

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

УДК 517.444 : 621.391.1 : 681.3.012

О. Е. БАКЛАНОВА, М. В. ЗЮЗИН, А. В. ЛЮЛЯКОВ
(Новосибирск)

О РЕАЛИЗАЦИИ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ЦИФРОВОЙ ФИЛЬТРАЦИИ НА МНОГОПРОЦЕССОРНОМ ВЫЧИСЛИТЕЛЬНОМ КОМПЛЕКСЕ ПС-2000

В задачах обработки экспериментальных данных большой популярностью пользуются методы цифровой фильтрации. Это связано, во-первых, с широким распространением быстродействующих вычислительных машин, функционирующих параллельно, и, во-вторых, с появлением новых, более эффективных алгоритмов реализации цифровых фильтров, в свою очередь легко поддающихся распараллеливанию.

В работе дается описание некоторых алгоритмов цифровой фильтрации сигналов и изображений, реализованных на многопроцессорном вычислительном комплексе (МВК) ПС-2000. Приводятся расчеты, демонстрирующие эффективность обработки экспериментальных данных.

Краткая характеристика МВК ПС-2000. Для использования преимуществ многопроцессорной системы необходимо учитывать возможности функционирования ее аппаратных средств. Остановимся на некоторых особенностях МВК ПС-2000 [1].

МВК ПС-2000 относится к машинам типа SIMD (один поток команд — много потоков данных) и включает в себя ЭВМ СМ-2, выполняющую управление системой, и матричный процессор ПС-2000, производящий основную обработку данных. Матричный процессор может содержать до 64 однородных процессорных элементов (ПЭ), действующих по команде, поступающей с устройства управления (УУ). Каждый процессорный элемент имеет собственную оперативную память (ОП) 16 К 24-разрядных слов и арифметико-логическое устройство (АЛУ), включающее в себя 16 регистров общего назначения, выполняющих быструю обработку данных. Обмен данных между соседними ПЭ осуществляется по регулярным каналам (РК), а передача данных между ПЭ и УУ — по магистральным каналам (МК). Упрощенная структурная схема МВК ПС-2000 приведена на рис. 1. Число процессорных элементов p , работающих в системе, может быть кратно 8, т. е. $p = 8, 16, \dots, 64$. Все ПЭ синхронно выполняют вычислительные операции над данными, хранящимися на регистрах общего назначения или слоях памяти процессорных элементов. Каждый регистр и слой памяти состоит из совокупности p 24-разрядных слов. Операции с векторами длины p производятся покомпонентно и одновременно:

$$(c_1, \dots, c_p) = (a_1, \dots, a_p) * (b_1, \dots, b_p), \quad c_i = a_i * b_i, \quad i = 1, \dots, p,$$

где $*$ — некоторая арифметическая или логическая операция. Для ускорения операций умножения могут использоваться различные сдвиги двоичных разрядов слов. При передаче данных между процессорными элементами применяется операция циклической перестановки, осуществляющая перемещение компонентов вектора вправо или влево. Например,

