

## Лекция 1.4. Алгоритмы отсечения

Отсечение — это процесс выделения некоторой части базы данных, описывающей изображение. Основное применение алгоритмов отсечения — это отбор той информации, которая необходима для визуализации конкретной сцены или вида, как части более обширной обстановки. Но, кроме этого, отсечение применяется в алгоритмах удаления невидимых линий и поверхностей, при построении теней, а также при формировании фактуры.

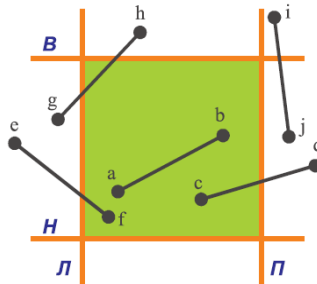
Алгоритмы отсечения подразделяются на дву- и трехмерные и применяются к каноническим (регулярным, изотетичным) и нерегулярным областям и объемам.

### Простой алгоритм определения полной видимости и тривиальной невидимости отрезка

Рассмотрим плоскую сцену и отсекающее окно регулярной формы.

Окно задается левым (Л), правым (П), верхним (В) и нижним (Н) двумерными ребрами. Регулярным отсекающим окном является прямоугольник, стороны которого параллельны осям координат объектного пространства или осям координат экрана.

Поскольку в обычных сценах необходимо отсекаль большое число отрезков или точек, то повышение эффективности (быстродействия) алгоритмов отсечения является актуальной задачей. Важно уметь быстро отбирать отрезки и точки, которые лежат целиком внутри или вне отсекающего окна.



Точки, лежащие внутри отсекающего окна, удовлетворяют условию:

$$x_{\text{л}} \leq x \leq x_{\text{п}} \text{ и } y_{\text{н}} \leq y \leq y_{\text{в}}.$$

Точки, лежащие на границе окна, считаются находящимися внутри него.

Отрезок является видимым, если обе его концевые точки лежат внутри окна (ab на рис.). Однако если оба конца отрезка лежат вне окна, то этот отрезок не обязательно лежит целиком вне окна (gh на рис.). Если же оба конца отрезка лежат справа, слева, выше или ниже окна, то этот отрезок целиком лежит вне окна, а значит, невидим (это условие не касается отрезка gh).

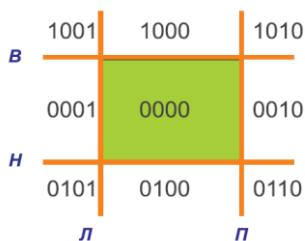
### Метод Д. Козна и А. Сазерленда

В этом методе для определения той из девяти областей, которой принадлежит конец ребра, вводится четырехразрядный (битовый) код. Коды этих областей показаны на рис. Крайний правый бит кода считается первым.

В соответствующий бит заносится 1 при выполнении следующих условий:

- для первого бита - если точка левее окна;
- для второго бита - если точка правее окна;
- для третьего бита - если точка ниже окна;
- для четвертого - если точка выше окна;

В противном случае в бит заносится ноль.



Отсюда, очевидно, следует, что если коды обоих концов ребра равны нулю, то обе эти точки лежат внутри окна, и отрезок видимый. Коды концевых точек можно использовать также и для тривиального отбрасывания полностью невидимых отрезков. Если побитовое логическое произведение концевых кодов отрезка не равно нулю, то отрезок полностью невидим и его можно отбросить тривиально. Однако если логическое произведение равно нулю, то отрезок может оказаться целиком или частично видимым или даже целиком невидимым. Поэтому для определения полной видимости необходимо проверять значения кодов обоих концов отрезка по отдельности.

### Вычисление пересечений отрезка со сторонами окна

Если прежде всего найдены целиком видимые и тривиально невидимые отрезки, то подпрограмме, вычисляющей пересечение отрезков, передаются только отрезки, которые, возможно, частично видимы, т.е. те, для которых результат логического умножения кодов их концевых точек равен нулю. Конечно же, эта подпрограмма должна правильно определять переданные ей (также) целиком невидимые отрезки.

