РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я
№ 1 1995

УДК 681.3.06

А. М. Ковалев
(Новосибирск)

ПРОИЗВОДИТЕЛЬНОСТЬ ГЕНЕРАТОРОВ ИЗОБРАЖЕНИЙ С РЕКУРСИВНЫМ ПОИСКОМ ПИКСЕЛОВ

Рассматривается вычислительный конвейер для отображения на дискретный растр многоугольников, заданных в форме линейных уравнений и функций, путем рекурсивного деления плоскости изображений. Определяется количество окон, генерируемых многоугольником на каждом уровне подразделения плоскости изображений, и время выполнения этого объема работы. Приводятся оценки производительности и эффективности параллельного выполнения операций на уровнях подразделения для различных вариантов архитектур: процессора со стеком, конвейера процессоров, четверичного дерева процессорных элементов и комбинированных вариантов. Показано увеличение производительности генераторов изображений, использующих многоуровневое приоритетное маскирование для удаления скрытых поверхностей, что обеспечивает уникальную возможность синтеза в реальном времени изображений виртуальной среды, обладающей одновременно большой визуальной и глубинной сложностью.

Введение. В [1] предложено полностью векторизованное проективное преобразование трехмерных поверхностей, генерирующее описание двумерных примитивов в виде линейных уравнений и функций, в [2] — их антиэлайзинговое отображение на дискретный растр путем рекурсивного деления плоскости изображений.

Предметом настоящей работы являются оценка производительности и эффективности ряда процессоров с параллельным выполнением операций на уровнях подразделения плоскости изображений, а также исследование метода многоуровневого приоритетного маскирования для повышения производительности при увеличении глубинной сложности изображений.

Обобщенная схема растрового процессора. Синтез изображений виртуальной среды осуществляется вычислительным конвейером, содержащим геометрический, растровый и пиксельный процессоры (рис. 1, *а*).

Геометрический процессор обеспечивает проективное преобразование графических данных из объектного пространства в пространство изображений. Описание поверхностей объектов, спроецированных на плоскость изображения в виде набора выпуклых многоугольников, задается пересечением полуплоскостей, ограниченных прямыми линиями, и содержит коэффициенты уравнений этих линий вида $Ax + By + C = 0$ и коэффициенты линейных функций, определяющие параметры поверхностей (цвет, координаты текстурного рисунка и др.) в виде отношения $(Ax + By + C)/(Dx + Ey + F)$.

Растровый процессор обеспечивает антиэлайзинговое отображение многоугольников на дискретный растр, во-первых, путем поиска всех координатных позиций растра, таких что апертура фильтра, ассоциированная с каждой из них, оказывается накрытой областью многоугольника частично или полностью, и, во-вторых, вычисления параметров поверхности для найденных пикселов и данных для дискретного фильтра. При использовании приоритетного маскирования растровый процессор обеспечивает также удаление невидимых поверхностей.

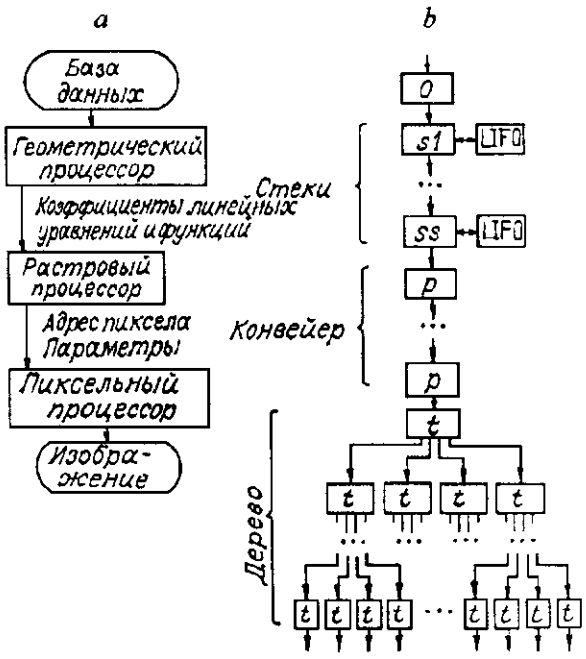

Рис. 1

Пиксельный процессор выполняет завершающие вычисления, фильтрацию и запоминание цвета в растровой видеобуферной памяти.

