

## 5 ОТСЕЧЕНИЕ ОТРЕЗКОВ

Если изображение выходит за пределы экрана, то на части дисплеев увеличивается время построения за счет того, что изображение строится в "уме". В некоторых дисплеях выход за пределы экрана приводит к искажению картины, так как координаты просто ограничиваются при достижении ими граничных значений, а не выполняется точный расчет координат пересечения (эффект "стягивания" изображения). Особенно актуальным является отсечение сцены по границам окна видимости в связи с широким использованием технологии просмотра окнами.

В простых графических системах достаточно двумерного отсечения, в трехмерных пакетах используется трех и четырехмерное отсечение.

Программное исполнение отсечения достаточно медленный процесс, поэтому, естественно, в мощные дисплеи встраивается соответствующая аппаратура. Первое сообщение об аппаратуре отсечения, использующей алгоритм отсечения делением отрезка пополам и реализованной в устройстве Clipping Divider, появилось в 1968 г.

Рассмотрим программные реализации алгоритмов отсечения. Целью алгоритмов отсечения является определение тех точек отрезков или их частей, которые лежат внутри отсекающего окна. Эти элементы остаются для визуализации, а остальные отбрасываются.

Отсекаемые отрезки могут быть трех классов (рис.5.1) - целиком видимые (отрезок 1), целиком невидимые (отрезок 2) и пересекающие окно (отрезки 3, 4).

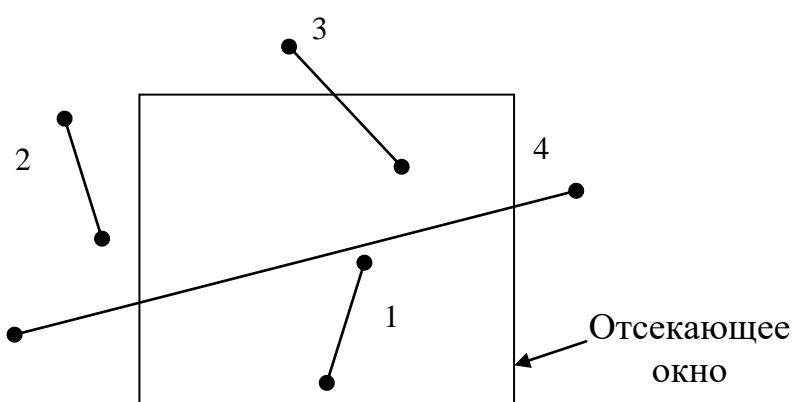


Рисунок 5.1 – Классификация отрезков по расположению относительно окна

## 5.1 Простой алгоритм двумерного отсечения

Обозначим ребра отсекающего окна следующим образом (рис.5.2):

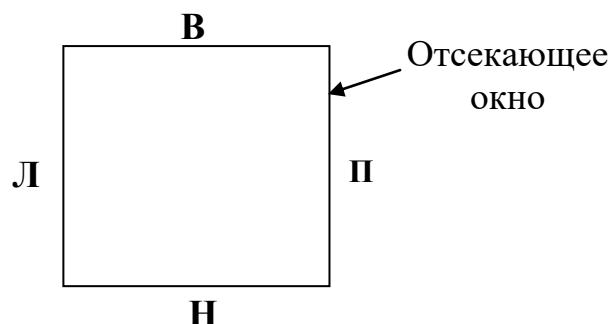


Рисунок 5.2 – Ребра отсекающего окна

Если точка имеет координаты ( $x, y$ ) и удовлетворяются условия:

$x_{\underline{L}} \leq x \leq x_n$ , то точка ( $x, y$ ) лежит внутри отсекающего окна.  
 $y_{\underline{n}} \leq y \leq y_e$

Пусть концы отрезка имеют координаты:  $(x_1, y_1)$  и  $(x_2, y_2)$ .

