Московский государственный технический университет
им.Н.Э.Баумана

С.М.Авдеева, А.В.Куров

# АЛГОРИТМЫ ОТСЕЧЕНИЯ

Методические указания по выполнению лабораторных работ по курсу "Машинная графика"

Издательство МГТУ им. Н.Э.Баумана

1997

## ВВЕДЕНИЕ

В данных методических указаниях продолжается изложение основных алгоритмов плоской машинной графики, начатое в пособии "Базовые алгоритмы машинной графики". В предлагаемой работе рассматриваются математические основы алгоритмов отсечения на плоскости и сами алгоритмы.

Задача отсечения изображения, в частности его простейших составляющих - отрезков и многоугольников - может рассматриваться как довольно интересная самостоятельная математическая задача. Однако в рамках машинной графики она приобретает и важное прикладное значение.

Отсечение используется в алгоритмах устранения ступенчатости, при создании трехмерных и реалистических изображений для удаления невидимых линий и поверхностей, для учета теней, формирования фактуры.

Отсечение используется при построении сцен реальных объектов для выделения тех фрагментов, которые попадают в поле видимости наблюдателя.

В предлагаемой работе рассматриваются алгоритмы отсечения отрезков и многоугольников на плоскости. Данные геометрические объекты являются наиболее распространенными при построении реальных сцен. Отсечение в пространстве либо сводится к отсечению на плоскости, либо сами плоские алгоритмы без труда распространяются на трехмерный случай. Об этом будет сказано по ходу изложения материала.

## ОБЩИЕ ПОЛОЖЕНИЯ

В ходе выполнения лабораторных работ студенты реализуют, как правило, один алгоритм отсечения отрезка прямоугольным отсекателем, алгоритм отсечения отрезка произвольным выпуклым окном и алгоритм отсечения многоугольника выпуклым или произвольным отсекателем.

При выполнении данных работ исходными данными для прямоугольного отсекателя являются координаты его левой, правой, верхней и нижней границ; для произвольного выпуклого отсекателя следует задать количество вершин и их координаты.

Отрезок задается координатами двух своих концов. Отсекаемый многоугольник задается количеством вершин и их координатами. Если же многоугольник имеет отверстия, то эту информацию надо задать как для внешней границы, так и для всех внутренних границ.

Поскольку приходится задавать довольно большое количество исходных данных, то целесообразно организовать их ввод с использованием "мыши" путем указания точек экрана, являющихся вершинами отрезков (многоугольников).

Так как проверка правильности работы программы осуществляется ее тестированием при различных исходных данных (различных положениях отрезков и многоугольников), то удобно производить отсечение сразу нескольких отрезков или многоугольников одним отсекателем (то есть не вводить многократно координаты отсекателя).

Проверка правильности полученных результатов облегчается, если отсекатель изображать одним цветом, исходный отрезок (многоугольник) - вторым цветом, а полученный результат - третьим цветом.

## ЛАБОРАТОРНАЯ РАБОТА N1
## ОТСЕЧЕНИЕ ПРОИЗВОЛЬНОГО ОТРЕЗКА ОКНОМ ПРЯМОУГОЛЬНОЙ ФОРМЫ С ПОМОЩЬЮ ПРОСТОГО АЛГОРИТМА ОТСЕЧЕНИЯ

Цель работы: изучение и программная реализация простого алгоритма отсечения отрезка.

В данной и последующих двух работах в качестве отсекателя берется прямоугольник со сторонами, параллельными координатным осям. Такой отсекатель называют отсекателем стандартной формы. Его положение задается четырьмя величинами: $X_л$, $X_{пр}$ - абсциссами левой и правой границ, $Y_в$, $Y_н$ - ординатами верхней и нижней границ.

Положение точки с координатами X,Y относительно окна можно описать четырехбитовым кодом. Каждый бит этого кода формируется по следующему правилу:

1, если $X<X_л$ (точка находится левее окна)

$T_1$= 0, иначе

1, если $X>X_{пр}$ (точка лежит правее окна)

$T_2$= 0, иначе

1, если $Y<Y_н$ (точка лежит ниже окна) $T_3$= 0, иначе

1, если $Y>Y_в$ (точка лежит выше окна) $T_4$= 0, иначе

Точка, лежащая внутри окна, имеет нулевые значения всех четырех разрядов. Отрезок является полностью видимым, то есть лежащим внутри отсекателя, если обе его вершины лежат внутри отсекателя. Таким образом, если обе суммы кодов концов отрезка равны нулю, то отрезок будет видимым.