На рис. 1, *b* приведена обобщенная схема растрового процессора в виде вычислительного конвейера, содержащего блок 0 начальной установки, *s* стеков, *p*-каскадный конвейер и *t*-уровневое четверичное дерево.

Блок начальной установки для окна подразделения, равного максимальному размеру изображения, для каждой реберной линии многоугольника вычисляет, во-первых, начальное значение функции покрытия вида

$$d_0 = Ax_c + By_c + C + [|A|(2^n - 1 + \alpha) + |B|(2^n - 1 + \beta)]/2,$$

где A, B, C — коэффициенты уравнений реберных линий; x_c, y_c — координаты центра изображения; $2^n \times 2^n$ — разрешение дискретного растра; $\alpha \times \beta$ — апертура фильтра, и, во-вторых, для каждого параметра поверхности — его начальное значение вида

$$f_0 = D(x_c - 2^{n-2}) + E(y_c - 2^{n-2}) + F,$$

где D, E, F — коэффициенты линейной функции параметра.

Процессорные элементы стека, конвейера и дерева делят поступившее на обработку окно на четыре равных подокна, для каждого подокна определяют наличие пересечения области подокна и области многоугольника с помощью теста покрытия и в случае положительного результата теста поразрядно вычисляют функцию параметров. Функции покрытия, необходимые для теста на i-м уровне подразделения, находятся из рекуррентного соотношения

$$d_i(x_{n-i}, y_{n-i}) = d_{i-1} - 2^{n-i}|A|e_{iA} - 2^{n-i}|B|e_{iB},$$

где $x_{n-i}, y_{n-i} \in (0, 1)$ — двоичный адрес подокна; e_{iA}, e_{iB} — логические функции адреса и знаков коэффициентов A, B. Для центра накрытых подокон функции параметров вычисляются в виде

$$f_i(x_{n-i}, y_{n-i}) = f_{i-1} + 2^{n-i}Dx_{n-i} + 2^{n-i}Ey_{n-i}.$$

риведенные соотношения для функций покрытия и параметров найдены в [2].

Процессорный элемент стека обрабатывает несколько уровней подразде-ения плоскости изображения, помещая необработанные окна предыдущего ровня в память типа LIFO (Last In — First Out). Процессорный элемент онвейера обрабатывает один уровень подразделения, последовательно выда-ая накрытые подокна на выход. Процессорный элемент четверичного дерева акже обрабатывает один уровень подразделения, но обеспечивает адресную ередачу накрытых подокон четырем процессорным элементам следующего ровня.

В зависимости от типа и количества используемых процессорных элемен-тов растровый процессор может иметь разную архитектуру, которую в общем случае можно обозначить как

$$\text{arch} = a_1 s_1 a_2 s_2 \ldots a_s s_s bpct, \qquad (1)$$

причем $\sum_j a_j + b + c = n$, где $a_j > 2$ — число уровней подразделения, обраба-тываемых стеком s_j; b — число каскадов конвейера; $c \geq 2$ — число уровней етверичного дерева; n — общее число уровней.

Примечание. Для упрощения обозначений будем опускать индекс j при $s = 1$, а также символы s, p, t при a_j, b, $c = 0$.

Далее по тексту три основные архитектуры будем называть: ns — «сте-ком»; np — «конвейером»; nt — «деревом», а комбинированные варианты — с учетом примечания в соответствии с обозначением (1). Например, $9s1p$ — стек на девять начальных уровней подразделения и конвейерный элемент на деся-м уровне; $4s_1 3s_2 1p2t$ — архитектура, содержащая два стека, один конвейер-ный элемент, обрабатывающий восьмой уровень подразделения, и четверич-ное дерево на два последних уровня.

Число окон, генерируемых многоугольником на i-м уровне. Каждый i-й ровень подразделения плоскости изображения ($1 < i < n$) можно пред-тавить в виде совокупности окон, покрывающих эту плоскость. Центры окон бразуют дискретный растр (целочисленную решетку) с $2^i \times 2^i$ точками рис. 2). Расстояния между точками этого растра вдоль осей x и y одинаковы и равны 2^{n-i} единиц. Размеры окна составляют $(2^{n-i} - 1 + \alpha) \times (2^{n-i} - 1 + \beta)$ кв. ед.