Рассмотрим простой алгоритм двумерного отсечения:

If  $x_1 < x_{\underline{L}}$  OR  $x_1 > x_n$  Then goto 1;

If  $x_2 < x_{\underline{L}}$  OR  $x_2 > x_n$  Then goto 1;

If  $y_1 < y_{\underline{n}}$  OR  $y_1 > y_e$  Then goto 1;

If  $y_2 < y_{\underline{n}}$  OR  $y_2 > y_e$  Then goto 1;

goto 3;

1: If  $x_1 < x_{\underline{L}}$  AND  $x_2 < x_{\underline{L}}$  Then goto 2;

    If  $x_1 > x_n$  AND  $x_2 > x_n$  Then goto 2;

    If  $y_1 > y_e$  AND  $y_2 > y_e$  Then goto 2;

    If  $y_1 < y_{\underline{n}}$  AND  $y_2 < y_{\underline{n}}$  Then goto 2;

Отрезок частично виден. Определяем точки пересечения отрезка с

окном.

2: Отрезок невидим.

3: Отрезок полностью видим, перейти к следующему отрезку.

Найдем точки пересечения прямой со сторонами окна. Уравнение бесконечной прямой, проходящей через точки  $P_1 (x_1, y_1)$  и  $P_2 (x_2, y_2)$ , имеет вид:

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

где  $m = \frac{y_2 - y_1}{x_2 - x_1}$  - наклон данной прямой.

При расчете пересечения используется горизонтальность либо вертикальность сторон окна, что позволяет определить координату X или Y точки пересечения без вычислений. Другая координата точки пересечения данной прямой со сторонами окна находится по формулам:

с левой стороной ( $x_{\text{л}}$ ):  $y = m(x_{\text{л}} - x_1) + y_1$

с правой стороной ( $x_{\text{п}}$ ):  $y = m(x_{\text{п}} - x_1) + y_1$

с верхней стороной ( $y_{\text{в}}$ ):  $x = x_1 + \left(\frac{1}{m}\right) \cdot (y_{\text{в}} - y_1)$ ,  $m \neq 0$

с нижней стороной ( $y_{\text{н}}$ ):  $x = x_1 + \left(\frac{1}{m}\right) \cdot (y_{\text{н}} - y_1)$ ,  $m \neq 0$

## 5.2 Двумерный алгоритм Коэна-Сазерленда

Этот алгоритм позволяет быстро выявить отрезки, которые могут быть или приняты или отброшены целиком. Вычисление пересечений требуется, когда отрезок не попадает ни в один из этих классов. Данный алгоритм особенно эффективен в двух крайних случаях:

- 1) большинство отрезков содержится целиком в большом окне,
- 2) большинство отрезков лежит целиком вне относительно маленького окна.

Проще говоря – суть алгоритма заключается в следующем – найти тривиально видимую часть отрезка

Идея алгоритма состоит в следующем.

Окно отсечения и прилегающие к нему части плоскости вместе образуют 9 областей (рис.5.3). Каждой из областей присвоен 4-х разрядный код.

Смысл разрядов кода:

- 1 pp = 1 - точка над верхним краем окна;
- 2 pp = 1 - точка под нижним краем окна;
- 3 pp = 1 - точка справа от правого края окна;
- 4 pp = 1 - точка слева от левого края окна.

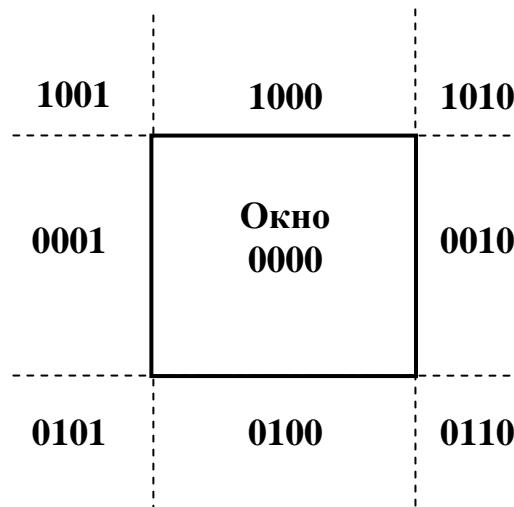


Рисунок 5.3

Две конечные точки отрезка получают 4-х разрядные коды, соответствующие областям, в которые они попали.