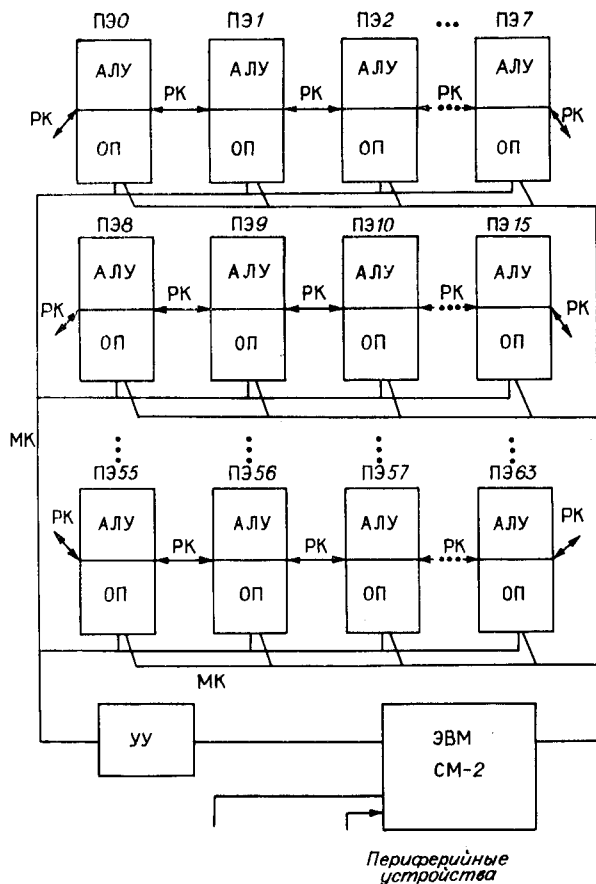


Рис. 1

рекомендаций: 1) вычислительный алгоритм представить в векторной форме, длина векторов должна быть пропорциональна числу ПЭ; 2) часто используемые данные хранить на регистрах; 3) чередовать вычислительные операции, обращения к памяти, ввод и вывод данных, так как эти действия могут выполняться параллельно; 4) по возможности исключить из алгоритма операции деления.

Обработка сигналов в режиме реального времени. Часто при обработке сигналов (звуковых, радиолокационных и т. д.) возникает необходимость реализации цифровых фильтров в реальном времени. Предположим, что отсчеты сигнала $f(n)$ непрерывно подаются с некоторой частотой ω , ..., $f(n-1)$, $f(n)$, $f(n+1)$, ... Необходимо подействовать на $f(n)$ фильтром

$$g(n) = \sum_{l=0}^N k(l) f(n-l), \quad n = 0, 1, \dots, \quad (1)$$

с заданной импульсной характеристикой $k(n)$ и получить последовательность отсчетов $g(n)$, выходящих с такой же частотой.

Таблица 1

Операция	Количество операций, с ⁻¹			Операция	Количество операций, с ⁻¹		
	1 ПЭ	16 ПЭ	64 ПЭ		1 ПЭ	16 ПЭ	64 ПЭ
+ _ф	10 ⁶	16 · 10 ⁶	64 · 10 ⁶	* _ф	0,4 · 10 ⁶	6,4 · 10 ⁶	25,6 · 10 ⁶
+ _п	0,8 · 10 ⁶	12,8 · 10 ⁶	51,2 · 10 ⁶	* _п	0,3 · 10 ⁶	4,8 · 10 ⁶	19,2 · 10 ⁶

в результате циклической перестановки вектора (a_1, \dots, a_p) на три слова вправо будет получен вектор $(a_{p-2}, a_{p-1}, a_p, a_1, a_2, \dots, a_{p-3})$. При выполнении некоторых операций можно инициировать деятельность только определенной части ПЭ. Остальные ПЭ будут простаивать. Наряду с векторной оперативной памятью процессорных элементов, обозначаемой через M , имеется память H , используемая для скалярных операций и хранения констант.

В табл. 1 даны некоторые характеристики производительности системы при выполнении основных арифметических операций. Здесь «+_ф» и «+_п» — операции сложения с фиксированной и плавающей точкой; «*_ф» и «*_п» — соответствующие операции умножения.

При работе на МВЦ ПС-2000 желательно придерживаться следующих

В дальнейшем будут рассмотрены некоторые алгоритмы обработки в реальном времени сигнала $f(n)$ фильтром (1), реализованные на ПС-2000.

Приведем выражение (1) к эквивалентному виду

$$g(n) = \sum_{l=0}^N k_1(l) f(n_1 + l), \quad n = 0, 1, \dots,$$

где $k_1(l) = k(N - l)$; $l = 0, \dots, N$; $n_1 = n - N$.