Очевидно, что уравнение бесконечной прямой, проходящей через точки:

$P_1(x_1, y_1)$  и  $P_2(x_2, y_2)$ , имеет вид

$$y = m(x - x_1) + y_1 \text{ или } y = m(x - x_2) + y_2,$$

где  $m = (y_2 - y_1) / (x_2 - x_1)$  - это наклон данной прямой.

Точки пересечения этой прямой со сторонами окна имеют следующие координаты:

$$\text{с левой: } x_L, y = m(x_L - x_1) + y_1 \quad m \neq \infty$$

$$\text{с правой: } x_P, y = m(x_P - x_1) + y_1 \quad m \neq \infty$$

$$\text{с верхней: } y_B, x = x_1 + (1/m)(y_B - y_1) \quad m \neq 0$$

$$\text{с левой: } y_H, x = x_1 + (1/m)(y_H - y_1) \quad m \neq 0$$

Рассмотрим отсекающее окно и отрезки, изображенные на рис.

$$P_1(-3/2, 1/6) \text{ до } P_2(1/2, 3/2) \text{ равен}$$

$$m = (y_2 - y_1) / (x_2 - x_1) = (3/2 - 1/6) / [1/2 - (-3/2)] = 2/3.$$

Его пересечения со сторонами окна таковы:

с левой:  $x = -1 \quad y = (2/3)[-1 - (-3/2)] + 1/6 = 1/2$

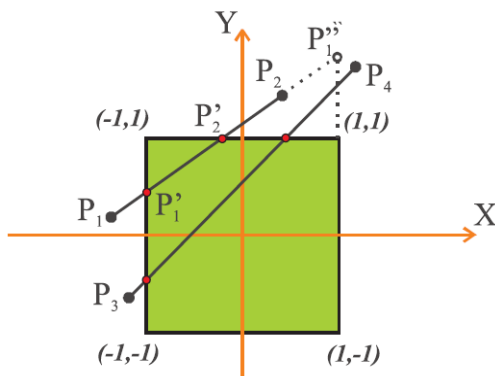
с правой:  $x = 1 \quad y = (2/3)[1 - (-3/2)] + 1/6 = 11/6$

(последнее число больше, чем  $x$ , и поэтому отвергается) - проверка корректности пересечения

с верхней:  $y = 1 \quad x = -3/2 + (3/2)[1 - 1/6] = -1/4$

с нижней:  $y = -1 \quad x = -3/2 + (3/2)[-1 - (1/6)] = -13/4$

(последнее число меньше, чем  $x$  и поэтому отвергается).



## Простой алгоритм двумерного отсечения

В целом простой алгоритм [двумерного](#) отсечения можно представить следующими основными шагами:

1. Вычисление кодов концевых точек отрезка. Занесение этих кодов в два массива M1 и M2 размерностью  $1 \times 4$  каждый.
2. Инициализация флажка видимости.
3. Инициализация наклона.
4. Проверка полной видимости отрезка (коды обоих концов равны нулю): суммирование элементов M1 и M2. Если сумма элементов M1 и сумма элементов M2 равны нулю, то вычерчивание отрезка.
5. Проверка случая тривиальной невидимости. Если логическое произведение соответствующих элементов матриц не равно нулю, то

флажок видимости устанавливается в «невидимость» и осуществляется переход к следующему отрезку.

6. Процедура отсечения отрезка каждой из сторон окна и получения точек пересечения с помощью проверки их принадлежности внутренней области окна. Эта последняя процедура реализуется по следующей схеме: сначала она применяется к отрезку  $PP_2$  (рис.) и получается отрезок  $P_1P_2$ , а затем — к отрезку  $P_1P_2$  и получается результирующий отрезок  $P_1P_2'$ .

### **Замечания:**

Для эффективной реализации данного алгоритма необходимо иметь в виду следующее:

1) Если наклон бесконечен, то отрезок параллелен левой и правой сторонам окна и надо искать его пересечение только с верхней и нижней сторонами.