Полностью невидимый отрезок (целиком лежащий за пределами окна) можно определить, вычислив побитное логическое произведение кодов его концов. Если оно отлично от нуля, то отрезок - полностью невидимый. В случае равенства нулю логического произведения отрезок может быть целиком или частично видимым или целиком невидимым. В этом случае и приходится использовать собственно алгоритм отсечения.

Для нахождения координат точек пересечения отрезка со сторонами отсекателя можно воспользоваться параметрическим уравнением отрезка:

$Y=m(X-X_1)+Y_1$ или $Y=m(X-X_2)+Y_2$,

где $(X_1,Y_1)$, $(X_2,Y_2)$ - координаты концов отрезка,

$m= (Y_2-Y_1)/(X_2-X_1)$ - тангенс угла наклона отрезка. С левой и правой границами отсекателя может пересечься любой отрезок, кроме вертикального, то есть m=00, и координаты точек пересечения с этими границами определяются из следующих выражений:

с левой границей $X=X_л$, $Y=m(X_л-X_1)+Y_1$

с правой границей X=Xпр, Y=m(Xпр-X1)+Y1

С верхней и нижней границами отсекателя может пересечься любой отрезок, кроме горизонтального, то есть m=0, и координаты точек пересечения с этими границами определяются из двух других выражений:

с верхней границей $Y=Y_в$, $X=(Y_в-Y_1)/m+X_1$

с нижней границей Y=Yн, X=(Yн-Y1)/m+X1

В простом алгоритме отсечения при поиске точки пересечения отрезка с очередной границей отсекателя прежде всего проверяется возможность пересечения отрезка с данной границей. Если пересечение возможно, то находится точка пересечения, для которой проверяется корректность. Под корректностью здесь понимается попадание точки в пределы отсекателя (найденная точка пересечения может лежать на продолжении стороны отсекателя. Если пересечение оказалось некорректным, то проведенные вычисления оказываются фактически напрасными и в этом случае ищется пересечение со следующей стороной отсекателя.

Если пересечение оказалось корректным, то оно заносится в результат и далее ищется вторая точка пересечения отрезка с отсекателем. Известно, что прямая может пересечь выпуклый многоугольник только в двух точках. Найденный отрезок

вычерчивается. Простой алгоритм отсечения отрезка может быть представлен в следующем виде:

1. Ввод координат отсекателя $X_л$, $X_{пр}$,$Y_н$, $Y_в$.

2. Ввод координат концов отрезка $P_1(X_1,Y_1)$, $P_2(X_2,Y_2)$.

3. Вычисление кодов концов отрезка $T_1$, $T_2$.

Вычисление сумм кодов концов отрезка $S_1$, $S_2$.

4. Установка признака видимости отрезка pr=1

(pr=1 - отрезок видимый; pr=-1 - отрезок невидимый).

5. Задание начального значения тангенса угла наклона отрезка $m=10^{30}$ (большое число, предполагается, что отрезок вертикальный).

6. Проверка полной видимости отрезка:

если $(S_1=0)\&(S_2=0)$=true, то отрезок видимый; выполнение в этом случае следующих действий: занесение в результат координат концов отрезка $R_1=P_1$, $R_2=P_2$ и переход к п. 31.

7. Вычисление логического произведения кодов концов отрезка P.

8. Проверка тривиальной невидимости отрезка: если P=0, то отрезок невидим. В этом случае установка признака pr=-1 и переход к п. 31.

9. Проверка видимости первого конца отрезка:

если $S_1=0$ (первый конец видим), то выполнение следующих действий: $R_1=P_1$ (занесение этой вершины в результат), $Q=P_2$ (занесение координат другой вершины в рабочую переменную Q), i=2 (номер шага отсечения), переход к п.15.

10. Проверка видимости второго конца отрезка:

если $S_2=0$ (первый конец видим), то выполнение следующих действий: $R_1=P_2$ (занесение этой вершины в результат), $Q=P_1$ (занесение координат другой вершины в рабочую переменную Q), i=2 (номер шага отсечения), переход к п.15.

11. Установка начального значения шага отсечения i=0.

12. Вычисление текущего номера шага отсечения i=i+1.

13. Проверка завершения процедуры отсечения: если i>2, то переход к п.31.

14. Занесение в рабочую переменную Q координат i-ой вершины $Q=P_i$.

15. Определение расположения отрезка: если $X_2=X_1$ (отрезок вертикальный), то переход к п.23 (не может быть пересечения с левой и правой границами отсекателя).

