

РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я

№ 5

1993

УДК 681.3.06

А. М. Ковалев

(Новосибирск)

СИНТЕЗ ВИРТУАЛЬНОЙ СРЕДЫ
С РЕКУРСИВНЫМ ДЕЛЕНИЕМ ПЛОСКОСТИ ИЗОБРАЖЕНИЯ*

Обсуждается проблема синтеза изображений фотореалистического качества в реальном времени для виртуальной среды. Рассматриваются новые алгоритмы проективного преобразования и отображения на дискретный растр поверхностей трехмерных объектов среды. Предложен рекурсивный алгоритм деления плоскости изображений для отображения граней поверхностей, заданных в форме линейных уравнений и функций. Для дискретных растров с учетом апертуры антиэлайсингового фильтра определены рекуррентные соотношения для вычисления параметров поверхностей и «функций покрытия», устанавливающих пересечение выпуклой грани-многоугольника и области подразделения плоскости изображения. Показана уникальная возможность многоуровневого приоритетного маскирования для удаления скрытых поверхностей. Приведен ряд мультипроцессорных архитектур для реализации предложенного алгоритма.

Введение. Трехмерная компьютерная графика, способная конкурировать по качеству синтезированных изображений с фотографиями реальных сцен, наряду с полноцветным видео-, стереозвук и геометрически заданным текстом, является составной частью бурно развивающейся области комплексного представления информации в персональных многосредних (multi-media) компьютерах [1].

Наиболее трудоемким и дорогим является процесс генерации в реальном времени синтетического трехмерного пространства или виртуальной среды, в которой мог бы двигаться и действовать оператор-наблюдатель. Виртуальная окружающая обстановка является коммуникационной средой [2], освоенной пока лишь в авиационных тренажерах [3, 4] и рабочих станциях с суперграфическими возможностями [1, 5]. Создание массового рынка и новых сфер приложения виртуальной среды невозможно без значительного снижения стоимости оборудования при одновременном увеличении его производительности. Опыт показывает, что удовлетворение этих противоречивых требований обеспечивается не только совершенством технологий проектирования и производства специализированных СБИС, но и алгоритмами синтеза изображений, допускающими высокоэффективную мультипроцессорную обработку геометрических примитивов.

Представляется, что в многосредних компьютеры будет перенесена хорошо изученная технология синтеза изображений, в которой вычислительный процесс организуется как каскад параллельно работающих процессов или процессоров, образующих вычислительный конвейер [3]. Традиционный конвейер решает пять известных подзадач: 1) создание базы графических данных, организация данных в структуру; 2) просмотр структуры и извлечение данных; 3) проективное преобразование графических данных из объектного пространства в пространство изображений; 4) отображение геометрических примитивов (точек, линий, многоугольников) на дискретный растр, вычисление

* Работа поддержана грантом Международного благотворительного фонда «Культурная инициатива».

и запоминание цвета каждого элемента изображения (пиксела) в растровой видеобufferной памяти; 5) воспроизведение изображений путем сканирования растровой памяти.

В рамках классической конвейерной схемы фотореалистическое качество изображений может быть заложено уже при формировании базы данных с применением метода излучательности [6]. Следует заметить характерное увеличение объема данных, так как даже плоские поверхности при этом необходимо делить на куски меньшей площади. Увеличению реализма способствует введение зеркальных и прозрачных поверхностей, использование фототекстуры. Чтобы обеспечить реальное время синтеза изображений фотореалистического качества, необходимо увеличивать скорости вычислений на этапах проективного преобразования и отображения поверхностей на дискретный растр. По результатам патентных исследований перспективные генераторы изображений должны иметь производительность до сотен тысяч многоугольников на кадр при частоте кадров 50 Гц или до нескольких миллионов многоугольников на кадр в секунду.

Таким образом, исследование новых алгоритмов преобразования и отображения поверхностей трехмерных объектов и разработка на их основе высокопроизводительных и недорогих геометрических машин и видеопроцессорных устройств представляются актуальной задачей.