При такой интерпретации i-го уровня задача отыскания числа окон, ге-нерируемых многоугольником, сводится к определению площади фильтрован-ного многоугольника [3] с апертурой фильтра, равной тносительным размерам ок-а. На рис. 2 область фильтро-анного многоугольника огра-ичена штриховыми лини-ми.

Число точек растра, попа-ющих в область фильтро-нного многоугольника, рав-о числу окон, накрытых мно-оугольником полностью или стично, в среднем равно лощади фильтрованного ногоугольника и может быть енено как

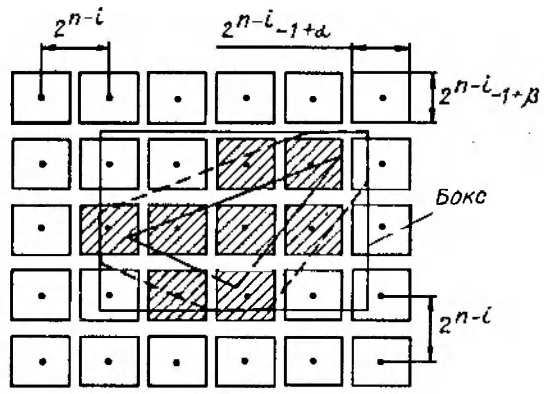

Рис. 2

$$w_i \le S_i + L\sqrt{(\alpha_i^2 + \beta_i^2)\pi S_i} + \alpha_i \beta_i,$$

где w_i — искомое число окон; S_i — площадь многоугольника, измеренная на i-м растре; L — коэффициент формы многоугольника; α_i, β_i — относительные размеры окна. (Эта оценка является обобщением выражения (9), приведенного в [3], для фильтра с квадратным основанием.) С учетом расстояния между точками i-го растра число окон принимает вид

$$w_i(S) \le 4^{i-n}S + 2^{i-n}L\sqrt{(\alpha_i^2 + \beta_i^2)\pi S} + \alpha_i \beta_i, \qquad (2)$$

где S — площадь многоугольника, измеренная в количестве пикселов, а относительные размеры окна равны

$$\alpha_i = 1 + 2^{i-n}(\alpha - 1), \qquad \beta_i = 1 + 2^{i-n}(\beta - 1), \qquad (3)$$

где $\alpha \times \beta$ — апертура фильтра на пиксельном n-м уровне.

В [4] показано, что на острых углах многоугольников могут образоваться дополнительные площади, что увеличит число окон по сравнению с (2). Одним из способов отсечения дополнительных окон является помещение отображаемого многоугольника в прямоугольный бокс, как показано на рис. 2, причем число ребер бокса может быть ограничено количеством острых углов.

Число окон (2) можно считать функцией двух переменных: уровня подразделения и площади многоугольника. Апертура фильтра является конструктивным параметром генератора изображений, обеспечивающим антиэлайзинг; коэффициент формы является константой для некоторого «среднего» многоугольника, выбранного в качестве типового тестового.

Для современных средств воспроизведения изображений с разрешением порядка 1000×1000 пикселов достаточно десяти уровней подразделения. Диапазон изменения площади многоугольника от 1 до 1000 пикселов актуален как для перспективных генераторов ($1 < S < 100$), способных синтезировать в реальном времени с частотой 50 Гц до сотен тысяч многоугольников, так и для существующих генераторов ($100 < S < 1000$) с производительностью до десяти тысяч многоугольников на кадр. Апертура фильтра для средств воспроизведения на ЭЛТ с прогрессивной разверткой обычно выбирается квадратной с параметрами $\alpha = \beta = 1, 2, 3$. Для чересстрочной развертки TV-мониторов иногда используют прямоугольную апертуру с $\alpha \times \beta = 1 \times 2$ или 2×3. И наконец, типовым тестовым многоугольником для машинной графики можно принять прямоугольник с различным соотношением сторон.

