

И. А. АЙДЕМИРОВ, Ш.-М. А. ИСМАИЛОВ, В. М. ХАЧУМОВ
(Махачкала)

ОБ ОДНОМ АЛГОРИТМЕ ОТСЕЧЕНИЯ ПЛОСКИХ МНОГОУГОЛЬНИКОВ

Отсечение плоских многоугольников является распространенным видом преобразования в машинной графике [1—3]. Оно дает возможность отображать на экране дисплея видимые отрезки графического объекта, находящиеся в определенной области, называемой окном. Часто используются окна прямоугольной формы со сторонами, параллельными осям координат, что облегчает задачу отсечения. Программная реализация подобной процедуры дается в [1], а в [4] предлагается структура соответствующего конвейерного спецпроцессора. Однако задача существенно усложняется, когда окно, в свою очередь, задается многоугольником общего вида. В известных алгоритмах [2, 3] решение ее сводится к нахождению точек пересечения ребер многоугольников и проверке, содержатся ли вершины одного многоугольника внутри другого. Предлагаемые алгоритмы громоздки и, к сожалению, плохо приспособлены для аппаратной реализации. Кроме того, они предполагают разнесение во времени процессов получения векторного описания результата пересечения и его преобразования в растровую форму для вывода на экран. В настоящей работе рассматривается алгоритм, совмещающий процессы отсечения и генерации точек видимого графического объекта. Такой подход позволяет заменить процедуру поиска точки пересечения отрезков прямых на более простую процедуру определения принадлежности генерируемой точки ребру многоугольника. Он становится возможным благодаря широкому использованию вычислительных алгоритмов Д. Волдера [5], допускающих эффективную аппаратную реализацию.

Основные предложения по растровой развертке векторов рассматривались авторами в [6]. Пусть плоский многоугольник задается начальной точкой $[X_0, Y_0]$ и последовательностью двоек $(\Delta X_i, \Delta Y_i)$; $i = \overline{1, N}$, где $\Delta X_i, \Delta Y_i$ — горизонтальная и вертикальная компоненты i -го ребра (вектора), причем начало первого ребра и конец N -го совпадают с начальной точкой. Тогда координаты текущей точки k -го ребра многоугольника можно записать в параметрическом виде:

$$S(X, t) = X_0 + \sum_{i=1}^{k-1} \Delta X_i + \frac{\Delta X_k}{n_k} (t - t_k); \quad (1)$$

$$S(Y, t) = Y_0 + \sum_{i=1}^{k-1} \Delta Y_i + \frac{\Delta Y_k}{n_k} (t - t_k). \quad (2)$$

Здесь t ($t_{k-1} < t < t_k$) — параметр; t_k — значение параметра, соответствующее верхней границе k -го ребра; $n_k = \sqrt{\Delta X_k^2 + \Delta Y_k^2}$. Из выражений (1)—(2) видно, что если предварительно вычислить величины приращений $DX_i = \Delta X_i/n_i$, $DY_i = \Delta Y_i/n_i$, $i = \overline{1, N}$, то генерация точек ребра многоугольника сведется к итерационной процедуре суммирования. В [6] показано, что величины DX_i, DY_i могут быть вычислены в два этапа с использованием итерационных алгоритмов Волдера «Вектор» и «Множительно-делительная». Приращения DX_i, DY_i находятся для всех ребер многоугольников заранее и включаются в описания исходного графического объекта и окна.

Другой составляющей предлагаемого алгоритма отсечения является процедура определения принадлежности генерируемой точки одному из

ребер другого многоугольника. Выберем в качестве критерия близости текущей точки $[X_i, Y_i]$ к некоторой границе многоугольника абсолютную величину разности между суммой длин векторов, соединяющих точку с концами $[X_1, Y_1]$, $[X_2, Y_2]$ граничного отрезка, и длиной этого отрезка. Действительно, когда значение выбранного критерия минимально, имеет место

$$|\sqrt{(X_2 - X_i)^2 + (Y_2 - Y_i)^2} + \sqrt{(X_1 - X_i)^2 + (Y_1 - Y_i)^2} - \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}| = 0, \quad (3)$$

откуда после возведения в квадрат и соответствующих преобразований получим

$$[(Y_2 - Y_i)(X_1 - X_i) - (X_2 - X_i)(Y_1 - Y_i)]^2 = 0.$$

Из аналитической геометрии на плоскости известно, что выражение, заключенное в квадратные скобки, обращается в нуль для точки $[X_i, Y_i]$, лежащей непосредственно на граничном отрезке. Таким образом, для любого генерируемого вектора теоретически может существовать только одна точка пересечения с заданной границей, отвечающая критерию (3).