Проективное преобразование триангулированных поверхностей. В [7] предложен новый алгоритм проективного преобразования триангулированных поверхностей трехмерных объектов при следующих ограничениях:

1) исходное описание поверхностей объектов составлено из треугольников, заданных трехмерными координатами вершин и набором параметров при этих вершинах (цвет, яркость, координаты текстурного рисунка и др.);

2) результирующее описание поверхностей объектов, спроецированных на плоскость изображения, задается пересечением полуплоскостей, ограниченных прямыми линиями, и содержит коэффициенты уравнений этих линий вида $Ax + By + C = 0$ и коэффициенты линейных функций, определяющие параметры в плоскости изображения в виде отношения $(Ax + By + C)/(Dx + Ey + F)$.

Характерной особенностью преобразования является отсутствие операции «перспективного» деления. Перенос этой операции в пространство изображений обеспечивает линейную интерполяцию параметров вдоль плоскости треугольников в объектном пространстве, что гарантирует достоверное отображение триангулированных поверхностей как при использовании метода излучательности [8], так и при текстурировании [9].

Преобразование полностью векторизовано и легко реализуется на процессорах векторно-конвейерного типа в RISC-суперкомпьютерной технологии параллельной обработки геометрических примитивов.

Задание треугольника в виде совокупности уравнений линий, проходящих через его ребра, и линейных функций для параметров располагает в пространстве изображений к параллельным вычислениям, потому что дает информацию для каждого отдельного пиксела в позиции x, y , т. е. описывает всю плоскость треугольника.

Pixel-Planes. Известен, по крайней мере, один класс видеопроцессорных устройств типа Pixel-Planes [10, 11], основанный на патентованном способе вычислений линейных выражений вида $f(x, y) = Ax + By + C$ с помощью двоичных деревьев одноразрядных сумматоров и одноразрядных арифметико-логических устройств (АЛУ) в каждой ячейке растровой видеобufferной памяти [12].

Для каждого направленного ребра выпуклого многоугольника, заданного уравнением $Ax + By + C = 0$, и для каждого пиксела плоскости изображений Pixel-Planes вычисляет функцию $f(x, y) = Ax + By + C$ и по знаку $f(x, y)$ определяет полуплоскость, в которой лежит многоугольник. Множество пикселов, покрытых многоугольником, находится как логическое пересечение полученных полуплоскостей.

Параметры вычисляются также для всей плоскости изображения, но присваиваются пикселям, найденным в предыдущей процедуре. Для текстурных параметров перспективное деление осуществляется с помощью АЛУ на пиксельном уровне.

Несмотря на оригинальные технические идеи и решения, устройства типа Pixel-Planes имеют один существенный недостаток — это большой объем неэффективно работающего оборудования.

Например, для генерации изображений с разрешением 1024×1024 пиксела устройство должно содержать более миллиона одноразрядных АЛУ, что составляет огромные вычислительные ресурсы и эквивалентно примерно 32000 32-разрядных АЛУ. Полезная работа Pixel-Planes пропорциональна площади многоугольника (или треугольника для триангулированных поверхностей). При большом количестве отображаемых треугольников (десятки и сотни тысяч на кадр) их средняя площадь составляет сотни или даже десятки пикселей. Поэтому при этих условиях коэффициент полезного действия оборудования Pixel-Planes можно оценить в сотые или даже тысячные доли процента.

Опыт Pixel-Planes показывает, что эффективный алгоритм отображения должен работать в пределах области многоугольника, а количество процессоров, параллельно обрабатывающих этот многоугольник, не может превышать его среднюю площадь.

На наш взгляд, высокоэффективная мультипроцессорная обработка геометрических примитивов, заданных линейными уравнениями и функциями, может быть обеспечена алгоритмами поиска пикселей путем рекурсивного деления плоскости изображений на подобласти, как в алгоритме Варнока для удаления невидимых поверхностей [13], с помощью матрицы процессоров значительно меньшей размерности, чем матрица пикселей в плоскости изображений [14].

