

УДК 681.324 : 519.68

О. Г. Монахов, Э. А. Монахова
(Новосибирск)

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ НА ПАРАЛЛЕЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ

Рассмотрены алгоритмы предварительной обработки изображений. Предложен подход к распараллеливанию алгоритмов обработки изображений и рассмотрена реализация полученных параллельных алгоритмов на вычислительной системе (ВС) с программируемой структурой МИКРОС. Приведены оценки эффективности исполнения параллельных алгоритмов обработки изображений на ВС МИКРОС.

1. Вычислительная система МИКРОС: структура и архитектура. Микромашинальная система МИКРОС — экспериментальная реализация концепции вычислительных систем с программируемой структурой [1—3] в условиях отечественной элементной базы.

В основу системы МИКРОС положены следующие архитектурные принципы: параллелизм (процессов управления, доступа к данным и их обработки); децентрализация (распределенность средств управления и данных); программируемость структуры; модульность и конструктивная однородность; асинхронность взаимодействия модулей и процессов; наращиваемость, статическая и динамическая реконфигурация системы. Системы с программируемой структурой представляют собой совокупность вычислительных модулей — элементарных машин (ЭМ), объединенных программно-настраиваемой сетью связи (рис. 1). Каждая ЭМ включает в себя универсальную ЭВМ (в качестве которых были выбраны отечественные микроЭВМ семейства «Электроника» со всем спектром внешних устройств и сопроцессоров) и системное устройство (СУ), осуществляющее функции программно-управляемого коммутатора сети связи ВС.

Модуль СУ выполнен в трех вариантах, различающихся числом связей с соседними ЭМ (от двух до четырех дуплексных связей) и набором аппаратурно-реализуемых функций информационного канала. Модуль СУ занимает плату конструктива микроЭВМ «Электроника-60М». В одной ЭМ может быть от одного до трех модулей СУ (в зависимости от варианта его реализации).

Конструкция СУ обеспечивает для ВС МИКРОС формирование как сосредоточенных, так и распределенных звеньев микроструктуры ВС. Рекомендуе-

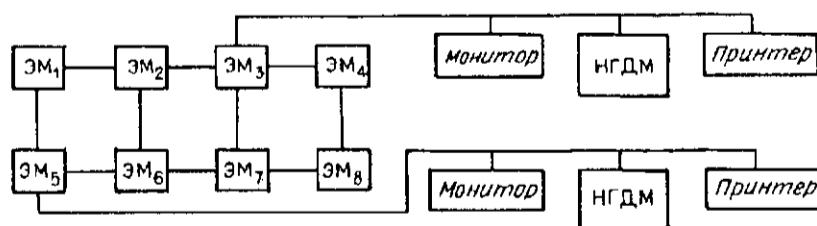


Рис. 1

мые структуры сети межмашинных связей — D_n , $R_s(N, v, g)$ - и $\bar{L}(N, v, g)$ -структуры, где N — число вершин, v — степень вершины (число межмашинных связей для каждой ЭМ), g — обхват (длина кратчайшего простого цикла), $n = v/2$ [4—6]. Например, широко используемые структуры типа гиперкуба — частный случай $L(2^n, v, 4)$ -структур. Разработаны алгоритмы синтеза оптимальных макроструктур перечисленных классов, которые обеспечивают минимум задержек при передаче информации и максимум показателей структурной живучести и коммутационности [4—6].

Создана децентрализованная распределенная операционная система (ОС), резидентная часть которой включает уровни драйвера СУ, ядра ОС, интерфейсов с ОС РАФОС и терминалом. Разработаны средства управления конфигурацией ВС, позволяющие задавать структуру подсистемы адекватно структуре решаемой задачи [7]. Реализован уровень распределенного управления заданиями в ВС, включающий децентрализованную файловую систему ВС и интерпретатор языка управления заданиями, представляющий расширение команд монитора РАФОС [8]. Для программирования параллельных алгоритмов созданы языки параллельного программирования: Паскаль-С [9], Фортран-С и параллельный Макроассемблер. Построение данных параллельных языков основано на введении в язык программирования штатной ОС РАФОС операций над семафорами и процессами с помощью примитивов ядра базовой ОС МИКРОС.

Средства ОС позволяют использовать ВС МИКРОС в режимах: 1) параллельной обработки информации, 2) сети ЭВМ, 3) распределенной мультимикропроцессорной системы.

2. Алгоритмы предобработки изображений. Исследуемые алгоритмы предобработки изображений включают в себя алгоритмы сглаживания и оконтуривания. Данные алгоритмы основаны на локальной (внутрикадровой) обработке, при которой фрагмент изображения (кадр 32×32 точки) сканируется окном 3×3 . При этом на внутренние элементы окна воздействует оператор, определенный алгоритмом обработки.