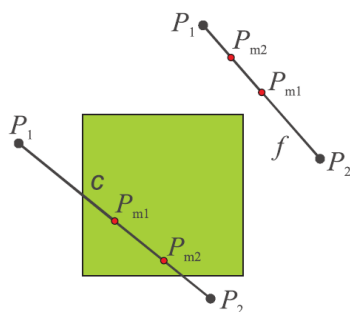
2) Если наклон равен нулю, то отрезок параллелен верхней и нижней сторонам окна, а искать его пересечения надо только с левой и правой сторонами.

3) Если код одного из концов отрезка равен нулю, то этот конец лежит внутри окна, и поэтому отрезок может пересечь только одну сторону окна.

### **Алгоритм разбиения средней точкой**

В простом алгоритме и в алгоритме Сазерленда-Коэна требуется вычислить пересечение отрезка со стороной окна. Алгоритм разбиения средней точкой позволяет избежать непосредственного вычисления. Этот алгоритм отсечения реализует двоичный поиск пересечения путем деления отрезка его средней точкой. Идея этого алгоритма была предложена Спруллом и Сазерлендом в 1968г. для аппаратной реализации. Программная реализация этого алгоритма медленнее, чем реализация предыдущего алгоритма, аппаратная — быстрее и эффективнее, поскольку можно использовать параллельную архитектуру. Кроме того, аппаратные сложения и деления на 2 очень быстры (деление на 2 аппаратно эквивалентно сдвигу каждого бита вправо).

В алгоритме используются коды концевых точек отрезка и проверки, выявляющие полную видимость отрезков. Отрезки  $s$  и  $f$  разбиваются на две равные части. Затем те же проверки применяются к каждой из половин до тех пор, пока не будет обнаружено пересечение со стороной окна или длина разделяемого отрезка не станет пренебрежимо малой, т.е. пока он не выродится в точку. После вырождения определяется видимость полученной точки. Максимальное число разбиений пропорционально точности задания координат концевых точек отрезка.



Рассмотрим отрезки  $s$  и  $f$ . Отрезок  $f$  не может быть тривиально отвергнут. Разбиение его средней точкой  $P_{m1}$  позволяет тривиально отвергнуть половину  $P_{m1}P_2$ . Однако половина  $P_{m1}P_1$  тоже пересекает диагональ окна, и ее нельзя отвергнуть тривиально. Далее проводится разбиение точкой  $P_{m2}$ , которое позволяет отвергнуть невидимый отрезок  $P_{m2}P_1$ . Разбиение оставшегося куска  $P_{m1}P_{m2}$ , продолжается до тех пор, пока не будет найдено пересечение этого отрезка с прямой, несущей правую сторону окна, с наперед заданной точностью. Затем исследуется обнаруженная точка и она оказывается невидимой. Следовательно, и весь отрезок невидим.

Разбиение отрезка  $s$  его средней точкой  $P_{m1}$  приводит к одинаковым результатам для обеих половин. Отложив отрезок  $P_{m1}P_1$  на потом, разобьем отрезок  $P_{m1}P_2$  точкой  $P_{m2}$ . Теперь отрезок  $P_{m1}P_{m2}$  полностью видим, а отрезок  $P_{m2}P_2$  видим частично. Точка  $P_{m2}$  запоминается как текущая видимая точка, которая наиболее удалена от  $P_1$ . А разбиение

отрезка  $P_{m2}P_2$  продолжается. Каждый раз, когда обнаруживается видимая средняя точка, она объявляется текущей наиболее удаленной от  $P$  до тех пор пока не будет обнаружено пересечение с нижней стороной окна с заранее заданной точностью. Это пересечение и будет объявлено самой удаленной от  $P_1$  видимой точкой. Затем точно так же обрабатывается отрезок  $P_{m1}P_1$ . Наиболее удаленной от  $P_2$  видимой точкой будет точка его пересечения с левой стороной окна.