Деление плоскости изображений. На начальном этапе в плоскости изображений устанавливается «окно», равное максимальному размеру изображения. Если область многоугольника не покрывает окно, что определяется специальным тестом покрытия, то многоугольник выбывает из дальнейшего рассмотрения. Этот этап эквивалентен клипированию по видимому объему. Если область покрывает начальное окно, переходят к следующему этапу.

Окно делят на четыре равные области или подокна, как показано на рис. 1. Для каждого из этих подокон проводят тест покрытия и оставляют для дальнейшего рассмотрения только те из них, в которых тест дал положительный результат. Для оставшихся подокон процедура подразделения и тестирования повторяется до тех пор, пока размер окна не достигнет пиксельного разрешения.

Таким образом, в отличие от Pixel-Planes алгоритм на каждом уровне подразделения отбрасывает области плоскости изображений, не имеющие отношения к делу.

Удобно делить плоскость изображения с разрешением $2^n \times 2^n$ пикселей. При этом глубина рекурсии алгоритма равна n , а на каждом i -м уровне подразделения легко определить: 1) два разряда координатной позиции покрытого пиксела: один разряд для координаты X и другой — для Y (рис. 1 иллюстрирует эту возможность); 2) линейную функцию параметров, соответствующую найденным разрядам координат, в виде

$$f_i(x_{n-i}, y_{n-i}) = f_{i-1} + 2^{n-i}Ax_{n-i} + 2^{n-i}By_{n-i} \quad (1)$$

где $i = 1, 2, \dots, n$ — уровень подразделения; $x_{n-i}, y_{n-i} \in (0, 1)$ — двоичный адрес окна; $f_0 = C$ — начальное значение функции.

Тест покрытия. Пусть в плоскости изображений заданы окно размером $a \times b$ с координатами центра окна (x_c, y_c) (см. рис. 1) и уравнение ребра многоугольника в виде $f(x, y) = Ax + By + C = 0$, причем $f(x, y) \geq 0$, если точка с координатами (x, y) находится в той же полуплоскости, что и отображаемый многоугольник.

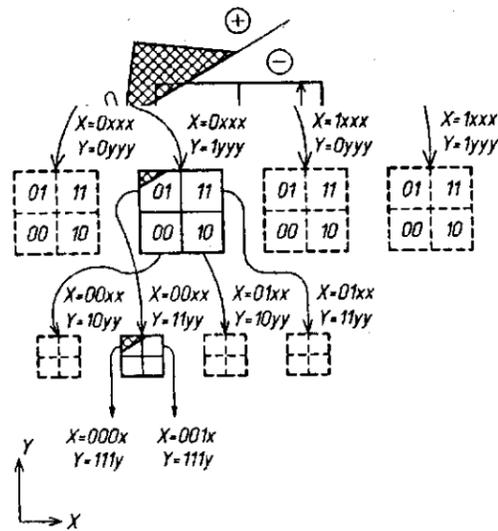


Рис. 1

Необходимым и достаточным условием отсутствия пересечения окна и положительной полуплоскости $f(x, y)$, а значит и отображаемого выпуклого многоугольника, является отрицательное значение $f(x, y)$ для всех угловых точек окна. Подставив координаты угловых точек в уравнение ребра, получим

$$f(x_c, y_c) \pm Aa/2 \pm Bb/2 = 0. \quad (2)$$

Для непокрытого окна $f(x_c, y_c)$ является заведомо отрицательной величиной. Поэтому равенство (2) может быть достигнуто чаще всего в одной угловой точке окна, для которой

$$\varepsilon = \max(\pm Aa/2 \pm Bb/2) = (|A|a + |B|b)/2.$$

Отсюда отсутствие покрытия устанавливается, если функция покрытия

$$d = f(x_c, y_c) + \varepsilon < 0, \quad (3)$$

а частичное или полное покрытие, если

$$d \geq 0. \quad (4)$$

Нетрудно показать, что полное покрытие окна положительной полуплоскостью $f(x, y)$ происходит при условии

$$d \geq 2\varepsilon. \quad (5)$$

Окно покрыто выпуклым многоугольником, если для всех его ребер выполняется условие (4). Если для ребра выполняется к тому же условие (5), то оно может не участвовать в процессе дальнейшего подразделения окна. Если ус-