Обсудим возможные способы реализации полученного критерия. Для вычисления выражений вида $\sqrt{\Delta X^2 + \Delta Y^2}$ можно воспользоваться алгоритмом Воулдера «Вектор». С учетом того, что длины ребер исходных многоугольников найдены заранее, потребуется двукратное обращение к алгоритму для каждой текущей точки. Такой подход следует признать нецелесообразным из-за сложности. Введем обозначение: $Z = \min(|\Delta X|, |\Delta Y|) / \max(|\Delta X|, |\Delta Y|)$. Тогда, используя разложение в биномиальный ряд, получим

$$\sqrt{\Delta X^2 + \Delta Y^2} = \max(|\Delta X|, |\Delta Y|) \times \sqrt{1 + Z^2} = \max(|\Delta X|, |\Delta Y|) (1 + Z^2/2 - Z^4/8 + Z^6/18 - \dots). \quad (4)$$

Рассмотрим приближение вида

$$\sqrt{\Delta X^2 + \Delta Y^2} \approx \max(|\Delta X|, |\Delta Y|) + 1/2 \min(|\Delta X|, |\Delta Y|). \quad (5)$$

Использование выражения (5) приводит к упрощенной модификации критерия (3):

$$\begin{aligned} & \max(|X_2 - X_i|, |Y_2 - Y_i|) + 1/2 \min(|X_2 - X_i|, |Y_2 - Y_i|) + \\ & + \max(|X_1 - X_i|, |Y_1 - Y_i|) + 1/2 \min(|X_1 - X_i|, |Y_1 - Y_i|) - \\ & - \max(|X_2 - X_1|, |Y_2 - Y_1|) - 1/2 \min(|X_2 - X_1|, |Y_2 - Y_1|) = 0. \end{aligned} \quad (6)$$

Заметим, что выражение (5) использовалось в [2] для управления яркостью луча векторного дисплея. Критерий (6) позволяет находить точные значения координат точек пересечения с ребрами, расположенными под углами 0° , 45° , 90° , ... к оси OX . Для ребер многоугольника, имеющих другой наклон, будет иметь место «параллелограмм ошибок», как это показано на рис. 1. Все точки, лежащие в области параллелограмма, считаются точками ребра. К сожалению, использование большего числа членов разложения (4) с целью уменьшения области ошибок не представляется возможным из-за их существенной нелинейности, что ведет к неоправданному увеличению объема вычислений. Критерий (6), на наш взгляд, может быть эффективно использован для отыскания начала участка генерируемого вектора, принадлежащего «параллелограмму ошибок». Далее должен под-

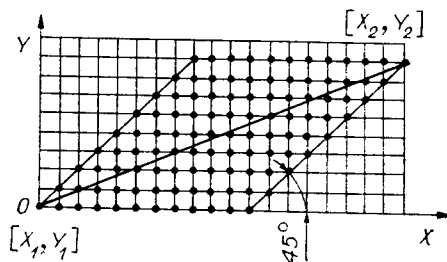


Рис. 1

ключаться более точный аппарат определения принадлежности ребру. Приведем выражение (3) путем дальнейших преобразований к виду

$$Y_i = Y_1 - \frac{Y_2 - Y_1}{X_2 - X_1}(X_1 - X_i). \quad (7)$$

С учетом того, что величины $\Delta X = X_2 - X_1$, $\Delta Y = Y_2 - Y_1$ найдены заранее, проверка условия (7) потребует выполнения двух операций алгебраического сложения: одной множително-делительной операции и одного сравнения. Таким образом, выражение (7) целесообразно использовать для точных вычислений.

Другой составляющей предлагаемого алгоритма отсечения является проверка принадлежности текущей точки области, заданной окном. Эта проверка осуществляется с помощью известного теста, основанного на подсчете числа пересечений генерируемой прямой линии с границами окна [2]. Особо отметим, что в предлагаемом алгоритме тестирование производится только один раз для начальной точки $[X_0, Y_0]$ многоугольника.