### Алгоритм Кируса-Бека

Алгоритм применим для отсекаемого двумерного отрезка произвольным выпуклым окном. В АКБ для определения местоположения точки, принадлежащей отрезку, относительно окна (внутри, на границе или вне) используется вектор нормали и параметрическое представление отрезка. Параметрическое уравнение отрезка от  $P_1$  до  $P_2$  имеет вид

$$P(t) = P_1 + (P_2 - P_1) \cdot t, \quad 0 \leq t \leq 1, \quad (1)$$

где  $t$  — параметр.

Условие задает именно отрезок, а не бесконечную прямую.

В двумерной декартовой системе координат уравнение (1) сводится к паре одномерных параметрических уравнений вида:

$$x(t) = x_1 + (x_2 - x_1) \cdot t, \quad 0 \leq t \leq 1 \quad (2)$$

$$y(t) = y_1 + (y_2 - y_1) \cdot t, \quad 0 \leq t \leq 1, \quad (3)$$

В случае прямоугольного отсекающего окна одна из координат пересечения отрезка с каждой стороной известна. Нужно вычислить только вторую. Из (1) получаем:

$$t = \frac{P(t) - P_1}{P_2 - P_1},$$

а из (2) и (3):

$$t = \frac{x_{\Pi} - x_1}{x_2 - x_1},$$

для лев. ст.

$$t = \frac{x_{\Pi} - x_1}{x_2 - x_1},$$

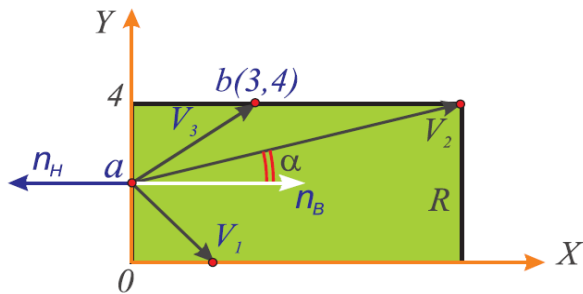
для прав. ст.

$$t = \frac{y_{\Pi} - y_1}{y_2 - y_1},$$

для ниж. ст.

$$t = \frac{y_{\Pi} - y_1}{y_2 - y_1}.$$

для верх. ст.



Если решения этих уравнений дают значения  $t$  за пределами интервала, то такие решения отвергаются, поскольку они соответствуют точкам, лежащими вне исходного отрезка. Пусть  $R$  — отсекающая область, являющаяся выпуклым многоугольником.

Рассмотрим пример вычисления пересечений отрезка с двумерным окном с использованием метода Кируса-Бека.

На рис. представлены два отрезка.

Для  $P1(-1, 1)$ ,  $P2(9, 3)$  уравнение прямой

$$y = 0.2(x - 6),$$

а точки пересечения с прямоугольным окном:

$$(0, 1.2) (8, 2.8).$$



Проверим это.

Параметрическое уравнение этого отрезка:

$$\begin{aligned} P(t) &= P_1 + (P_2 - P_1), \\ t &= [-1 \ 1] + [10 \ 2], \\ t &= (10t - 1)i + (2t + 1)j, \end{aligned}$$

где  $i, j$  — единичные векторы, ориентированные вдоль осей  $X$  и  $Y$  соответственно.

Обозначим и вычислим внутренние нормали к сторонам области отсечения:

$$n_L = i, \quad n_{II} = -i, \quad n_H = j, \quad n_B = -j.$$

Выбрав точку  $f(0, 0)$  на левой стороне окна, имеем:

$$\begin{aligned} P(t) - f &= (10t - 1)i + (2t + 1)j, \\ n_L[P(t) - f] &= 10t - 1 = 0, \quad t = 1/10. \end{aligned}$$

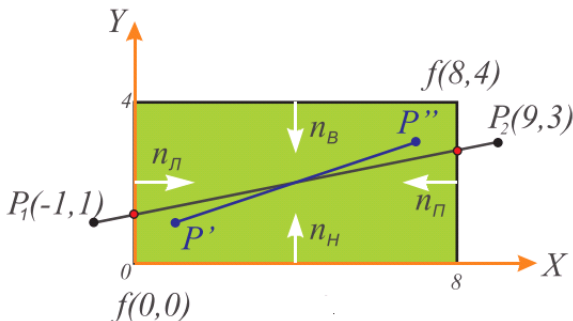
Подстановка

$$P(1/10) = [-1 \ 1] + [10 \ 2](1/10) = [0 \ 1.2],$$

дает то же значение, которое было вычислено другим методом.