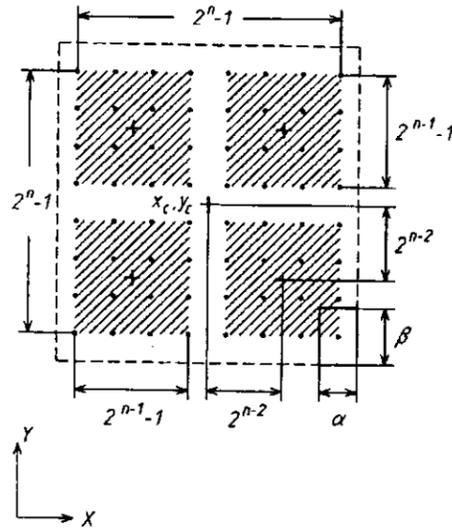


Рис. 2

ловие (5) выполняется для всех ребер, то область многоугольника покрывает окно полностью.

Вычисление функции покрытия на дискретном растре. Задачей антиэлайсингового отображения многоугольников на дискретный растр является поиск всех координатных позиций растра, таких что апертура фильтра, ассоциированная с каждой из них, оказывается накрытой областью многоугольника частично или полностью [16]. Апертуру фильтра можно представить окном размером $\alpha \times \beta$, центром которого является координатная позиция растра.

Пусть начальное окно содержит $2^n \times 2^n$ пикселей и имеет координаты центра (x_c, y_c) . При единичном расстоянии между пикселями размеры начального окна равны $a = 2^n - 1 + \alpha$, $b = 2^n - 1 + \beta$ (рис. 2). Согласно (3) для начального окна

$$d_0 = Ax_c + By_c + C + [|A|(2^n - 1 + \alpha) + |B|(2^n - 1 + \beta)]/2. \quad (6)$$

Разделив окно на четыре равных по количеству пикселей растра, получим подокна, содержащие по $2^{n-1} \times 2^{n-1}$ пикселей, размерами $(2^{n-1} - 1 + \alpha) \times (2^{n-1} - 1 + \beta)$ и координатами центров $(x_c \pm 2^{n-2}, y_c \pm 2^{n-2})$. С этими новыми параметрами значения функций покрытия согласно (3) станут следующими:

$$d_1 = A(x_c \pm 2^{n-2}) + B(y_c \pm 2^{n-2}) + C + [|A|(2^{n-1} - 1 + \alpha) + |B|(2^{n-1} - 1 + \beta)]/2,$$

или с учетом (6)

$$d_1 = d_0 - 2^{n-2}(|A| \mp A) - 2^{n-2}(|B| \mp B). \quad (7)$$

В полученном соотношении выражения в скобках могут принимать значения $2|A|$, $2|B|$ или нуль в зависимости от двоичного адреса подокон x_{n-1} , y_{n-1} и знаков $\text{sgn}A$, $\text{sgn}B$ коэффициентов A и B . Поэтому для первого уровня подразделения

$$d_1(x_{n-1}, y_{n-1}) = d_0 - 2^{n-2}|A|e_{1A} - 2^{n-2}|B|e_{1B},$$

где e_{1A} , $e_{1B} \in (0, 1)$ — логические функции x_{n-1} , y_{n-1} , $\text{sgn}A$, $\text{sgn}B$.

Для i -го уровня подразделения

$$d_i(x_{n-i}, y_{n-i}) = d_{i-1} - 2^{n-i}|A|e_{iA} - 2^{n-i}|B|e_{iB}. \quad (8)$$

Для системы координат и адресов подокон, принятых на рис. 1, 2, выражения для логических функций следующие:

$$\begin{aligned} e_{iA} &= x_{n-i} \cdot \text{sgn}A + \overline{x_{n-i}} \cdot \overline{\text{sgn}A}, \\ e_{iB} &= y_{n-i} \cdot \text{sgn}B + \overline{y_{n-i}} \cdot \overline{\text{sgn}B}. \end{aligned} \quad (9)$$

Таким образом, при делении плоскости изображений вычисление функций покрытия на дискретном растре согласно (8), (9) по степени сложности становится эквивалентным поразрядному вычислению линейных функций по формуле (1).