На рис. 3, *а* приведены зависимости $w_n(S) = S_f = S + \alpha L\sqrt{2\pi S} + \alpha^2$ для фильтрованных с квадратной апертурой $\alpha = 0, 1, 2, 3$ прямоугольников с соотношением сторон 1/1, 5/1 и 10/1, повернутых на 45° по отношению к координатным осям изображения. Можно заметить, что относительное число окон S_f/S значительно увеличивается в области малых S за счет фильтрации. Коэффициент формы L не оказывает такого существенного влияния.

При уменьшении номера уровня подразделения в (2) от n до 1 площадь многоугольника S_i уменьшается в 4 раза при переходе на каждый последующий уровень, достигая ничтожно малых величин на первых уровнях подразделения. Какова бы ни была апертура фильтра на пиксельном уровне, при уменьшении номера уровня относительные размеры окна из (3) и их произведен в (2) стремятся к единице. При $\alpha, \beta = 1$ дополнительная площадь в (2), зависящая от формы многоугольника L, уменьшается вдвое при переходе на каждый последующий уровень подразделения; $\alpha, \beta > 1$ ускоряют это уменьшение $\alpha, \beta < 1$ замедляют. Можно утверждать, что для алгоритмов с фильтрацией т. е. при $\alpha, \beta > 1$,

$$w_1(S) > 1, \qquad w_{i+1}(S) > w_i(S) > \tfrac{1}{4} w_{i+1}(S). \qquad (4)$$

При $\alpha, \beta < 1$ неравенство может нарушаться в области малых S.

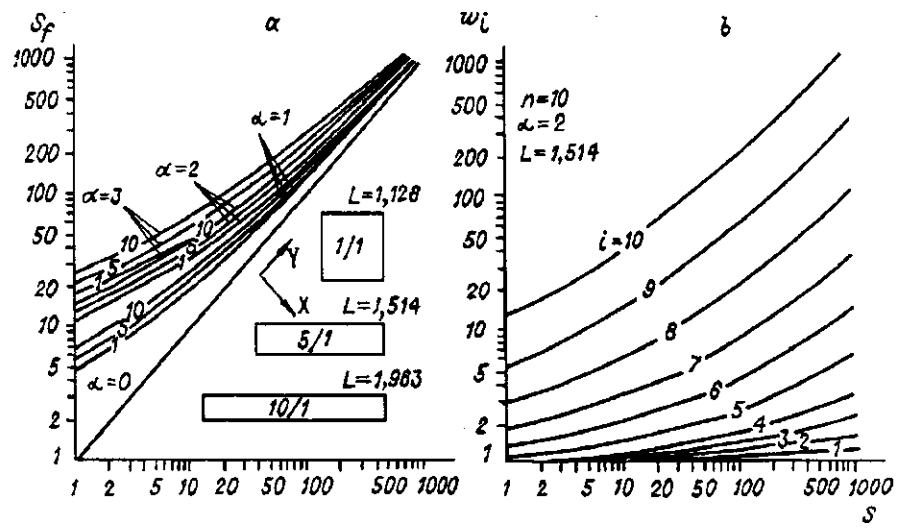

Рис. 3

Следует заметить, что число окон из (2) является предельным средним на множестве многоугольников, одинаковых по площади и форме, но разных по расположению на растре. Поэтому не должно казаться странным, что многоугольники генерируют вещественные, а не целые числа окон. Рис. 3, *b* иллюстрирует зависимость $w_i(S)$ при $1 \le i \le 10$, $\alpha = 2$, $L = 1{,}514$, которая фактически определяет объем работ на i-х уровнях подразделения плоскости изображений.

Время обработки многоугольника на i-м уровне складывается из двух величин:

$$t_i(S) \le 4t_e w_{i-1}(S) + t_p w_i(S), \tag{5}$$

где $1 \le i \le n$; $w_0(S) = 1$; t_e — время теста покрытия одного из четырех подокон окна; t_p — время вычисления параметров одного подокна.