Выбрав т.  $f(8, 4)$  на правой стороне окна, имеем:

$$\begin{aligned} P(t) - f &= (10t - 9)i + (2t - 3)j, \\ n_{II}[P(t) - f] &= -(10t - 9) = 0, \quad t = 9/10, \\ P(9/10) &= [-1 \ 1] + [10 \ 2](9/10) = [8 \ 2.8]. \end{aligned}$$



Для нижней стороны  $f(0,0)$ :

$$n_H[P(t) - f] = 2t + 1 = 0, \quad t = -1/2,$$

$t$  вне интервала  $0 \leq t \leq 1$ , поэтому отвергается.

Для верхней стороны  $f(8,4)$ :

$$n_B[P(t) - f] = -(2t - 3) = 0, \quad t = 3/2,$$

$t$  вне интервала  $0 \leq t \leq 1$ , поэтому отвергается.

Таким образом, видимый участок отрезка  $P_1P_2$  лежит в интервале  $1/10 \leq t \leq 9/10$  или от точки  $[0, 1.2]$  до точки  $[8, 2.8]$ .

Для полностью видимого отрезка  $P'(1,1)$   $P''(7,3)$  с параметрическим описанием

$$P(t) = [1 \ 1] + [6 \ 2]t$$

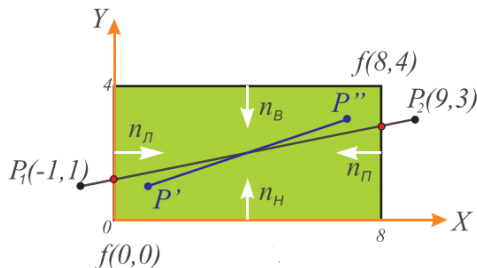
результаты вычисления:

- для левой стороны:  $t = -1/6$ ,
- для правой стороны:  $t = 7/6$ ,
- для нижней стороны:  $t = -1/2$ ,
- для верхней стороны:  $t = 3/2$ .

Все значения  $t$ , которые соответствуют

точкам пересечения, лежат вне интервала  $0 \leq t \leq 1$ .

Значит отрезок полностью видим.



Пусть дана некоторая выпуклая область. Точка  $A$  принадлежит ее границе, причем в этой точке можно получить внутреннюю нормаль  $n$ .

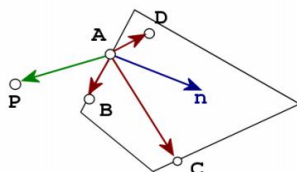
Тогда для каждой внутренней точки (например  $D$ ), равно как и для каждой граничной точки ( $C$ , либо  $B$ ) скалярное произведение нормали и направления на точку из  $A$  будет не отрицательным:

$$AD * n > 0, \quad AC * n > 0, \quad AB * n = 0$$

В то же время скалярное произведение не принадлежащей области точки ( $P$ ) и нормали всегда будет отрицательным:

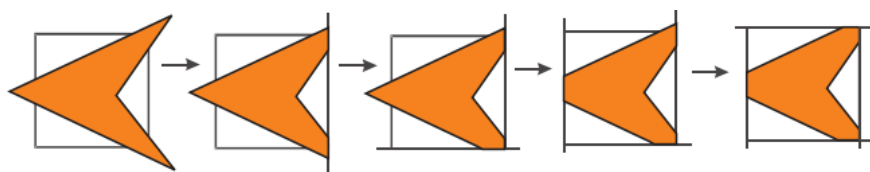
$$AP * n < 0$$

Данное утверждение становится более очевидным, если вспомнить, что скалярное произведение векторов – есть косинус наименьшего угла между ними. Эта идея может быть использована для определения местоположения произвольной точки относительно данной грани.