Анализ рекуррентного соотношения (8) показывает, что вычислительные затраты для определения функции покрытия сводятся в среднем к одному сложению на одно ребро на одно подокно. Тест на полное покрытие согласно (5) добавляет еще одно сложение на подокно. Вычисление линейной функции параметров также требует в среднем одного сложения на один параметр на подокно.

Поиск пикселей на разреженном растре. Представление о разреженном растре появляется при обработке плоскости изображений матрицей $p \times q$ процессоров [5, 14], причем каждый отдельный процессор обрабатывает не соседние пиксели, а каждый p -й пиксел каждой q -й строки (рис. 3).

Пусть разреженный растр содержит $2^n \times 2^n$ пикселей. Расстояние между пикселями в таком растре равно p единиц вдоль оси X и q единиц вдоль оси Y . Размеры начального окна для каждого процессора равны $a = p(2^n - 1) + \alpha$, $b = q(2^n - 1) + \beta$.

Поскольку полный дискретный растр образуется из разреженных растров, смещенных друг относительно друга на единичные межпиксельные расстояния, матрице процессоров соответствует матрица координат центров начальных окон $\{(x_c, y_c)_{j,k}\} = \{(x_0 + j, y_0 + k)\}$, где $0 \leq j \leq (p - 1)$ — столбец матрицы процессоров; $0 \leq k \leq (q - 1)$ — строка матрицы; (x_0, y_0) — координаты центра окна для процессора с номером $j, k = 0, 0$. В связи с этим и согласно (3) для каждого ребра многоугольника образуется матрица начальных значений функций покрытий

$$\{(d_0)_{j,k}\} = [f(x_0, y_0) + jA + kB + \varepsilon], \quad (10)$$

где $\varepsilon = [|A|(p2^n - p + \alpha) + |B|(q2^n - q + \beta)]/2$.

Нетрудно показать, что при делении плоскости изображений функция покрытия для j, k -го разреженного растра вычисляется в соответствии с рекуррентным соотношением

$$(d_i)_{j,k}(x_{n-i}, y_{n-i}) = (d_{i-1})_{j,k} - 2^{n-i}p|A|e_{iA} - 2^{n-i}q|B|e_{iB}, \quad (11)$$

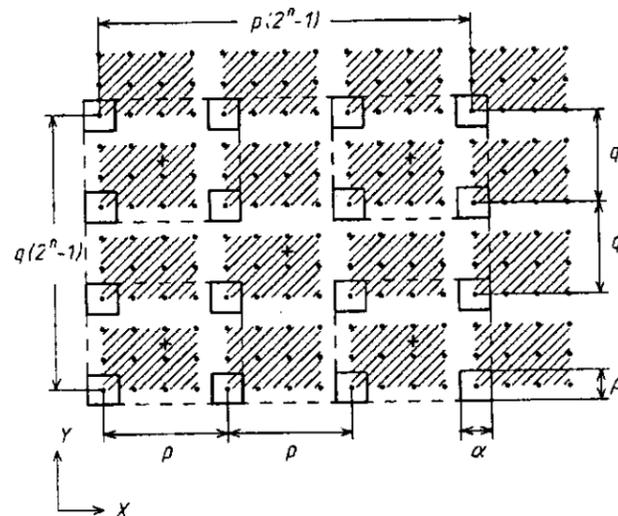


Рис. 3

где n — глубина рекурсии; e_{1A}, e_{1B} — логические функции согласно (9).

Аналогично для каждого параметра многоугольника, заданного линейной функцией, матрице процессоров соответствует матрица начальных значений функции параметра

$$[(f_0)_{j,k}] = [C + jA + kB], \quad (12)$$

C — начальное значение функции для $j, k = 0, 0$.