Предположим, что число коэффициентов импульсной характеристики $k_1(l)$ кратно числу p , $N + 1 = pc$, где $1 < c < 7$. Через R_i , $i = 1, \dots, 16$, обозначим регистры общего назначения; S_i , $i = 1, \dots, 16$ — слои оперативной памяти ПЭ. Будем использовать следующие операции:

- 1) сложение $R_i + R_j$ и умножение $R_i * R_j$;
- 2) запись значения R_i в R_j , $R_i \rightarrow R_j$;
- 3) сдвиг слов в регистре (циклическая перестановка);
- 4) запись слова памяти в регистр $S_i \rightarrow R_j$;
- 5) пересылка одного слова из R_i в R_j ;
- 6) ввод и вывод одного слова из регистра.

Обработку данных можно вести в формате как целых чисел (для этого вещественные коэффициенты $k_1(n)$ представляются в виде $m(n)/2^M$, где M и $m(n)$ — целые, $n = 0, \dots, N$), так и вещественных чисел. Целые числа впоследствии приводятся к вещественному виду.

Предположим, что каждый отсчет обрабатывается всеми процессорными элементами. Занесем коэффициенты $k_1(0), \dots, k_1(N)$ и отсчеты $f(-N), \dots, f(0)$ соответственно в регистры R_1, \dots, R_c и R_{c+1}, \dots, R_{2c} . Для вычисления значений $g(0), g(1), \dots$ используем

Алгоритм 1:

1. $n = 0$.
2. $0 \rightarrow R_{16}$, $i = 1$, $j = c + 1$, $l = 1$, $q = 1$.
3. R_i (коэффициенты) $* R_j$ (отсчеты) $+ R_{16} \rightarrow R_{16}$.
4. Если $i < c$, то слово 1 из R_{j+1} , иначе $f(n + 1) \rightarrow$ в слово 1 из R_j .
5. Сдвиг R_j на одно слово влево.
6. $i = i + 1$, $j = j + 1$, если $i \leq c$ на п. 3.
7. $R_{16} +$ (результат сдвига влево на l слов в $R_{16}) \rightarrow R_{16}$.
8. Если $q < \log_2 p$, то $l = 2l$, $q = q + 1$ на п. 7, иначе вывод $g(n)$ из слова 1 в R_{16} .
9. $n = n + 1$ на п. 2.

При реализации фильтра с большим количеством коэффициентов приходится использовать память процессорных элементов. Это приводит к массовым обменам между памятью и регистрами, что увеличивает время обработки.

Суть второго алгоритма заключается в обработке каждым процессорным элементом своего отсчета. Введем группы отсчетов

$$G(m) = \{g(n), \dots, g(n + p - 1)\} \text{ и } F(m) = \{f(n - N), \dots, f(n - N + p - 1)\}, \\ n = mp, m = 0, 1, \dots$$

Для вычисления значений группы $G(m)$ с помощью $F(m), \dots, F(m + c)$ применим следующий

Алгоритм 2:

1. $0 \rightarrow R_1$, $l = 0$, $F(m) \rightarrow R_3$.
2. $k_1(l) \rightarrow R_2$.
3. $R_1 + R_2 * R_3 \rightarrow R_1$.
4. Если $l = N$, то на п. 7.
5. $l = l + 1$, в слово 1 регистра R_3 занести $f(n - N + p + l)$.
6. Сдвиг R_3 на одно слово влево, на п. 2.
7. Вывод из R_1 группы $G(m)$.

Обработка $G(m + 1)$ требует повторного использования $F(m), \dots, F(m + c)$ и новой группы $F(m + c + 1)$. Хранение отсчетов следует организовать на свободных регистрах. В первые $N + 1$ слоев S_{i+1} , $l =$

Т а б л и ц а 2

Число обра- батываемых отсчетов	Число коэф- фициентов	Требуемое время, с	Затраченное время, с	
			Целый формат	Вещественный формат
768 000	32	16	12,35	12,50
1 024 000	32	21	16,36	16,67
768 000	48	16	15,40	—
1 152 000	48	24	23,10	—
1 536 000	48	32	30,80	—

$= 0, \dots, N$, оперативной памяти заносятся коэффициенты $k_1(l)$. Каждый слой S_{l+1} содержит p коэффициентов $k_1(l)$.