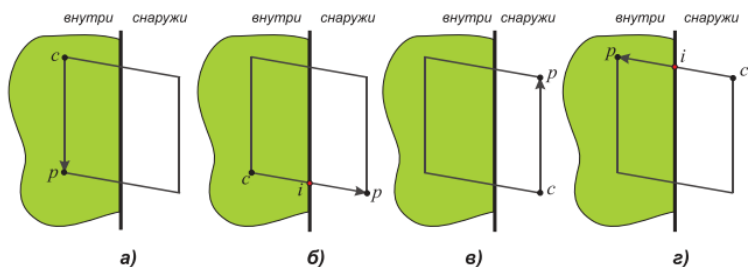


### Алгоритм Сазерленда - Ходжмана

Исходный и каждый из промежуточных многоугольников отсекается последовательно относительно одной стороны окна.



Исходный многоугольник задается списком ребер. Поскольку каждое ребро многоугольника отсекается независимо от других, то достаточно рассмотреть только возможные ситуации расположения одного ребра относительно одной отсекающей плоскости. Возможны 4 ситуации (рис.) для начальной точки  $c$ , конечной точки ребра  $p$  и отсекающей плоскости.



Результат — занесение в список вершин результирующего усеченного многоугольника полученных точек.

Для 1-ой вершины многоугольника необходимо определить только факт ее видимости. Если вершина видима, то она попадает в результат и становится начальной точкой с. Если вершина невидима, она тоже становится начальной точкой, но в результат не попадает.

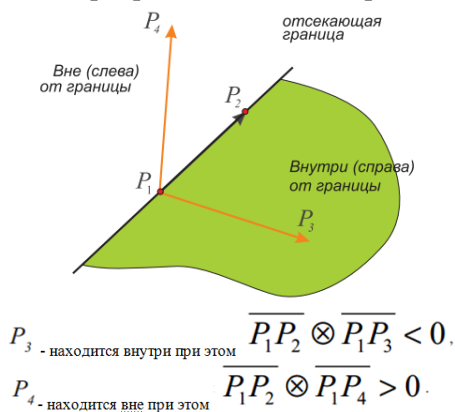
В случае а) ребро целиком лежит во внутренней области и точка р добавляется в результирующий многоугольник. В случае б) в результирующий многоугольник добавляется точка пересечения i. В случае в) обе вершины лежат во внешней плоскости и поэтому в результирующий многоугольник не добавляется ни одной точки. В случае г) в результирующий многоугольник добавляется точка пересечения i и точка р.

Видимость вершины можно определять разными способами:

1) как в АКБ с помощью скалярного произведения вектора, нормали на вектор, начинающийся в точке на границе и заканчивающийся в пробной точке;

2) при подстановке координат пробной точки в уравнение ориентированной прямой или плоскости.

3) Третий способ сводится к проверке знака координаты z у векторного произведения ребра и отсекающей границы (рис.).



### **Алгоритм отсечения многоугольника Вейлера- Азертон**

В предыдущих разделах были рассмотрены два алгоритма отсечения многоугольника, последовательно отсекающие произвольный (как выпуклый, так и невыпуклый) многоугольник каждой из сторон выпуклого окна. Зачастую же требуется отсечение по невыпуклому окну. Кроме того оба рассмотренных алгоритма могут генерировать лишние стороны для отсеченного многоугольника, проходящие вдоль ребра окна отсечения. Далее рассматриваемый алгоритм Вейлера-Азертон свободен от указанных недостатков ценой заметно большей сложности и меньшей скорости работы.

Предполагается, что каждый из многоугольников задан списком вершин, причем таким образом, что при движении по списку вершин в порядке их задания внутренняя область многоугольника находится справа от границы.

В случае пересечения границ и отсекаемого многоугольника и окна возникают точки двух типов:

- входные точки, когда ориентированное ребро отсекаемого многоугольника входит в окно,
- выходные точки, когда ребро отсекаемого многоугольника идет с внутренней на внешнюю стороны окна.