16. Вычисление тангенса угла наклона отрезка $m=(Y_2-Y_1)/(X_2-X_1)$.

17. Проверка возможности пересечения с левой границей отсекателя: если $Q_x>X_л$ (пересечения нет), то переход к п.20.

18. Вычисление ординаты точки пересечения отрезка с левой границей отсекателя: $Y_р=m(X_л-Q_x)+Q_y$.

19. Проверка корректности найденного пересечения: если

$(Y_р>Y_н)\&(Y_р<Y_в)$=true (пересечение корректное), то выполнение следующих действий: $R_{i.x}=X_л$, $R_{i.y}=Y_р$ (занесение полученных координат в результат), переход к п.12.

20. Проверка возможности пересечения отрезка с правой границей отсекателя: если $Q_{.x}<X_п$ (пересечения нет), то переход к п.23.

21. Вычисление ординаты точки пересечения с правой границей: $Y_р=m(X_п-Q_{.x})+Q_{.y}$.

22. Проверка корректности найденного пересечения: если $(Y_р>Y_н)\&(Y_р<Y_в)$=true (пересечение корректно), то выполнение следующих

действий : $R_{i.x}=X_п$, $R_{i.y}=Y_р$ (занесение полученных координат в результат), переход к п.12.

23. Проверка горизонтальности отрезка: если m=0, то переход к п.12.

24. Проверка возможности пересечения с верхней границей отсекателя: если $Q.y<Y_в$ (пересечения нет), то переход к п.27.

25. Вычисление абсциссы точки пересечения с верхней границей:

$X_р=(Y_в-Q.y)/m+Q.x$

26. Проверка корректности найденного пересечения:

если $X_р>X_л)\&(X_р<X_п)$=true (пересечение корректно), то выполнение следующих действий: $R_i.x=X_р$; $R_{iy}=Y_в$ (занесение полученных координат в результат); переход к п. 12.

27. Проверка возможности пересечения с нижней границей отсекателя: если $Q.x>Y_н$ (пересечения нет), то переход к п. 30 (вершина невидима и ни одно пересечение не является корректным, следовательно отрезок невидим).

28. Вычисление абсциссы точки пересечения с нижней границей: $X_р=(Y_н-Q.y)/m+Q.x$

29. Проверка корректности найденного пересечения:

если $(X_р>X_л)\&(X_р<X_п)$=true (пересечение корректно), то выполнение следующих действий: $R_i.x=X_р$; $R_{iy}=Y_н$ (занесение полученных координат в результат); переход к п. 12.

30. Установка признака видимости pr=-1 (отрезок невидим полностью, так как ни одно пересечение не оказалось корректным).

31. Проверка признака видимости: если pr=1, то вычерчивание отрезка $R_1R_2$.

32. Конец.

## ЛАБОРАТОРНАЯ РАБОТА N2
## ОТСЕЧЕНИЕ ПРОИЗВОЛЬНОГО ОТРЕЗКА ОКНОМ ПРЯМОУГОЛЬНОЙ ФОРМЫ С ПОМОЩЬЮ АЛГОРИТМА САЗЕРЛЕНДА-КОЭНА
(на основе разбиения отрезка окном)

Цель работы: изучение и программная реализация алгоритма Сазерленда-Коэна отсечения отрезка.

Алгоритм Сазерленда-Коэна, как и в предыдущем случае, предусматривает нахождение точек пересечения отрезка со сторонами окна прямоугольной формы. Однако здесь не производится проверка корректности найденных точек пересечения. Найденная точка пересечения разбивает отрезок на две части: полностью невидимую относительно очередной стороны отсекателя и видимую часть. Невидимой будет та часть отрезка, которая заключена от вершины отрезка, невидимой относительно текущей стороны окна, до точки пересечения. Этот факт используется в алгоритме для определения части отрезка, подлежащей отсечению. Это связано стем, что точка, попадающая на ребро отсекателя имеет нулевой код (она считается видимой), поэтому попарное логическое произведение кодов концов будет равно нулю.

В алгоритме предусматривается предварительный анализ сумм кодов концов отрезка и попарных логических произведений этих кодов с целью определения тривиально видимых и тривиально невидимых отрезков.

Если отрезок не является ни тривиально видимым, ни тривиально невидимым, то производится собственно его отсечение. При этом предполагается,

что невидимой относительно каждого ребра должна быть первая вершина отрезка. Поэтому в случае необходимости вершины меняются местами. Следует отметить, что одновременно обе вершины не могут быть невидимыми относительно одного текущего ребра отсекателя, так как этот факт позволил бы на предварительном этапе отбросить отрезок как тривиально невидимый.