Вычислительные затраты для определения функции покрытия сводятся в среднем к одному сложению на одно ребро многоугольника на подокно. Вычисление линейной функции параметров требует также в среднем одного сложения на один параметр на подокно [2]. Таким образом, время выполнения теста покрытия в среднем равно $t_e \cong k_e\tau$, где τ — время операции сложения; k_e — число ребер многоугольника с учетом бокса; время вычисления параметров равно $t_p \cong k_p\tau$, где k_p — число параметров поверхности. С учетом этого выражение (5) можно записать в виде

$$t_i(S) \le t_e[4w_{i-1}(S) + w_i(S)k_p/k_e]. \tag{6}$$

Отношение k_p/k_e может иметь разную величину в зависимости от конкретной реализации генератора изображений. При вычислении параметров вне растрового процессора $k_p/k_e = 0$. При обработке среднего четырехугольника, имеющего до семи параметров, таких как координата дальности z, компоненты цвета r, g, b, α, текстурные координаты u, v, без применения теста на полное покрытие отношение $k_p/k_e \cong 1$. При использовании теста на полное покрытие отношение k_p/k_e может иметь разную величину на разных уровнях подразделения из-за изменения числа обрабатываемых ребер многоугольника.

Производительность и средняя загрузка растрового процессора. Производительность генераторов изображений обычно оценивается числом многоугольников площади S, обрабатываемых в единицу времени. Производительность растрового процессора как вычислительного конвейера равна про-

изводительности каскада (каскадов) конвейера с наибольшим временем обработки многоугольника площади S. Из (6) аналогично (4) для алгоритмов с фильтрацией

$$t_{i+1}(S) > t_i(S) > \frac{1}{4} t_{i+1}(S). \tag{7}$$

Это неравенство дает возможность ранжировать производительность $P_{\text{arch}}(S)$, где индекс обозначает архитектуру согласно (1) в порядке убывания следующим образом:

$$\begin{gathered}
\text{1) дерево } P_{tr}(S) \geq 1/t_1(S); \\
\text{2) конвейер–дерево } P_{ip(n-i)t}(S) \geq 1/t_{i+1}, \quad \text{где } 1 \leq i \leq n-2; \\
\text{3) конвейер } P_{np}(S) \geq 1/t_n(S); \\
\text{4) стек } P_{ns}(S) \geq 1/\sum_i t_i(S).
\end{gathered} \tag{8}$$

Производительность полного четверичного дерева процессорных элементов определяется временем обработки на первом уровне подразделения, производительность процессора типа конвейер — дерево — номером уровня подразделения, определяющим корень частичного дерева; производительность конвейера соответствует времени работы на пиксельном n-м уровне; производительность стекового процессора определяется суммарным объемом работы по отображению многоугольников.

Зависимости относительной производительности для ряда архитектур растрового процессора $p(S) = P_{\text{arch}}(S)/P_{\max}$, где $P_{\text{arch}}(S)$ соответствует (8), а $P_{\max} = 1/(4t_e + t_p)$ из (5), показаны на рис. 4, *а* для $n = 10$, $\alpha = 2$, $L = 1{,}514$, $k_p/k_e = 1$. На этом же рисунке дана относительная производительность стекового процессорного элемента $9s$, который совместно с конвейерным элементом на 10-м уровне образует процессор типа $9s1p$.

Среднюю загрузку растрового процессора или показатель эффективности мультипроцессорной реализации можно определить как