Общая схема алгоритма Вейлера-Азертон для определения части отсекаемого многоугольника, попавшей в окно, следующая:

1. Строятся списки вершин отсекаемого многоугольника и окна.
2. Отыскиваются все точки пересечения. При этом расчете касания не считаются пересечением, т.е. когда вершина или ребро отсекаемого многоугольника инцидентна или совпадает со стороной окна (рис. ).
3. Списки координат вершин отсекаемого многоугольника и окна дополняются новыми вершинами - координатами точек пересечения. Причем если точка пересечения  $P_k$  находится на ребре, соединяющем вершины  $V_i, V_j$ , то последовательность точек  $V_i, V_j$  превращается в последовательность  $V_i, P_k, V_j$ . При этом устанавливаются двухсторонние связи между одноименными точками пересечения в списках вершин отсекаемого многоугольника и окна.

Входные и выходные точки пересечения образуют отдельные подспски входных и выходных точек в списках вершин.

4. Определение части обрабатываемого многоугольника, попавшей в окно выполняется следующим образом:

- Если не исчерпан список входных точек пересечения, то выбираем очередную входную точку.

- Двигаемся по вершинам отсекаемого многоугольника пока не обнаружится следующая точка пересечения; все пройденные точки, не включая прервавшую просмотр, заносим в результат; используя двухстороннюю связь точек пересечения, переключаемся на просмотр списка вершин окна.

- Двигаемся по вершинам окна до обнаружения следующей точки пересечения; все пройденные точки, не включая последнюю, прервавшую просмотр, заносим в результат.

- Используя двухстороннюю связь точек пересечения, переключаемся на список вершин обрабатываемого многоугольника.

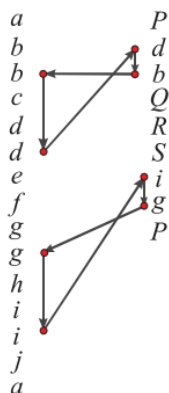
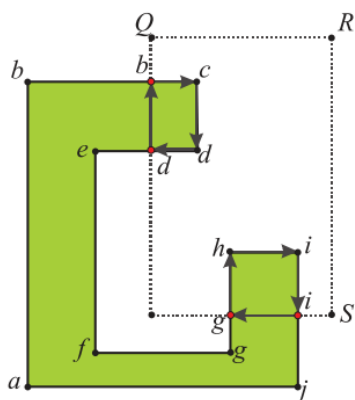
Эти действия повторяем пока не будет достигнута исходная вершина — очередная часть отсекаемого многоугольника, попавшая в окно, замкнулась. Переходим на выбор следующей входной точки в списке отсекаемого многоугольника.

Модификация этого алгоритма для определения части отсекаемого многоугольника, находящейся вне окна, заключается в следующем:

- исходная точка пересечения пересечения берется из списка выходных точек,

- движение по списку вершин окна выполняется в обратном порядке, т.е. так чтобы внутренняя часть отсекаемого была слева.

На рис. иллюстрируется отсечение многоугольника abcdefghi окном PQRS по алгоритму Вейлера- Азертонa.



### Трехмерное отсечение

Как и при двумерном отсечении, отрезки, которые полностью видимы или тривиально невидимы, можно идентифицировать с использованием обобщения кодов концевых точек Козна-Сазерленда.

В [трехмерном](#) случае используется 6-битовый код:

- в первый бит - если конец ребра левее объема,
- во второй бит - если конец ребра правее объема,
- в третий бит - если конец ребра ниже объема,
- в четвертый бит - если конец ребра выше объема,
- в пятый бит - если конец ребра ближе объема,
- в шестой бит - если конец ребра дальше объема.

Условия полной видимости отрезка и тривиальной невидимости аналогичны двумерному отсечению.

Поиск кодов точки относительно отсекающего прямоугольного параллелепипеда является прямым обобщением соответствующего двумерного алгоритма. Если отсекатель — усеченная пирамида, то один из методов заключается в преобразовании отсекателя в каноническую форму. В этом случае проверка кодов концевых точек заметно упрощается.