При делении плоскости изображений значение параметра для j, k -го разреженного растра может быть найдено в виде

$$(f_i)_{j,k}(x_{n-i}, y_{n-i}) = (f_{i-1})_{j,k} + 2^{n-i}pAx_{n-i} + 2^{n-i}qBy_{n-i}. \quad (13)$$

Таким образом, вычисление функций покрытия и параметров на разреженных растрах принципиально не отличается от обычного «монорастрового» варианта (см (6), (8) и (1)). Характерно лишь увеличение суммируемых компонент в p и q раз.

Можно отметить три очевидных достоинства применения матрицы процессоров и разреженных растров: 1) каждый отдельный процессор обрабатывает $(pq)^{-1}$ часть всего растра; 2) размеры апертуры фильтра на разреженном растре оказываются уменьшенными в pq раз; 3) процессоры матрицы связаны лишь процедурой формирования начальных значений функций покрытия и параметров. При ослаблении этой связи, например, с помощью очередей типа FIFO (First In — First Out) процессоры матрицы могут работать параллельно даже в том случае, когда отображаемые многоугольники накрывают области изображения, меньшие, чем $p \times q$ пикселей.

Удаление скрытых поверхностей и антиэлайсинг. Покажем возможности предложенного алгоритма для двух методов удаления невидимых поверхностей: метода Z -буфера для несортированных по глубине многоугольников и метода маскирования для многоугольников, поступающих на обработку в порядке прямого приоритета (первый — ближе к наблюдателю).

Z -буфер. При апертуре фильтра $\alpha = \beta = 0$ на последнем уровне подразделения подокна «стягиваются» в центры пикселей, что обеспечивает точечную выборку, характерную для метода Z -буфера. Удаление скрытых поверхностей производится на основе сравнения Z -координат многоугольников, вычисленных в качестве параметров для центров пикселей.

С помощью Z -буфера трудно осуществить эффективный антиэлайсинг. Однако все известные методы улучшения изображений, связанные с вычислениями на субпиксельном уровне, могут быть реализованы, во-первых, за счет увеличения времени обработки с использованием многопроходных алгоритмов [15], во-вторых, за счет увеличения количества процессоров в матрице в $s \times t$ раз, где s и t — соответственно число столбцов и строк субпиксельного растра.

Метод маскирования. Рассматриваемый алгоритм открывает уникальную возможность удаления скрытых многоугольников путем многоуровневого приоритетного маскирования с помощью масок «занятых» окон и субпиксельных масок. Последние можно использовать также для антиэлайсинга посредством дискретных фильтров типа "Bad of Nails" [16, 17].

Маски занятых окон могут быть сформированы на любом уровне подразделения. Для этого используется 4-разрядная память — по одному разряду на каждое подокно. Разряд маски устанавливается, если подокно полностью покрыто либо одним непрозрачным многоугольником, что определяется тестом на полное покрытие, либо совокупностью непрозрачных многоугольников, что передается по обратной связи в том случае, если заполнены четыре дочерних подокна. Подокна последнего уровня (пиксели) устанавливаются занятыми, если заполнены их субпиксельные маски.

Субпиксельная маска для каждого отдельного ребра, заданного нормализованным уравнением, формируется с помощью таблицы, индексированной расстоянием до ребра, равным $d_n - \epsilon_n$, знаками $\text{sgn}A, \text{sgn}B$ и величинами $|A|$

или $|V|$. Битовый образ многоугольника на субпиксельной маске образуется логическим произведением масок ребер. Заполненность субпиксельной маски определяется логической суммой масок всех многоугольников в пикселе.

На любом уровне подразделения многоугольник может быть удален маской занятых окон. С другой стороны, положительный тест на полное покрытие окна позволяет заполнить пиксели пустых дочерних подокон параметрами многоугольника обычным инкрементным способом. И только в том случае, если многоугольник накрывает пустое окно частично, процесс подразделения необходимо продолжать.

Таким образом, метод многоуровневого приоритетного маскирования обеспечивает эффективное удаление скрытых поверхностей на всех возможных фазах подразделения плоскости изображения, причем субпиксельный уровень достигается лишь на контурах видимых многоугольников. Многоуровневое маскирование уменьшает потери производительности (число многоугольников/ед. времени) при увеличении глубинной сложности изображений. Тем самым стабилизируется нагрузка на пиксельные процессоры, которые выполняют завершающие вычисления, фильтрацию и запоминание цвета пикселей в растровой видеобufferной памяти.