С учетом вышеизложенного алгоритм отсечения строится следующим образом. Для каждой генерируемой точки графического объекта производится проверка ее принадлежности одному из ребер окна с помощью условий (6) и (7). При выполнении критерия принадлежности осуществляются либо дальнейшая генерация и визуализация точек многоугольника, если они лежат в окне, либо переключение на визуализацию точек границы окна в противном случае. Таким образом, в процессе работы алгоритма окно и многоугольник могут меняться ролями. Одновременно с визуализацией усеченного многоугольника производится последовательная запись найденных точек пересечения, а также включенных в описание концов генерируемых векторов в специальный буфер, где и формируется векторное описание усеченного графического объекта. Алгоритм прекращает свою работу при совпадении координат первой и текущей точек буфера. Для правильного выполнения процедуры отсечения ребра многоугольников нумеруются единообразно, например по часовой стрелке. Укрупненная блок-схема алгоритма отсечения представлена на рис. 2. Здесь приняты следующие обозначения: $M1$ — число ребер окна; I, N — текущие номера ребер окна; $M1$ — число ребер многоугольника; J, M — текущие номера ребер многоугольника; $K1$ — признак принадлежности точки области окна; L — текущий номер точки буфера. Программная реализация алгоритма содержит около 200 операторов языка Пл/1. На рис. 3 и 4 показан пример отсечения многоугольника, выполненного путем моделирования работы данного алгоритма на ЭВМ. Для наглядности окно выбрано традиционно прямоугольным. Как показывают исследования, ошибки при определении точек пересечений могут накапливаться, поэтому целесообразно ввести коррекцию местоположения текущей точки, например при достижении конца генерируемого вектора. Заметим также, что программная реализация содержит механизм, отфильтровывающий «лишние» точки пересечения двух отрезков, которые могут возникнуть из-за ограниченной точности вычислений.

Вопросы аппаратной реализации алгоритмов Волдера, составляющих основу настоящего алгоритма отсечения, достаточно хорошо описаны в [5, 7]. Отметим лишь, что число итераций при выполнении операций «Вектор» и «Множително-делительная», обеспечивающее хорошее качество визуализации, достигает 12. Данное обстоятельство определяет величину разрядной сетки соответствующего вычислительного устройства с учетом необходимых дополнительных разрядов равной 16. Поскольку выполнение алгоритмов Волдера сводится к простым операциям сдвига и суммирования, их аппаратная реализация не вызывает затруднений. За основу, например, может быть принята структура специализированного устройства, предложенная в [7].

В заключение отметим следующие особенности разработанного алгоритма отсечения плоских многоугольников. Алгоритм позволяет едино-