Для удобства работы данные, задающие отсекатель, заносятся в массив "окно" O(4), в котором $O_1=X_л$, $O_2=X_п$, $O_3=Y_в$, $O_4=Y_н$. Отсечение производится в определенном порядке: левой, правой, верхней, нижней границами отсекателя. Поэтому для отыскания точек пересечения в выражения ( ) следует на i-ом шаге подставлять i-ые элементы массива O. Такое задание исходных данных позволяет для горизонтального отрезка не проводить третий и четвертый этапы, а для вертикального отрезка - первый и второй этапы.

В алгоритме используется признак (флаг) Fl, определяющий расположение отрезка: Fl=-1 - отрезок вертикальный, Fl=0 - общего положения, Fl=1 - горизонтальный.

Сам алгоритм Сазерленда-Коэна можно представить в следующем виде:

1. Ввод координат отсекателя $X_л(O_1)$, $X_п(O_2)$, $Y_в(O_3)$, $Y_н(O_4)$.
2. Ввод координат концов отрезка $P_1(X_1,Y_1)$, $P_2(X_2,Y_2)$.
3. Установка начального значения флага Fl=0.
4. Проверка вертикальности отрезка: если $P_{2.x}-P_{1.x}=0$ (вертикальный), то Fl=-1, иначе вычислить тангенс угла наклона отрезка

$m=(P_{2.y}-P_{1.y})/(P_{2.x}-P_{1.x})$

5. Проверка горизонтальности отрезка: если m=0, то Fl=1.
6. Начало цикла по i от 1 до 4 отсечения отрезка по всем четырем сторонам отсекателя.
7. Обращение к алгоритму (подпрограмме) определения видимости отрезка $P_1P_2$ относительно заданного окна. Подпрограмма возвращает признак pr, принимающий следующие значения: pr=1 - отрезок видимый; pr=-1 - отрезок полностью невидимый; pr=0 - отрезок может быть частично видимым.
8. Анализ полученного признака видимости: если pr=-1, то
   переход к п. 20; если pr-1, то преход к п. 19.
9. Проверка видимости обеих вершин отрезка относительно текущей i-ой стороны окна:
   если $T_1(i)=T_2(i)$, то переход к п.18.
10. Проверка видимости первой вершины:
   если $T_1(i)=0$ (вершина видима), то обмен местами вершин:
   $R=P_1$; $P_1=P_2$; $P_2=R$.
11. Проверка вертикальности отрезка: если Fl=-1, то
   переход к п. 14.
12. Анализ номера шага отсечения: если i>3, то переход к п. 14.
13. Вычисление координат точки пересечения с i-ым ребром отсекателя (левым или правым):
   $P_{1.y}=m(O_i-P_{1.x})+P_{1.y}$; $P_{1.x}=O_i$.
   Переход к п. 18.
14. Проверка горизонтальности отрезка: если Fl=1, то переход к п. 18.
15. Проверка вертикальности отрезка: если Fl=-1, то переход к п.17.

16. Вычисление абсциссы точки пересечения отрезка общего положения со стороной отсекателя (верхней или нижней):

$P_{1.x}=(O_i-P_{1.y})/m+P_{1.x}$

17. Присвоение ординате вершины отрезка ординаты стороны отсекателя:

$P_{1.y}=O_i$

18. Конец цикла по i (вычисление нового значения параметра цикла i=i+1, анализ его значения и переход на повторное выполнение цикла или выход из цикла).

19. Вычерчивание отрезка $P_1P_2$.

20. Конец.

При работе с вертикальными отрезками нет необходимости определять номер шага отсечения. Невидимый вертикальный отрезок относительно бокового ребра отсекателя будет полностью невидимым, это обнаружится в самом начале. Если же он видим относительно бокового ребра, то на первых двух шагах автоматически произойдет переход к следующему шагу цикла.

Алгоритм определения видимости отрезка относительно окна можно представить следующим образом.