В дальнейшем будем предполагать, что сигнал поступает с частотой $\omega = 48$ кГц. Это соответствует принятому стандарту при создании устройств цифровой обработки звуковых сигналов студийного тракта. Для процессора ПС-2000 реализованы предложенные алгоритмы на языке ППС [4]. Численные эксперименты по обработке сигналов в реальном времени проводились на системе, содержащей 16 ПЭ. В результате установлено, что с помощью алгоритма 1 удастся реализовать фильтр, функционирующий в реальном времени, с числом коэффициентов, не превышающим 16. Для алгоритма 2 вычисления велись с целыми и вещественными числами. Результаты приведены в табл. 2. Как видно из таблицы, фильтр с 48 коэффициентами справляется с обработкой сигналов, поступающих с частотой 48 кГц на ПС-2000 с 16 ПЭ.

Реализация алгоритмов обработки изображений. Нерекурсивный фильтр, действующий на изображение $f(n_1, n_2)$, определяется преобразованием типа свертки:

$$g(n_1, n_2) = k * f(n_1, n_2) = \sum_{l_1, l_2=-\infty}^{\infty} k(l_1, l_2) f(n_1 + l_1, n_2 + l_2), \quad (2)$$

где $k(n_1, n_2)$ — импульсная характеристика фильтра. Через $K(\xi_1, \xi_2)$ обозначим частотную характеристику:

$$k * e^{i(\xi_1 n_1 + \xi_2 n_2)} = K(\xi_1, \xi_2) e^{i(\xi_1 n_1 + \xi_2 n_2)}.$$

Обычно требуется произвести обработку изображения $f(n_1, n_2)$ цифровым фильтром с некоторой импульсной характеристикой $k(n_1, n_2)$, отвечающей заданной функции $K(\xi_1, \xi_2)$.

При расчете широкого класса двумерных цифровых фильтров (полосовых, режекторных, обратных и т. д.) часто используется эффективная операция комбинационной сборки [2—4]. В этом случае импульсная характеристика представляется в виде

$$k_1(n_1, n_2) = \sum_{j=0}^N b_j k_0^{*j}(n_1, n_2). \quad (3)$$

Здесь b_j — коэффициенты преобразования; $k_0^{*j}(n_1, n_2)$ — свертка с собой j раз функции $k_0(n_1, n_2)$, имеющей небольшое количество коэффициентов (например, сосредоточенной в 3×3 точки сетки). Такое разложение функции $k_1(n_1, n_2)$ позволяет значительно сократить объем вычислительных операций при действии цифрового фильтра.

Опишем реализованные на МВК ПС-2000 алгоритмы обработки изображений цифровыми фильтрами с импульсными характеристиками $k(n_1, n_2)$ и $k_1(n_1, n_2)$. Предположим, что значения изображения $f(n_1, n_2)$, $n_1 = 1, \dots, M1$, $n_2 = 1, \dots, M2$, размещены в матрице $A(M1, M2)$, число строк и столбцов которой кратно числу процессорных элементов p , $M1 = K1p$, $M2 = K2p$. Пусть импульсная характеристика $k(n_1, n_2)$ захватывает $N1 \times N2$ узлов сетки, $n_1 = -\bar{n}_1, \dots, \bar{n}_1$, $n_2 = -\bar{n}_2, \dots, \bar{n}_2$, $N1 = 2\bar{n}_1 + 1$,

$N2 = 2\bar{n}_2 + 1$. Необходимо вычислить

$$g(n_1, n_2) = \sum_{l_1=-\bar{n}_1}^{\bar{n}_1} \sum_{l_2=-\bar{n}_2}^{\bar{n}_2} k(l_1, l_2) f(n_1 + l_1, n_2 + l_2) \quad (4)$$

в точках $n_1 = 1 + \bar{n}_1, \dots, M1 - \bar{n}_1, n_2 = 1 + \bar{n}_2, \dots, M2 - \bar{n}_2$.