$$R(S) = P_{\text{arch}}(S)/[N_{\text{arch}} P_{ns}(S)], \tag{9}$$

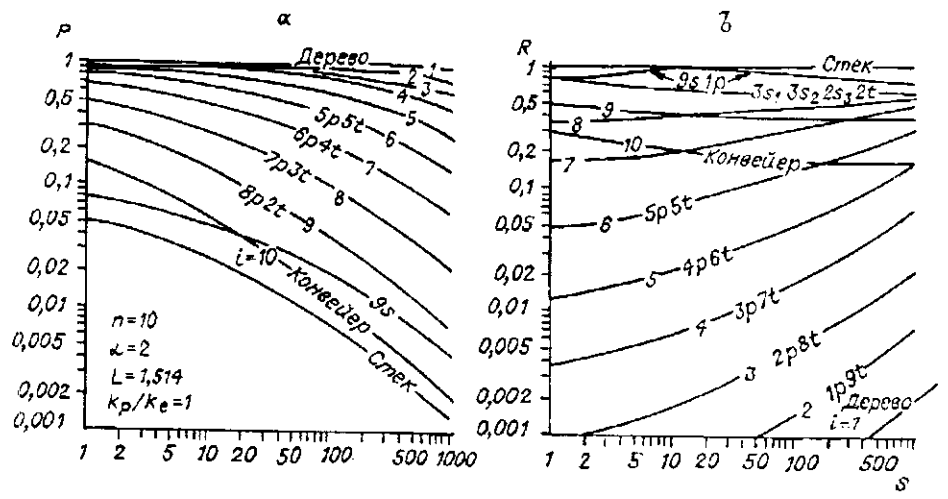

Рис. 4

где $N_{arch}(S)$ — число процессорных элементов, равное

$$N_{arch} = i - 1 + (4^{n-i+1} - 1)/3 \qquad (10)$$

при $1 \le i \le n$.

На рис. 4, *b* показаны зависимости $R(S)$ для рассматриваемых архитектур, а в таблице приведено число элементов N_{arch}.

i	Архитектура	N_{arch}
1	$10t$	$3{,}495 \cdot 10^5$
2	$1p9t$	$8{,}738 \cdot 10^4$
3	$2p8t$	$2{,}185 \cdot 10^4$
4	$3p7t$	$5{,}464 \cdot 10^3$
5	$4p6t$	$1{,}369 \cdot 10^3$
6	$5p5t$	346
7	$6p4t$	91
8	$7p3t$	28
9	$8p2t$	13
10	$10p$	10

Процессор со стеком выполняет весь объем работ на одном процессорном элементе, не имеет простоев, но обладает минимальной производительностью. Производительность конвейера определяется последним каскадом, выполняющим работу пиксельного уровня без простоев. Остальные каскады конвейера вынуждены работать с простоями, что и определяет среднюю загрузку конвейера на уровне 20—30 %. В результате производительность конвейера из 10 процессорных элементов превышает производительность стека всего лишь в 1,5—3 раза. Полное четверичное дерево обладает максимальной производительностью. При этом время обработки многоугольника произвольной площади S примерно одинаково и минимально. Однако эффективность такой архитектуры настолько мала, что средняя загрузка процессорных элементов составляет десятые и даже сотые доли процента. Наиболее эффективными являются мультипроцессоры типа $8p2t$ и $7p3t$, которые имеют загрузку на уровне 40—50 %, что и обеспечивает повышение производительности по сравнению со стеком в 5—14 раз.

Одним из способов балансировки нагрузок каскадов растрового процессора является применение стековых процессорных элементов. На рис. 4 приведены показатели процессора $9s1p$ со средней загрузкой 80—100 % и производительностью, которая определяется стеком $9s$ в диапазоне S от 1 до 20 пикселов и конвейером $10p$ при $S > 20$ пикселов. Другой пример: процессор $3s_1 3s_2 2s_3 2t$ обеспечивает одинаковую производительность с упомянутым выше процессором $8p2t$ во всем диапазоне S, содержит 8 процессорных элементов вместо 13 и имеет загрузку на уровне 60—80 %.

Многоуровневое приоритетное маскирование. Метод удаления скрытых поверхностей с помощью масок занятых окон описан в [2]. Многоугольники, поступающие на обработку в порядке прямого приоритета, могут быть удалены на любом уровне подразделения по данным 4-разрядной маски (один разряд на каждое подокно окна). Разряд маски устанавливается, если подокно полностью покрыто либо одним непрозрачным многоугольником, либо совокупностью непрозрачных многоугольников, что передается по обратной связи в том случае, если заполнены четыре дочерних подокна. Подокна пиксельного уровня устанавливаются занятыми, если заполнены их субпиксельные маски.

Метод многоуровневого приоритетного маскирования можно с успехом использовать в системах с жестким ограничением времени обработки кадра изображения. Например, для синтеза виртуальной среды в реальном времени с частотой кадров 50 Гц время обработки многоугольников не должно превышать $t_0 = 20$ мс. Производительность таких систем измеряется количеством многоугольников на кадр при оговоренной глубинной сложности изображений. Представляет интерес исследовать изменение производительности растрового процессора с маскированием при изменении глубинной сложности изображений ζ, под которой в данном случае понимается отношение суммарной площади многоугольников, обработанных за заданное время t_0, к площади плоскости изображения Σ_0.