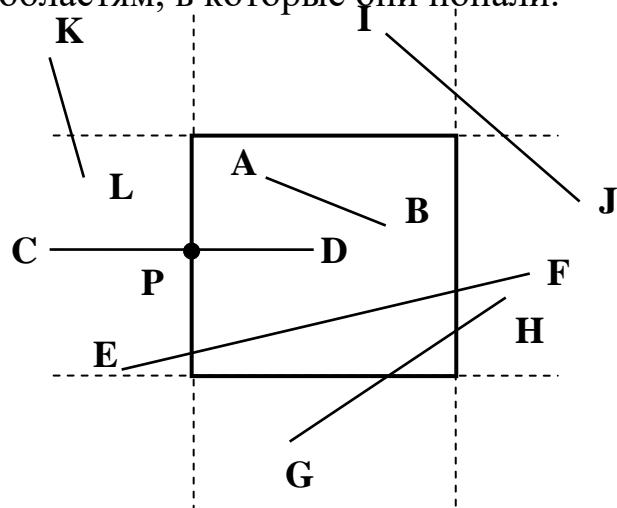


Рисунок 5.4 - Отсечение по методу Коэна-Сазерленда

Определение того лежит ли отрезок целиком внутри окна или целиком вне окна выполняется следующим образом:

- если *коды обоих концов отрезка равны 0*, то отрезок целиком внутри окна, отсечение не нужно, отрезок принимается как тривиально видимый (отрезок АВ на рис. 5.4);
- если *логическое И (&) кодов обоих концов отрезка не равно нулю*, то отрезок целиком вне окна, отсечение не нужно, отрезок отбрасывается как тривиально невидимый (отрезок KL на рис. 5.4);
- если *логическое И (&) кодов обоих концов отрезка равно нулю*, то отрезок подозрительный, он может быть частично видимым (отрезки CD, EF, GH) или целиком невидимым (отрезок IJ); для него нужно определить координаты пересечений со сторонами окна и для каждой полученной части определить тривиальную видимость или невидимость. При этом, например, для отрезка CD потребуется вычисление одного пересечения, для EF и GH - двух.

Опять при расчете пересечения используется горизонтальность либо вертикальность сторон окна, что позволяет определить координату X или Y точки пересечения без вычислений. Другая координата точки пересечения находится по формулам, приведенным в п.5.1.

При непосредственном использовании описанного выше способа отбора целиком видимого или целиком невидимого отрезка после расчета пересечения потребовалось бы вычисление кода точки пересечения. Для примера рассмотрим отрезок CD. Точка пересечения обозначена как Р. В силу того, что граница окна считается принадлежащей окну, то можно просто в качестве видимого отрезка принять только часть PD, попавшую в окно. Часть же отрезка CR, на самом деле оказавшаяся вне окна, потребует дальнейшего рассмотрения, так как логическое И кодов точек C и R даст 0, т.е. отрезок CR нельзя просто отбросить. Для решения этой проблемы Коэн и Сазерленд предложили заменять конечную точку с ненулевым кодом конца на точку, лежащую на стороне окна либо на его продолжении.

В целом схема алгоритма Коэна-Сазерленда следующая:

1. Рассчитать коды конечных точек отсекаемого отрезка.

В цикле повторять пункты 2-6:

2. Если логическое И кодов конечных точек не равно 0, то отрезок целиком вне окна. Он отбрасывается и отсечение закончено.

3. Если оба кода равны 0, то отрезок целиком видим. Он принимается и отсечение закончено.

4. Если начальная точка внутри окна, то она меняется местами с конечной точкой.

5. Анализируется код начальной точки для определения стороны окна, с которой есть пересечение, и выполняется расчет пересечения. При этом вычисленная точка пересечения заменяет начальную точку.

6. Определение нового кода начальной точки.

*Почему меняем местами начальную точку с конечной? Ключом к алгоритму Сазерленда - Коэна является информацию о том, что одна из концевых точек отрезка лежит вне окна. Поэтому тот отрезок, который заключен между этой точкой и точкой пересечения, можно отвергнуть как невидимый. Фактически это означает замену исходной концевой точки на точку пересечения*

## Пример 1

1) Например, рассмотрим отрезок CD.