Произведем запись матрицы A в память процессорных элементов M . В память первого процессора последовательно загрузим строки с номерами $1, \dots, K1 + N1 - 1$, второго — начиная от строки $K1 + 1$ до $2K1 + N1 - 1$ и т. д. Процессор с номером J будет содержать строки $(J - 1)K1 + 1, \dots, J K1 + N1 - 1, J = 1, \dots, p$.

Коэффициенты импульсной характеристики $k(n_1, n_2)$ последовательно запишем в скалярную память H в виде одномерного вектора B длины $M1 \times N2$:

$$B(i) = k(n_1, n_2), n_1 = -\bar{n}_1, \dots, \bar{n}_1, n_2 = -\bar{n}_2, \dots, \bar{n}_2, \\ i = (n_1 + \bar{n}_1)N2 + n_2 + \bar{n}_2 + 1.$$

Первые $N2$ слоев памяти умножим на элементы $1, \dots, N2$ вектора B и сложим на регистре $R1$. Слои с номерами $M2 + 1, \dots, M2 + N2$ умножим на последующие коэффициенты $B(N2 + 1), \dots, B(2 \cdot N2)$ и также суммируем с $R1$. Произведем соответствующие операции со слоями $2KM + 1, \dots, 2KM + M2, K = 2, \dots, M1 - 1$. В результате на регистре $R1$ будут получены значения изображения $g(n_1, n_2)$ с номерами $n_1 = (J - 1)K1 + \bar{n}_1 + 1, J = 1, \dots, p, n_2 = \bar{n}_2 + 1$.

В памяти процессорных элементов M выполним сдвиг на один слой вниз и повторим описанные операции. Тем самым будут вычислены элементы $g(n_1, n_2), n_1 = (J - 1)K1 + \bar{n}_1 + 1, J = 1, \dots, p, n_2 = \bar{n}_2 + 2$.

Повторяя в цикле $M1 \times M2/p$ раз указанную последовательность действий, получим остальные значения изображения $g(n_1, n_2)$ (см. (4)) и завершим обработку. Очевидно, приведенный алгоритм позволяет максимально загрузить процессорные элементы.

На основе операции типа свертки осуществляется распараллеливание реализации фильтра с импульсной характеристикой $k_1(n_1, n_2)$, рассчитанной операцией комбинационной сборки (3). Для этого в отдельной части оперативной памяти M необходимо многократно выполнить свертку изображения $f(n_1, n_2)$ с функцией $k_0(n_1, n_2)$. Первые $M1 \times M2/p$ слоев памяти используются для накопления результата.

Формально алгоритм работы фильтра (4) может быть записан в следующем виде:

1. $n = 1$.
2. $0 \rightarrow R1, l = 0, k = n - 1, i = 1$.
3. $j = 1$.
4. $R1 + S(k + j) * B(l + j) \rightarrow R1$.
5. $j = j + 1$, если $j \leq N2$ на п. 4.
6. $i = i + 1$, если $i > N1$ на п. 8.
7. $l = l + N2, k = k + M2$ на п. 3.
8. $R1 \rightarrow S(n), n = n + 1$ на п. 2.