В более естественном методе, меньше искажающем форму отсекателя, отрезок, соединяющий центр проекции с центром

The diagram consists of two parts, (a) and (b), illustrating the projection of a point  $P_1$ .

Part (a) shows a perspective view. A point  $P_1$  is shown in the upper left. A dashed line connects  $P_1$  to the center of projection, labeled "Центр проекции" and  $Z_{цп}$ . The center of projection is located on the  $+Z$  axis. The  $Z$  axis is horizontal, with  $+Z$  pointing left and  $-Z$  pointing right. The  $X$  axis is vertical, with  $+X$  pointing down and  $-X$  pointing up. A green cone represents the field of view. The cone is divided into two regions: "ближняя" (near) and "дальняя" (far). The "ближняя" region is green, and the "дальняя" region is orange. The boundary between them is a vertical line at  $Z_0$ . The far boundary is at  $Z_n$ . The distance from the center of projection to the far boundary is labeled  $a$ . A point  $P_2$  is shown on the  $-X$  axis within the orange region.

Part (b) shows an orthographic view. It is a 3D representation of the orange region from part (a). The point  $P_2$  is shown inside the volume. A dashed line connects  $P_2$  to the point  $P_1$  on the bottom face of the volume. The distance from the bottom face to the top face is labeled  $b$ .

$$x = \frac{z - z_{un}}{z_{\partial} - z_{un}} \cdot z_n = z\alpha_1 + \alpha_2$$

$$\alpha_1 = \frac{x_n}{z_\partial - z_{un}}, \alpha_2 = -\alpha_1 z_{un}$$

76



$$f_n = x - z\alpha_1 - \alpha_2 \begin{cases} > 0, \text{ если } P \text{ справа от плоскости} \\ = 0, \text{ если } P \text{ на плоскости} \\ < 0, \text{ если } P \text{ слева от плоскости} \end{cases}$$

Пробные функции для левой, верхней и нижней границы имеют вид:

$$f_n = z\beta_1 - \beta_2 \begin{cases} > 0, \text{ если } P \text{ справа от плоскости} \\ = 0, \text{ если } P \text{ на плоскости} \\ < 0, \text{ если } P \text{ слева от плоскости} \end{cases}$$

$$\beta_1 = \frac{x_n}{z_\partial - z_{цп}}, \beta_2 = -\beta_1 z_{цп}$$

$$f_e = y - z\gamma_1 - \gamma_2 \begin{cases} > 0, \text{ если } P \text{ выше от плоскости} \\ = 0, \text{ если } P \text{ на плоскости} \\ < 0, \text{ если } P \text{ ниже от плоскости} \end{cases}$$

$$\gamma_1 = \frac{y_e}{z_\partial - z_{цп}}, \gamma_2 = -\gamma_1 z_{цп}$$

$$f_e = y - z\delta_1 - \delta_2 \begin{cases} > 0, \text{ если } P \text{ выше от плоскости} \\ = 0, \text{ если } P \text{ на плоскости} \\ < 0, \text{ если } P \text{ ниже от плоскости} \end{cases}$$

$$\delta_1 = \frac{y_i}{z_\partial - z_{цп}}, \delta_2 = -\delta_1 z_{цп}$$

Наконец, пробные функции для ближней и дальней граней имеют вид:

$$f_6 = z - z_6 \begin{cases} > 0, \text{ если } P \text{ ближе плоскости} \\ = 0, \text{ если } P \text{ на плоскости} \\ < 0, \text{ если } P \text{ дальше плоскости} \end{cases}$$

$$f_{\partial} = z - z_{\partial} \begin{cases} > 0, \text{ если } P \text{ ближе плоскости} \\ = 0, \text{ если } P \text{ на плоскости} \\ < 0, \text{ если } P \text{ дальше плоскости} \end{cases}$$

Чем ближе  $z_{\text{цп}}$  к бесконечности, тем больше форма отсекаателя приближается к прямоугольному параллелепеду. Пробные функции при этом тоже приближаются к соответствующим пробным функциям прямоугольного параллелепипеда.