Заключение. Полностью векторизованное проективное преобразование триангулированных поверхностей, генерирующее описание двумерных треугольников в виде линейных уравнений и функций, и их антиэлайсинговое отображение на дискретный растр путем рекурсивного деления плоскости изображений являются хорошей основой для высокопроизводительного синтеза виртуальной среды.

Проективное преобразование легко реализуется на процессорах векторно-конвейерного типа. Рекурсивный поиск пикселей можно реализовать в общем случае с помощью четырех архитектур: 1) процессор со стеком окон; 2) конвейер процессоров с числом каскадов, равным глубине рекурсии; 3) четверичное дерево процессоров; 4) целесообразная комбинация вышеприведенных вариантов.

По характеру обработки на i -м уровне подразделения возможны параллельная обработка ребер и параметров, к чему предрасположены триангулированные поверхности, и последовательная обработка — для многоугольников с нефиксированным количеством ребер.

Если принять во внимание разреженные растры, то возможны матричные архитектуры, в которых элементом матрицы является одна из четырех вышеприведенных архитектур, т. е. матрицы процессоров со стеками, матрицы конвейеров и т. д.

Исследование этого ряда архитектур, оценка их эффективности и производительности, а также практические рекомендации по применению являются предметом последующей работы.

СПИСОК ЛИТЕРАТУРЫ

1. Кларк Д. Компьютерная визуализация в 80-х и 90-х // Компьютерная графика в науке и искусстве: Мат. конф.—М., 1991.
2. Ellis S. R. Nature and origins of virtual environments: a bibliographical essay // Comput. Syst. Eng.—1991.—2, N 4.—P. 321.
3. Ковалев А. М., Талицкий Э. А. Машинный синтез визуальной обстановки // Автометрия.—1984.—№ 4.
4. Computer Image Generation /Ed. B. J. Schachter.—N. Y.: Wiley, 1983.
5. Akeley K. The silicon graphics 4D/240GTX superworkstation // IEEE Computer Graphics & Applications.—1989.—9, N 1.—P. 71.
6. Гринберг Д. Синтез реалистических изображений: трассировка лучей и излучательность // Компьютерная графика в науке и искусстве: Мат. конф.—М., 1991.
7. Ковалев А. М., Тарнопольский Ю. В. Проективное преобразование триангулированных поверхностей // Автометрия.—1991.—№ 1.
8. Cohen M. F., Greenberg D. P. The hemi-cube, a radiosity solution for complex environments // Siggraph'85.—1985.—19, N 3.—P. 31.

9. Ковалев А. М., Тарасов Ю. В. Текстура на произвольно ориентированных поверхностях // Автометрия.—1988.—№ 6.
10. Fuchs H., Goldfeather J., Hultquist J. P. et al. Fast spheres, shadows, texture, transparencies and image enhancements in Pixel-Planes // Computer Graphics.—1985.—19, N 3.
11. Eyles J., Austin J., Fuchs H. et al. Pixel-Planes 4: A summary // Advances in Computer Graphics Hardware II.—Berlin: Springer-Verlag, 1988.
12. Pat. 4590465 US. Graphics display systems using logic-enhanced pixel memory cells /H. Fuchs.—Publ. 20.05.86.
13. Ньюмен У., Спрулл Р. Основы интерактивной машинной графики.—М.: Мир, 1976.
14. Clark J., Hannah M. Distributed processing in high performance smart image memory // VLSI Design.—1980.—1, N 3.
15. Mammen A. Transparency and antialiasing algorithms implemented with the virtual pixel maps technique // IEEE Computer Graphics & Applications.—1989.—9, N 1.—P. 43.
16. Ковалев А. М., Токарев А. С. Синтез изображений без злайсинга и оконтуривания // Автометрия.—1990.—№ 4.
17. Pat. 4570233 US. Modular digital image generator /J. K. Yan, J. K. Florence.—Publ. 11.02.86.

Поступила в редакцию 10 января 1993 г.