Пример. На рис. 2 приведено изображение $g(n_1, n_2)$, содержащее 275×512 отсчетов, искаженное операцией типа свертки (2) с аппаратной функцией

$$k(n_1, n_2) = k_1 * k_1(n_1, n_2),$$

где $k_1(n_1, n_2)$ получена преобразованием Мак-Клелана [2, 4] одномерной функции-ступени

$$k_0(n) = \begin{cases} 1/5, & n = -2, \dots, 2; \\ 0 & \text{иначе.} \end{cases}$$

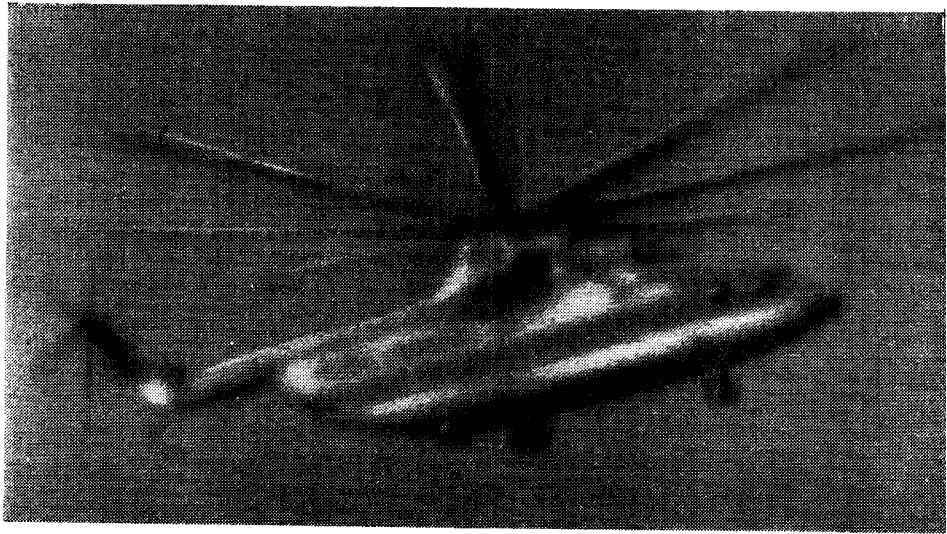


Рис. 2

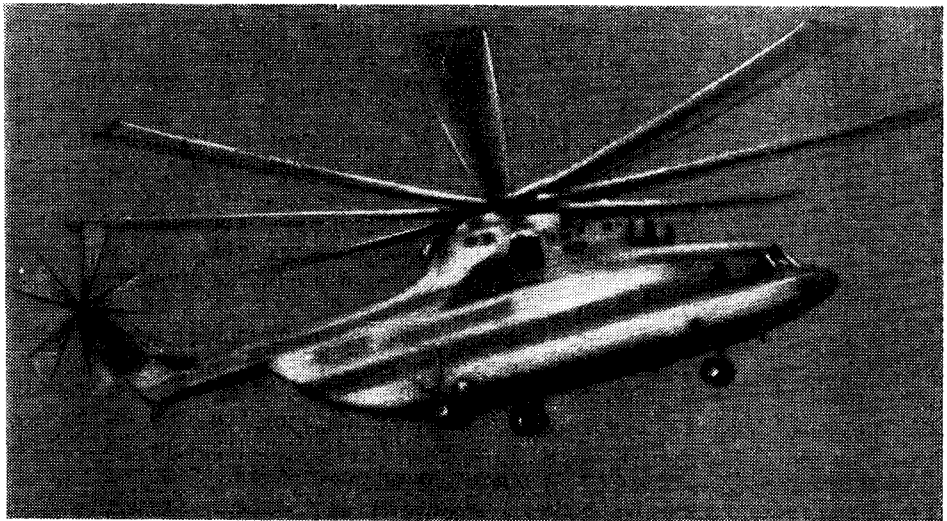


Рис. 3

Функция $k_1(n_1, n_2)$ представляется матрицей

$$\begin{pmatrix} 0,0125 & 0,05 & 0,075 & 0,05 & 0,0125 \\ 0,05 & 0,05 & 0 & 0,05 & 0,05 \\ 0,075 & 0 & 0,05 & 0 & 0,075 \\ 0,05 & 0,05 & 0 & 0,05 & 0,05 \\ 0,0125 & 0,05 & 0,075 & 0,05 & 0,0125 \end{pmatrix}.$$