Шаг 1. Код C – 0001, код D – 0000.

Шаг 2. Логическое И их равно 0, тогда выполняем шаг 3.

Шаг 3. Коды C и D не равны 0000, тогда выполняем шаг 4.

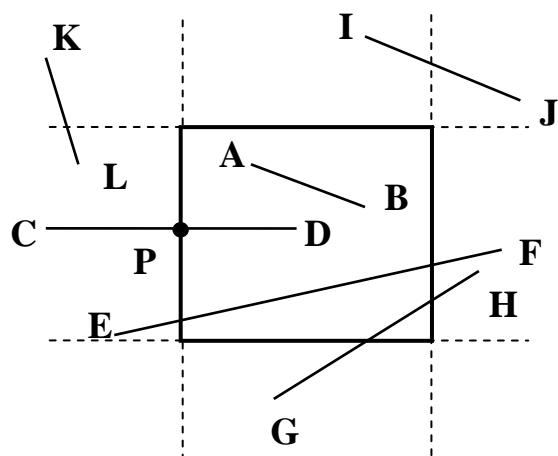
Шаг 4. Начальная точка С лежит вне окна.

Шаг 5. Значит можно найти сторону окна, с которой есть пересечение (левая сторона окна). Находим точку пересечения Р. Она становится начальной (т.е. теперь должны рассмотреть отрезок PD, часть CP совсем забыли)

Шаг 6. Определяем код точки Р. Т.к. она лежит на стороне окна, то ее код равен 0000. Переходим на шаг 2.

Шаг 2. Логическое И их равно 0, тогда выполняем шаг 3.

Шаг 3. Коды Р и D равны 0000, следовательно, отрезок видим. Его выводим на экран. Отсечение закончено.



## Пример 2

2) Например, рассмотрим отрезок GH.

Шаг 1. Код G – 0100, код H – 0010.

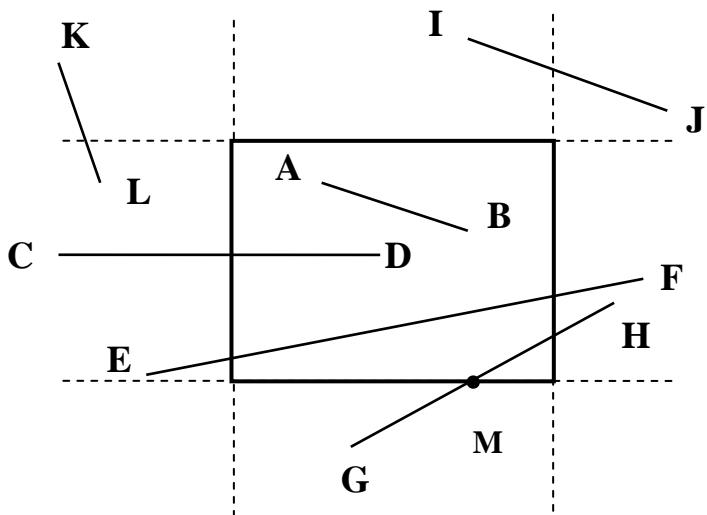
Шаг 2. Логическое И их равно 0, тогда выполняем шаг 3.

Шаг 3. Коды G и H не равны 0000, тогда выполняем шаг 4.

Шаг 4. Начальная точка G лежит вне окна.

Шаг 5. Значит можно найти сторону окна, с которой есть пересечение (нижняя сторона окна). Находим точку пересечения M. Она становится начальной (т.е. теперь должны рассмотреть отрезок MH, часть GM совсем забыли)

Шаг 6. Определяем код точки M. Т.к. она лежит на стороне окна, то ее код равен 0000. Переходим на шаг 2.