1. Вычисление кодов первой вершины отрезка $T_1$.
2. Вычисление кодов второй вершины отрезка $T_2$.
3. Вычисление суммы кодов первой вершины $S_1= \sum_{i=1}^{4} T_1(i)$
4. Вычисление суммы кодов второй вершины $S_2= \sum_{i=1}^{4} T_2(i)$
5. Определение полной видимости отрезка:
   если $(S_1=0)\&(S_2=0)$=true, то pr=1 и переход к п. 8.
6. Вычисление попарного логического произведения кодов 4 концов отрезка: $p= \sum_{i=1} T_1(i)T_2(i)$.
7. Определение полной невидимости отрезка: если p=1, то pr=-1,
   иначе pr=0.
8. Конец.

Логическую сумму можно вычислить или как арифметическую или как логическую сумму.

## ЛАБОРАТОРНАЯ РАБОТА N6
## ОТСЕЧЕНИЕ ПРОИЗВОЛЬНОГО МНОГОУГОЛЬНИКА ОКНОМ ПРОИЗВОЛЬНОЙ ФОРМЫ
### (на основе алгоритма Вейлера-Азертона)

Цель работы: изучение и программная реализация алгоритма Вейлера-Азертона отсечения многоугольников.

Данный алгоритм позволяет отсекать невыпуклые многоугольники с внутренними отверстиями отсекателем невыпуклой формы, также содержащим отверстия.

Границы многоугольников, получаемых в результате отсечения, совпадают либо с границами исходного многоугольника, либо с участками границ отсекателя. Никаких других новых границ не возникает.

Для реализации отсечения следует описать каждую границу отсекаемого многоугольника и отсекателя в виде циклического списка их вершин. При этом

внешние границы многоугольников должны обходиться по часовой стрелке, а внутренние границы - против часовой стрелки. Это приводит к тому, что при обходе вершин многоугольника в порядке их следования внутренняя область будет расположена справа от границы. Таким образом, всегда имеется минимум два списка вершин.

Границы отсекаемого многоугольника и отсекателя могут пересекаться. Алгоритм требует нахождения всех точек пересечения границ многоугольников. Координаты точек пересечения можно найти, решив простейшую систему двух уравнений с двумя неизвестными следующего вида:

$X_1+(X_2-X_1)t=X_3+(X_4-X_3)s$

$Y_1+(Y_2-Y_1)t=Y_3+(Y_4-Y_3)s$

где $(X_1,Y_1)$, $(X_2,Y_2)$ - координаты концов ребра одного многоугольника;

$(X_3,Y_3)$, $(X_4,Y_4)$ - координаты концов ребра другого многоугольника;

t,s - параметры, причем $0<t<1$, $0<s<1$.

Найденные значения параметров t,s должны удовлетворять указанным условиям, так как только в этом случае пересекаются действительно ребра многоугольников, а не их продолжения.

Найденные точки пересечения необходимо отсортировать на две группы: в одну войдут точки входа, в другую - точки выхода.

Точка пересечения является точкой входа, если в ней ребро отсекаемого многоугольника входит внутрь отсекателя; точка пересечения является точкой выхода, если в ней ребро многоугольника выходит из отсекателя. Общее количество точек пересечения должно быть четным, так как эти точки образуют пары: каждой входной точке соответствует точка выхода.

Для распознавания характера точки пересечения можно воспользоваться векторным произведением векторов, построенных на ребрах многоугольников. Если произведение вектора ребра отсекаемого многоугольника на вектор ребра отсекателя положительно, то их точка пересечения является точкой входа, в противном случае - точкой выхода.

Найденные точки пересечения необходимо добавить в ранее созданные циклические списки вершин отсекаемого и отсекающего многоугольников.

Далее можно приступить к собственно отсечению, которое будет сводиться к просмотру созданных списков границ. При этом можно отметить универсализм рассматриваемого алгоритма, позволяющего производить как внутреннее, так и внешнее отсечение. В первом случае находятся многоугольники (являющиеся частями исходного многоугольника), лежащие внутри отсекателя, во втором случае - многоугольники, лежащие за пределами отсекателя.

Для выполнения внутреннего отсечения выбирается очередная точка из списка входных точек. Эта точка находится в списке границ отсекаемого многоугольника, после чего просматривается список вершин этого многоугольника до встречи с точкой пересечения. В этом случае осуществляется переход к списку границ отсекателя, содержащему одноименную точку пересечения. Список вершин отсекателя также просматривается от начала к концу до встречи с точкой пересечения, после чего осуществляется возврат к списку вершин отсекаемого многоугольника. Процесс завершается в момент нахождения начальной точки пересечения.