Частотная характеристика $K_1(\xi_1, \xi_2)$, отвечающая $k_1(n_1, n_2)$, является знакопеременной, поэтому задача восстановления изображения $f(n_1, n_2)$ некорректна. Предварительно для усиления высоких частот использовался фильтр $\tilde{g}(n_1, n_2) = \hat{q} * g(n_1, n_2)$ с импульсной характеристикой

$$q(n_1, n_2) = \begin{pmatrix} -\frac{3}{2} & -3 & -\frac{3}{2} \\ -3 & 19 & -3 \\ -\frac{3}{2} & -3 & -\frac{3}{2} \end{pmatrix}.$$

Для расчета импульсной характеристики $\tilde{k}(n_1, n_2)$ обратного фильтра применялась операция комбинационной сборки [3].

$$\tilde{k}(n_1, n_2) = \sum_{j=0}^3 b_j (q * k(n_1, n_2))^{*j},$$

где $b_0 = 4$, $b_1 = -6$, $b_2 = 4$, $b_3 = -1$.

Восстановленное изображение $\tilde{f}(n_1, n_2) = \tilde{k} * \tilde{g}(n_1, n_2)$ приведено на рис. 3. Время обработки изображения на МК ПС-2000 с 16 процессорными элементами составило 5 с. Для сравнения: аналогичная обработка на ЭВМ ЕС 1061 заняла 39,5 с.

ЛИТЕРАТУРА

1. Школьник К. М. Технологическая поддержка ПС-2000 в системе ОБРАЗ // Теоретические и прикладные вопросы параллельной обработки информации.— Новосибирск: ВЦ СО АН СССР, 1984.
2. McClellan J. H. The design of two-dimensional digital filters by transformations // Proc. 7th Annual Princeton Conference on Information Sciences and Systems.— USA, 1973.— P. 247.
3. Зюзин М. В. Алгоритм сборки обратных фильтров // Автометрия.— 1987.— № 4.
4. Зюзин М. В. Экономичный способ сборки двумерных цифровых фильтров в задаче обработки изображений // Вариационно-разностные методы в задачах численного анализа.— Новосибирск: ВЦ СО АН СССР, 1987.

Поступила в редакцию 16 июля 1987 г.

УДК 681.3.06

А. М. КОВАЛЕВ, Ю. В. ТАРАСОВ

(Новосибирск)

ТЕКСТУРА НА ПРОИЗВОЛЬНО ОРИЕНТИРОВАННЫХ ПЛОСКИХ ПОВЕРХНОСТЯХ

Введение. Термин «текстура» определяется в машинной графике как многомерное изображение, которое переносится (отображается) на многомерное пространство [1]. В данной статье это понятие имеет более узкий смысл — двумерный массив изображения (текстурная карта), отображаемый на трехмерную плоскость. Перенос текстуры на поверхность является одним из средств повышения реализма синтезируемой сцены в системах машинной графики. Текстура значительно повышает визуальную сложность изображения при относительно малом увеличении количества вычислений. С помощью текстуры можно моделировать цвет поверхности, зеркальное отражение, прозрачность и другие эффекты. Перед авторами статьи стояла задача разработать способ отображения текстуры, который был бы простым в аппаратной реализации.

При отображении текстуры приходится решать две проблемы: геометрического отображения массива текстуры на поверхность и фильтрации изображения, которая устраняет на рисунке ступенчатые линии, муары и т. д. — все, что в литературе по машинной графике обозначается термином «элайсинг».

Способ отображения образца текстуры на поверхность путем прямой свертки рассмотрен в [2] (рис. 1). Образец текстуры представляется массивом чисел (например, интенсивностей). Четыре угла пиксела (элемента изображения, в данном случае прямоугольной формы), заданные в системе координат экрана с осями X_s , Y_s , проецируются на текстурную карту. Элементы текстурной карты (текселы), попавшие в проекцию пиксела, суммируются с некоторым весом, и полученное значение берется как интенсивность пиксела. В системах реального времени про-