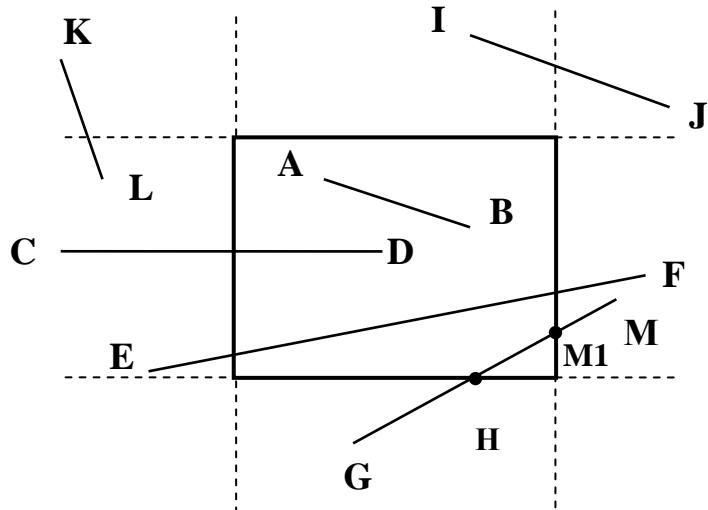


Шаг 2. Логическое И их равно 0, тогда выполняем шаг 3.

Шаг 3. Коды M и H не равны 0000, следовательно, переход на шаг 4.

Шаг 4. Начальная точка M лежит внутри окна. Меняем ее местами с конечной (код H теперь равен 0000, код M – 0010).

Шаг 5. Теперь находим сторону окна, с которой есть пересечение (правая сторона окна). Находим точку пересечения M1. Она становится начальной (т.е. теперь должны рассмотреть отрезок HM1, часть M1M совсем забыли)



Шаг 6. Определяем код точки М1. Т.к. она лежит на стороне окна, то ее код равен 0000. Переходим на шаг 2.

Шаг 2. Логическое И точек Н и М1 равно 0, тогда выполняем шаг 3.

Шаг 3. Коды Н и М1 равны 0000, следовательно, отрезок видим. Его (НМ1) выводим на экран. Отсечение закончено.

### Пример 3

3) Например, рассмотри отрезок IJ.

Шаг 1. Код I – 1000, код J – 0010.

Шаг 2. Логическое И их равно 0, тогда выполняем шаг 3.

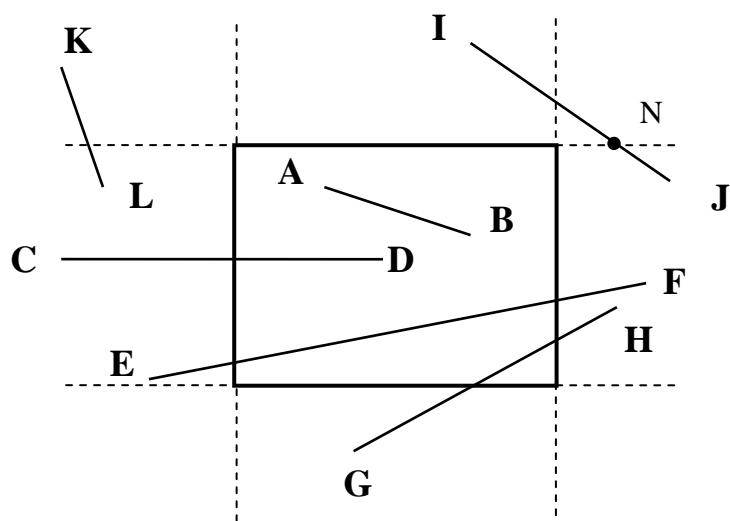
Шаг 3. Коды I и J не равны 0000, тогда выполняем шаг 4.

Шаг 4. Начальная точка I лежит вне окна.

Шаг 5. Значит можно найти сторону окна, с которой есть пересечение (верхняя сторона окна). Находим точку пересечения N. Она становится начальной (т.е. теперь должны рассмотреть отрезок NJ, часть IN совсем забыли)

Шаг 6. Определяем код точки N. Т.к. она лежит не на стороне окна, то ее код должен быть равен продолжению стороны – 1010.

Переходим на шаг 2.



Шаг 2. Логическое И точек Ni J не равно 0, следовательно, отрезок невидим. Его не выводим на экран. Отсечение закончено.