = 0$ . Решая эту систему уравнений, находят все значения  $t$  и  $f$ , при которых изменяется значение видимости отрезка или части отрезка (число возможных решений при  $n$  плоскостях равно  $n(n-1)/2$ ). Затем каждое решение в интервалах  $0 \leq t \leq 1, \quad f \geq 0$  подставляют во все остальные неравенства системы  $\mathbf{H} = \mathbf{U}[TE] = \mathbf{S}[TE] + \mathbf{d}t[TE] + \mathbf{V}f[TE] > 0$  для проверки того, что условие  $H_j > 0$  выполнено. Так определяют минимальное значение параметра  $t$  среди максимальных ( $t_{\min \max}$ ) и максимальное среди минимальных ( $t_{\max \min}$ ). Подставляя эти значения в уравнение  $\mathbf{0} = \mathbf{S} + \mathbf{d}t$ , определяют видимые участки отрезка на интервалах  $[0, t_{\max \min}]$  и  $[0, t_{\min \max}]$ . Условие экранирования рёбер или отрезков ребер является простым следствием из классической задачи линейного программирования:

$$t_{\max \min} < t < t_{\min \max}$$

Решения, удовлетворяющие неравенствам  $\mathbf{H}_j > 0$ , могут существовать и за пределами области, ограниченной условиями:

$$0 \leq t \leq 1, \quad f \geq 0$$

Поэтому к системе (4) необходимо добавить три уравнения, описывающие эти границы:

$$t = 0, \quad t - 1 = 0, \quad f = 0.$$

В том случае число решений увеличится до  $(n-2)(n-3)/2$ . Для ускорения работы алгоритма перед определением  $t_{\min \max}$  и  $t_{\max \min}$  желательно установить не только нелицевые ребра, но и полностью видимые ребра. Отрезок будет видим, если оба его конца лежат между точкой наблюдения и какой-либо видимой плоскостью. При  $f=0$  значение  $U$  легко вычислять по соотношению  $\mathbf{0} = \mathbf{S} + \mathbf{d}t$ . Затем при  $t = 0$  и  $t = 1$  определяют концевые точки отрезка. Из соотношений  $\mathbf{H}_j = \mathbf{p}_j + t\mathbf{q}_j + f\mathbf{w}_j > 0$ , видно, что  $\mathbf{p}_j$  является скалярным произведением концевой точки отрезка и  $j$ -й плоскости при  $t = 0$ , а  $(\mathbf{p}_j + \mathbf{q}_j)$  является скалярным произведением другой концевой точки отрезка и  $j$ -й плоскости. В свою очередь,  $j$ -я плоскость, ограничивающая объект, видима, если  $\mathbf{w}_j \leq 0$ . Следовательно, если  $\mathbf{w}_j \leq 0$ ,  $\mathbf{p}_j \leq 0$ ,  $\mathbf{p}_j + \mathbf{q}_j \leq 0$ , то отрезок целиком видим, а оба его конца лежат либо на видимой плоскости, либо между этой плоскостью и точкой наблюдения.

Для полностью невидимых отрезков объекта отсутствуют тесты вследствие бесконечности плоскостей. Поэтому полностью невидимые отрезки определяют так же, как и частично невидимые (в этом случае невидимый отрезок будет простирается от  $t = 0$  и  $t = 1$ ).

После определения частично видимых или полностью невидимых отрезков определяют пары объектов, связанных отношением протыкания (в случае протыкания объектов сцены ищутся решения на границе  $f = 0$ ), и вычисляют отрезки,

Рис. 7.4.2. Образование структуры протыкания.



которые образуются при протыкании объектами друг друга. Эти отрезки проверяют на экранирование всеми прочими объектами сцены. Видимые отрезки образуют структуру протыкания (рис. 7.4.2).

Таким образом, алгоритм подразделяется на следующие этапы.

1. Определение коэффициентов уравнения плоскости каждой грани, проверка правильности знака уравнения и формирование матрицы объекта визуализации.
2. Проведение видового преобразования матрицы объекта, вычисление прямоугольной охватывающей оболочки объекта.
3. Определение нелицевых граней, удаление их из списка граней и соответствующих ребер - из списка ребер.
4. Определение списка других объектов синтезируемой сцены, которые могут быть экранированы данным объектом визуализации на основании сравнений охватывающих оболочек объектов.
5. Формирование списка протыканий на основании сравнений охватывающих оболочек объектов.
6. Определение невидимых отрезков или участков отрезков.
7. Формирование списка возможных отрезков, соединяющих точки протыкания, для пар объектов, связанных отношением протыкания.
8. Проверка видимости полученных отрезков по отношению ко всем объектам сцены в соответствии с этапами 3 и 6.
9. Визуализация изображения.