Пусть $P(S)$ — производительность из (8) процессора без маскирования — и за время t_0 процессор обрабатывает F_0 многоугольников площади S_0, таких что плоскость изображения заполняется с $\zeta = 1$, при этом

$$\Sigma_0 = F_0 S_0 = P(S_0) t_0 S_0. \tag{11}$$

Для того чтобы увеличить глубинную сложность ζ и вовлечь удаление невидимых поверхностей, необходимо заполнить площадь Σ_0 за время $t < t_0$ многоугольниками площади $S > S_0$, т. е.

$$\Sigma_0 = F_v(S) S = P(S) t S, \tag{12}$$

где $F_v(S)$ — число видимых непрозрачных многоугольников, при отображении которых все имеющиеся в процессоре маски устанавливаются в состояние «Занято». Из (11) и (12), во-первых,

$$F_v(S) = F_0 S_0 / S \tag{13}$$

и, во-вторых,

$$t/t_0 = [S_0 P(S_0)]/[S P(S)]. \tag{14}$$

Поскольку изображение заполнено до $\zeta = 1$, за время $(t_0 - t)$ процессор будет удалять с помощью масок последующие невидимые многоугольники с производительностью $P_H(S) \geq P(S)$. При этом будет обработана площадь Σ_H скрытых многоугольников F_H:

$$\Sigma_H = F_H S = P_H(S)(t_0 - t) S,$$

которая обеспечит глубинную сложность изображения

$$\zeta(S) = (\Sigma_0 + \Sigma_H)/\Sigma_0 = 1 + [(t_0 - t) P_H(S)]/[t P(S)]$$

или с учетом (14)

$$\zeta(S) = 1 - P_H(S)/P(S) + [S P_H(S)]/[S_0 P(S_0)]. \tag{15}$$

Общее количество многоугольников F_T площади S, обработанное процессором с маскированием за заданное время t_0, или производительность процессора, учитывая (13), составит:

$$F_T(S) = F_v(S)\zeta(S) = F_0 \zeta(S) S_0 / S \tag{16}$$

на кадр изображения.

Полученные выражения (13), (15) и (16) определяют параметрическое задание функций $F_v(\zeta)$ и $F_T(\zeta)$, где переменным параметром является площадь многоугольника $S > S_0$.

На рис. 5 для $S_0 = 10$ и $10 < S < 1000$ показаны относительные величины $f_T(\zeta) = F_T(\zeta)/F_0$ и $f_v(\zeta) = F_v(\zeta)/F_0$ для конвейера без маскирования, конвейера с маскированием, охватывающим разные каскады конвейера при $10 > i > 2$, для дерева и стека. Производительности $P(S)$, необходимые для выражения (15), взяты из примера, приведенного на рис. 4, *а*. Следует заметить, что из (13) $f_v(S) = S_0/S$, поэтому ордината графика на рис. 5, *б* может быть проградуирована также и в величинах переменной S. Таким образом, рис. 5, *б* является одновременно и графиком функции $\zeta(S)$. Линии равных площадей на рис. 5, *а* показаны штриховыми линиями.

Среди процессоров без маскирования, для которых $P_H(S) = P(S)$, лучшие характеристики имеет четверичное дерево и худшие — конвейер, что легко объясняется видом зависимости $P(S)$. Для дерева $P_H(S) = P_м(S) \sim$ const, а для конвейера $P_H(S) = P_{пр}(S) \sim 1/S$, т. е. значительно падает с ростом площади S, и поэтому из (14) для дерева t/t_0 линейно уменьшается, а для конвейера