Внешнее отсечение выполняется аналогично, но начальная точка выбирается из списка выходных точек пересечения, а просмотр списков вершин отсекателя производится в обратном порядке - от конца к началу. Следует заметить, что если очередная точка входа (выхода) ранее уже попала в список вершин результирующего многоугольника, то просмотр списков вершин отсекаемого многоугольника и отсекателя, начиная с этой точки, новых многоугольников не даст (получится один из уже рассмотренных многоугольников).

Реализуя вышеописанную последовательность действий, получим правильный результат только в том случае, если все границы отсекаемого и отсекающего многоугольников пересекаются. Если же нет пересечения границ, то необходимо установить соответствие между полученной границей (она будет внешней или внутренней) и границей самого многоугольника или отсекателя, которая будет являться результирующей внутренней или внешней границей.

Границы многоугольника, лежащие внутри отсекателя, будут являться границами результирующего внутреннего многоугольника. Границы, лежащие вне отсекателя, будут являться границами получаемых внешних многоугольников. Границы отсекателя, попавшие внутрь многоугольника, образуют в нем отверстия. В итоге границы отсекающего многоугольника должны быть помещены в оба списка принадлежности: внутренний и внешний.

Формально алгоритм Вейлера-Азертона можно записать следующим образом:

1. Ввод количества вершин внешней границы отсекаемого многоугольника и их координат.
2. Ввод количества отверстий в отсекаемом многоугольнике, по каждому отверстию - количества вершин и их координат.
3. Ввод количества вершин внешней границы отсекателя и их координат.
4. Ввод количества отверстий в отсекателе, по каждому отверстию - количества вершин и их координат.
5. Формирование кольцевых двунаправленных списков вершин по всем границам отсекаемого многоугольника.
6. Формирование кольцевых двунаправленных списков вершин по всем границам отсекателя.
7. Вычисление координат всех точек пересечения отсекаемого многоугольника и отсекателя.
8. Добавление всех найденных точек пересечения на соответствующие места в списки вершин многоугольников.
9. Определение типа каждой точки пересечения и формирование двух списков - точек входа и точек выхода.
10. Определение границ многоугольников, не имеющих пересечений. Границы отсекаемого многоугольника, лежащие вне отсекателя, поместить в список внешней принадлежности, границы, попавшие внутрь отсекателя - в список внутренней принадлежности. Поместить границы отсекателя, попавшие внутрь отсекаемого многоугольника, в оба списка принадлежности. (Границы отсекателя, лежащие за пределами отсекаемого многоугольника, проигнорировать).
11. Проведение собственно отсечения. Организация для этого цикла по всем точкам входа.
12. Нахождение в списке вершин отсекаемого

многоугольника очередной точки входа.

13. Просмотр списка вершин отсекаемого многоугольника до нахождения точки пересечения. Копирование просмотренных вершин в список внутренней принадлежности.

14. Переход к списку вершин отсекателя и поиск в нем одноименной точки пересечения.

15. Просмотр списка вершин отсекателя до нахождения точки пересечения. Копирование просмотренных вкршин в список внутренней принадлежности.

16. Сравнение найденной точки пересечения с первой точкой: если эти точки не совпадают, то преход к п. 13, иначе к п. 17.

17. Определение необходимости формирования второй границы отсеченного многоугольника: если не все границы отсекаемого многоугольника имеют пересечение с границами отсекателя, то необходим поиск другой границы, иначе перход к п.19.

18. Получение второй границы результирующего многоугольника. В этом случае можно использовать правило: если пересечение имела только внешняя граница отсекаемого многоугольника, то полученный результат является внешней границей. Внутренняя граница будет совпадать с внутренними границами самого многоугольника и отсекателя (если она лежит внутри отсекаемого многоугольника), если отверстия отсекаемого многоугольника не лежат внутри отверстий отсекателя. В противном случае внутренняя граница будет совпадать с границей отверстия отсекателя.

Если пересечение имела только внутренняя граница с внешней границей отсекателя, то будет получен нужный результат. Если же пересечение имели внутренние границы многоугольников, то внешняя граница результирующего многоугольника совпадает с внешней границей исходного многоугльника (если эта граница лежит внутри отсекателя) или с внешней границей отсекателя (отсекатель лежит внутри отсекаемого многоугольника).

19. Конец цикла (выбор следующей точки входа из списка, если он не пуст и переход к п. 12, иначе выход из цикла).

20. Конец.

Для проведения внешнего отсечения цикл, начинающийся в п. 12, следует проводить по всем точкам из списка выходов, а просмотр списков границ отсекателя производить в обратном порядке, а в правиле п. 18 внутренние и внешние границы меняются местами.