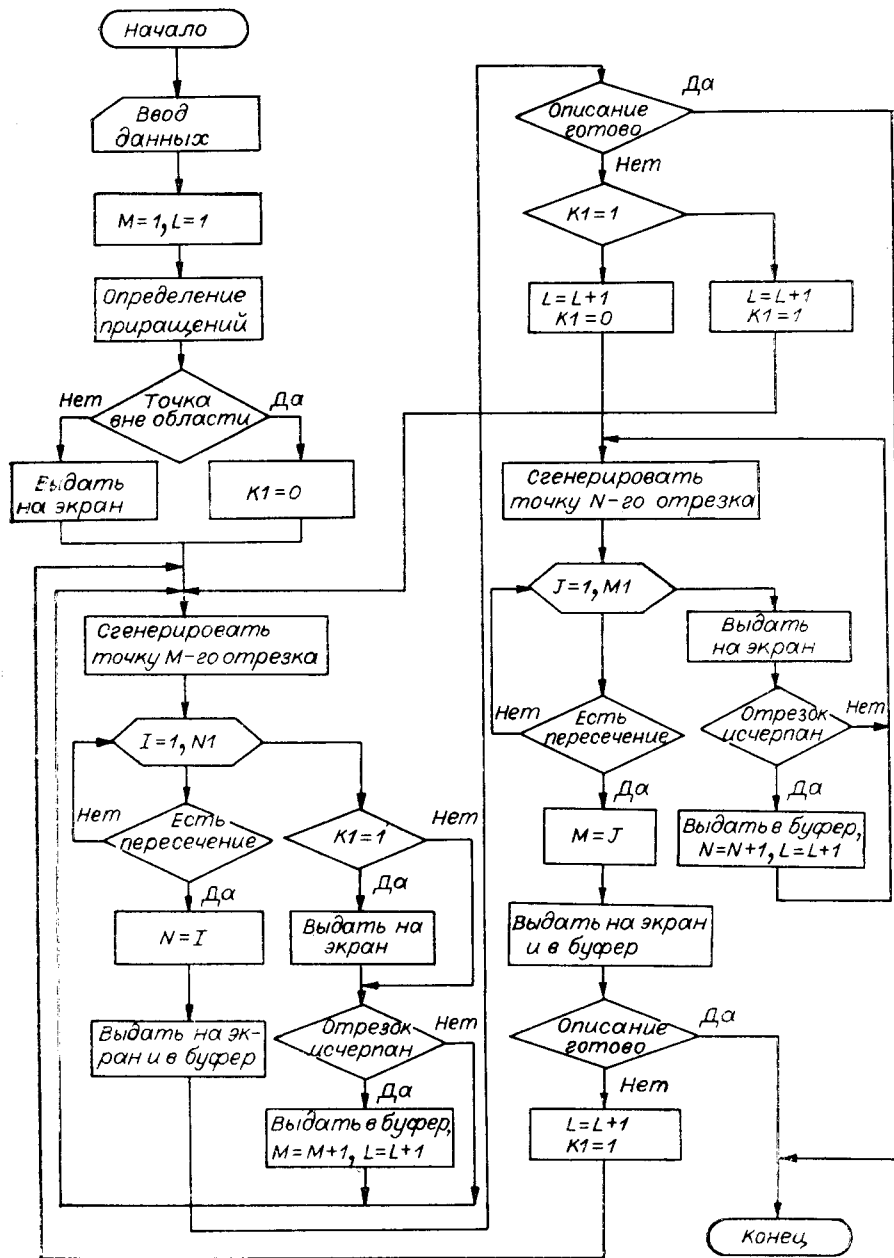


Рис. 2

образно описывать окна и преобразуемые графические объекты, что облегчает переход от одного описания к другому, обеспечивая формирование усеченного многоугольника за один проход. Формирование результирующего описания производится одновременно с процессом визуализации объекта на экране дисплея. Обеспечивается естественное включение в описание необходимых конечных точек ребер окна. За счет введения системы критериев, а также аппаратной реализации достигается быстрый просмотр графического объекта в окне произвольной конфигурации. В представленном виде алгоритм отсека не рассчитан на одновременное образование в окне двух и более изолированных многоугольников. Однако его доработка в этом направлении не вызывает затруднений. После замыкания в окне очередной изолированной фигуры необходимо вернуться в предшествующую точку буфера и продолжить генерацию

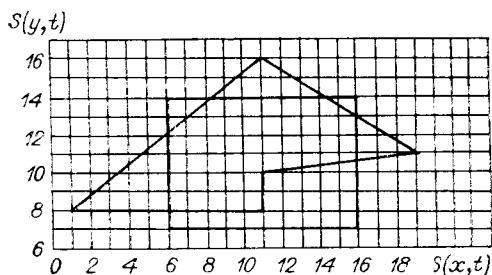


Рис. 3

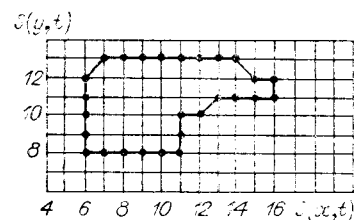


Рис. 4

исходного многоугольника. Окончание работы алгоритма совпадает в этом случае с возвратом в начальную точку преобразуемого графического объекта. Перечисленные особенности позволяют рекомендовать разработанный алгоритм отсечения для использования в специализированных графических системах реального времени.

СПИСОК ЛИТЕРАТУРЫ

1. Фолл Дж., ван Дэм А. Основы интерактивной машинной графики.— М.: Мир, 1985.— Кн. 2.
2. Гилой В. Интерактивная машинная графика: Структуры данных, алгоритмы, языки.— М.: Мир, 1981.
3. Павлидис Т. Алгоритмы машинной графики и обработки изображений.— М.: Радио и связь, 1986.
4. Кларк Д. Х., Дэйвис Т. Универсальная рабочая станция инженера-конструктора // Электроника.— 1983.— 56, № 20.
5. Байков В. Д., Смолов В. Б. Специализированные процессоры: Итерационные алгоритмы и структуры.— М.: Радио и связь, 1985.
6. Айдемиров И. А., Хачумов В. М., Шабалов Д. В. Применение алгоритмов Волдера для быстрого преобразования «вектор — растр» // Изв. вузов. Приборостроение.— 1988.— XXXI, № 7.
7. Батурицкий М. А. Применение аппаратного вычисления элементарных функций по алгоритму Волдера в СЦМ для обработки данных в реальном времени // Проектирование специализированных процессоров ЭВМ.— Л.: ЛГУ, 1986.— (Организация вычислительных структур и процессов.— Вып. 10).

Поступила в редакцию 28 ноября 1988 г.

УДК 681.3.06

А. В. БУШМЕЛЕВ, Е. П. КУЗЬМИН

(Москва)

СЖАТИЕ ИЕРАРХИЧЕСКИХ СТРУКТУР

Пространственные иерархические структуры в настоящее время широко применяются в системах машинной графики для представления трехмерной геометрической информации [1, 2]. Основным недостатком подобного представления является большой объем памяти, требующийся для хранения структур.

В работе предложены новые иерархические структуры — адаптивные бинарные деревья и бинарные сети, позволяющие значительно понизить количество узлов иерархических представлений.

Введение. Общая схема получения пространственных иерархических структур выглядит следующим образом. Рассмотрим мировое пространство — прямоугольный параллелепипед, содержащий трехмерную