Для описания алгоритмов обработки изображения введем следующие обозначения: u_{ij} — значение видеосигнала в точке (i, j) , где $i = 1, 2, \dots, 32$; $j = 1, 2, \dots, 32$; $0 \leq u_{ij} \leq 255$; U_{ij} — окно:

$$U_{ij} = \begin{vmatrix} u_{i-1, j-1} & u_{i, j-1} & u_{i+1, j-1} \\ u_{i-1, j} & u_{i, j} & u_{i+1, j} \\ u_{i-1, j+1} & u_{i, j+1} & u_{i+1, j+1} \end{vmatrix} \quad \text{или} \quad U_{ij} = \begin{vmatrix} u_1 & u_2 & u_3 \\ u_8 & u_0 & u_4 \\ u_7 & u_6 & u_5 \end{vmatrix};$$

$$A = \begin{vmatrix} a_1 & a_2 & a_3 \\ a_8 & a_0 & a_4 \\ a_7 & a_6 & a_5 \end{vmatrix} \text{ — оператор, воздействующий на окно; } AU_{ij} = \sum_{k=0}^8 a_k u_k.$$

2.1. Алгоритмы сглаживания. В данном разделе рассматриваются процессы сглаживания, которые включают в себя простые локальные операции, выполненные на зашумленном изображении. Если шум состоит из изолированных точек, а идеальное изображение таких точек не содержит, эти точки можно заменить средними значениями, вычисленными по соседним элементам. Таким образом, за счет локального усреднения каждого элемента изображения по соседям можно уменьшить шум внутри различных областей изображения [10]. К сожалению, такая операция также размывает границы областей, для устранения этого эффекта применяются рассмотренные ниже методы оконтуривания.

Первый алгоритм — «исключение выколотых точек». Сглаженное значение видеосигнала в точке (i, j) равно:

$$\bar{u}_{ij} = \begin{cases} u_{ij}, & \text{если } \left(\min_k u_k - m \right) \leq u_{ij} < \left(\max_k u_k + m \right); \\ M & \text{в противном случае,} \end{cases}$$

где $M = 1/8 \sum_{k=1}^8 u_k$, m — задаваемый допуск для значения видеосигнала.

Второй метод — «сравнение с матожиданием» — аналогичен первому, но

Третий метод («отсечение»):

$$u_{ij} = \begin{cases} u_{ij}, & \text{если } \min_k u_k \leq u_{ij} \leq \max_k u_k; \\ u_k, & \text{если } u_{ij} > \max_k u_k; \\ u_k, & \text{если } u_{ij} < \min_k u_k, \quad k = 1, 2, \dots, 8. \end{cases}$$

2.2. *Оконтуривание.* Методы оконтуривания используются для выделения границ (контуров) областей изображений и основаны на методе пространственного дифференцирования, т. е. на предположении о том, что граничные точки (контуры) имеют большую величину модуля градиента функции яркости анализируемого изображения. Получаемое после преобразования изображение называют градиентным изображением с усиленными границами. Вычислительная реализация данных методов сводится к синтезу численных алгоритмов оценки частных производных в некоторой точке изображения. Известные в настоящее время многочисленные алгоритмы пространственного дифференцирования, называемые, как правило, градиентными или дифференциальными операторами, отличаются друг от друга выбранным численным способом оценки частных производных функции яркости изображения по ортогональным направлениям [11].

2.2.1. Первый метод:

$$A_1 = \begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix}.$$

2.2.2. Второй метод:

$$A_2 = \begin{vmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{vmatrix}.$$

2.2.3. Третий метод:

$$A_3 = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}.$$

2.2.4. Четвертый метод:

$$A_4 \bar{U}_{ij} = |A_{41} \bar{U}_{ij}| + |A_{42} \bar{U}_{ij}|,$$

$$A_{41} = \begin{vmatrix} 0 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{vmatrix}, \quad A_{42} = \begin{vmatrix} 0 & 1 & 1 \\ 0 & -1 & -1 \\ 0 & 0 & 0 \end{vmatrix}.$$

2.2.5. Пятый метод:

$$A_5 \bar{U}_{ij} = \sum_{k=1}^8 |U_0 - U_k|.$$

2.2.6. Шестой метод:

$$A_6 \bar{U}_{ij} = |A_{61} \bar{U}_{ij}| + |A_{62} \bar{U}_{ij}|,$$

$$A_{61} = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix}, \quad A_{62} = \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix}.$$

3. Подход к распараллеливанию алгоритмов обработки изображений. Рассматриваемый подход к распараллеливанию алгоритмов обработки изображений основан на методике крупноблочного распараллеливания [1, 2]. При этом кадр, поступивший в одну из элементарных машин системы (корневая ЭМ) с видеокамеры, видеобуфера или с внешних устройств (ВнУ) (дисконд, магнитофон), разбивается на равные части (полосы) по числу машин подсистемы и рассылается во все ее машины.

Каждая машина, получив свою часть кадра, выполняет над ней заданный алгоритм обработки, взаимодействуя в случае необходимости с другими ЭМ подсистемы. По окончании обработки своей полосы все ЭМ подсистемы посылают (в случае необходимости) обработанные полосы в корневую ЭМ, где происходит сборка обработанного кадра.

Для минимизации межмашинных обменов в ходе исполнения алгоритма обработки предложено использовать разбиение кадра на горизонтальные полосы с перекрытием, что позволяет также избежать искажений при обработке крайних строк полосы.