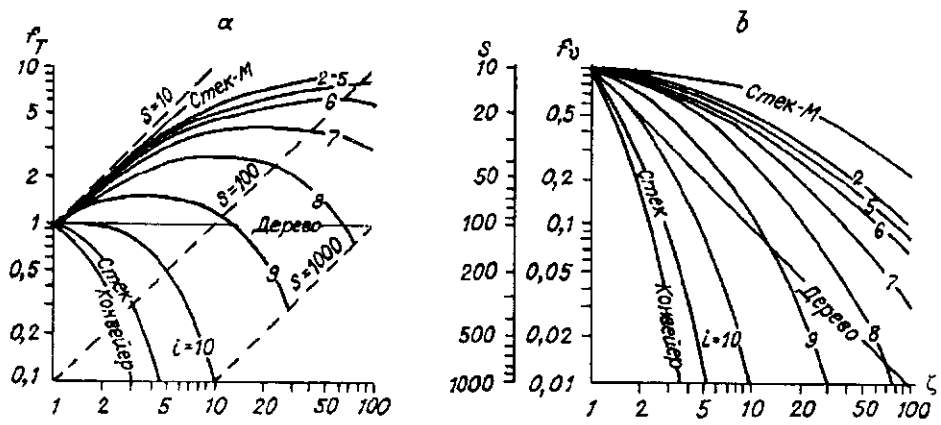

Рис. 5

$t/t_0 \sim 1$, т. е. труднее получить и использовать избыток времени $(t_0 - t)$ для увеличения глубинной сложности.

Однако в отличие от дерева конвейер имеет потенциальный запас производительности $P_H(S)$ на более высоких уровнях подразделения, чем пиксельный. Это очень важно, так как увеличение $\zeta(S)$ из (15) в основном зависит от отношения $P_H(S)/P(S_0)$. В конвейере с маскированием, если маски установлены «занятыми» на i-м уровне, производительность определяется $(i-1)$-м уровнем из-за того, что время проверки масок на i-м уровне несоизмеримо меньше тестирования и вычисления параметров на $(i-1)$-м уровне. На рис. 5, *а* показано увеличение производительности конвейера с ростом числа уровней, охваченных маскированием, причем при $i \geq 9$ производительность конвейера становится выше производительности дерева. Из рис. 5, *b* ясно, что при маскировании уменьшается падение числа видимых многоугольников f_v с ростом глубинной сложности ζ. Таким образом, конвейер с маскированием обладает лучшими возможностями для синтеза изображений виртуальной среды с высокой визуальной и глубинной сложностью.

Стековая архитектура, как это следует из рис. 5, обладает еще большими возможностями, поскольку имеет потенциально больший относительный запас производительности $P_H(S)/P(S_0)$ на начальных уровнях подразделения плоскости изображений.

Заключение. Значительная неравномерность распределения объема работ по уровням подразделения плоскости изображений в алгоритме рекурсивного поиска пикселов создает, с одной стороны, известные трудности построения высокопроизводительных и эффективных растровых процессоров с параллельным выполнением операций, но, с другой стороны, обеспечивает уникальные возможности для синтеза изображений виртуальной среды с высокой визуальной и глубинной сложностью при помощи многоуровневого приоритетного маскирования.

В рамках рассмотренного ряда архитектур растрового процессора наиболее эффективным является вычислительный конвейер смешанного типа: стек — конвейер — четверичное дерево на малое число уровней. Такая архитектура обладает достаточной производительностью для синтеза сложных изображений и имеет запас производительности для быстрого удаления невидимых поверхностей.

Направлением последующих работ по реализации алгоритма рекурсивного поиска пикселов является исследование разреженных растров и производительности матричных архитектур [2], т. е. подключение к параллельному выполнению операций параллелизма иного вида — распределения работы по плоскости изображения.

СПИСОК ЛИТЕРАТУРЫ

1. **Ковалев А. М., Тарнопольский Ю. В.** Проективное преобразование триангулированных поверхностей // Автометрия. 1991. № 1.
2. **Ковалев А. М.** Синтез виртуальной среды с рекурсивным делением плоскости изображения // Автометрия. 1993. № 5.
3. **Ковалев А. М., Токарев А. С.** К оценке производительности алгоритмов фильтрации синтезированных изображений // Автометрия. 1989. № 2.
4. **Котов Н. В., Курочкин В. В., Перебейнос С. В.** К оценке производительности одного из алгоритмов синтеза фильтрованных изображений // Автометрия. 1991. № 6.

Поступила в редакцию 25 февраля 1994 г.