Таким образом, обработку кадра на ВС можно разбить на следующие этапы:

1. Настройка подсистемы с заданным числом ЭМ и загрузка ее ветвями параллельной программы.
2. Ввод кадра в корневую машину подсистемы — ЭМ₀.
3. Рассылка частей кадра из ЭМ₀ во все машины подсистемы.
4. Параллельное исполнение алгоритма всеми машинами подсистемы.
5. Сбор сигнала о конце работы алгоритма со всех машин в ЭМ₀.
6. Исполнение одного из следующих этапов обработки:
 - а) передача команды из ЭМ₀ во все ЭМ об исполнении следующего алгоритма обработки над данным кадром;
 - б) сбор всего кадра в ЭМ₀ и вывод его на ВнУ;
 - в) ввод нового кадра с ВнУ.

Рассмотрим реализацию перечисленных этапов при выполнении параллельных алгоритмов обработки изображений (на примере алгоритмов сглаживания и оконтуривания).

4. Реализация параллельных алгоритмов обработки изображений на ВС МИКРОС. Для реализации параллельной программы в ВС МИКРОС (см. рис. 1) необходимо выделить подсистему с заданным числом машин и заданной конфигурации [7]. В данном случае рассматривается реализация параллельных алгоритмов обработки изображений на подсистеме типа «линейка», которая представляет собой цепочку из последовательно соединенных машин (на рис. 2 показана подсистема «линейка» из четырех ЭМ).

Параллельная программа написана таким образом, что она адаптивно (автоматически) настраивается на число машин, выделенных в подсистему. При этом каждая ветвь параллельной программы автоматически настраивается на номер машины, в которую данная ветвь загружена, и соответственно на номер полосы в кадре, которую необходимо обработать в данной ЭМ. Сам текст параллельного алгоритма отличается от последовательного только включением операторов приема и передачи обрабатываемого кадра и операторов настройки параметров цикла обработки на соответствующую данной ЭМ полосу

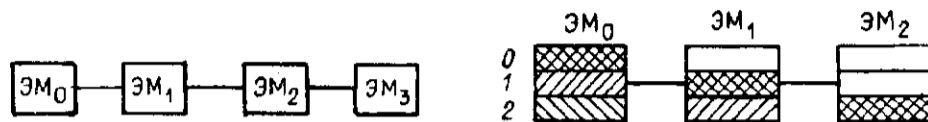


Рис. 2

Рис. 3

в кадре. Таким образом, параллельный алгоритм состоит из одинаковых ветвей, параметры которых настраиваются на соответствующую полосу на этапе загрузки. Исключение составляет ветвь, загружаемая в корневую машину — ЭМ₀. Данная ветвь включает в себя также и алгоритмы работы с ВнУ (дискководом и терминалом) для ввода/вывода кадров. Язык реализации всех подпрограмм алгоритмов обработки изображений — Фортран-С (язык Фортран штатной ОС РАФОС, расширенный средствами ядра системы МИКРОС, обеспечивающими порождение, уничтожение и взаимодействие асинхронных вычислительных процессов).

Ввод в ЭМ₀ кадра, записанного в файле на магнитном диске, осуществляется программой MAGD. Данная программа также обеспечивает запись обработанного кадра в заданный файл на диске.

Настройка параметров программ обработки на конкретную полосу в кадре в зависимости от номера обрабатываемой машины осуществляется в головных программах — ОР1М (для корневой ЭМ₀) и ОР1М (для всех остальных ЭМ). В корневой ЭМ головная программа поддерживает также диалог с пользователем через терминал.

Распределение полос введенного кадра по машинам подсистемы осуществляется с помощью программы TREX (L), где L — адрес массива, содержащего кадр. Результат работы алгоритма по распределению полос кадра из ЭМ₀ по машинам системы для трех ЭМ показан на рис. 3 (двойной штриховкой выделены полосы, обрабатываемые данной ЭМ, одинарной — пересылаемые в соседние ЭМ).

Сбор кадра в корневую ЭМ из всех ЭМ подсистемы осуществляется программой CLEX (L).

Передача сигналов управления (например, номера исполняемого алгоритма) из ЭМ₀ во все ЭМ подсистемы реализуется с помощью программы TREX1 (L, LEN), где L — адрес массива, LEN — его длина. Сбор сигналов готовности (об окончании исполнения заданного алгоритма) со всех ЭМ подсистемы в ЭМ₀ реализуется программой CLEX1 (L, LEN).

Программа CLEM (AMIN, AMAX) используется для определения максимального и минимального элементов в кадре. При входе в программу параметры должны иметь значение минимального и максимального элементов в полосе, находящейся в данной машине.

Вывод на экран дисплея и на печать заданного кадра осуществляется программой F.

Программа SGL1 реализует алгоритм сглаживания 1 («исключение выколотых точек»).

Программа SGL2 реализует алгоритм сглаживания 2 («сравнение с матожиданием»).

Программа SGL3 реализует алгоритм сглаживания 3 («отсечение»).

Программы ОК1, ОК2, ..., ОК6 реализуют алгоритмы оконтуривания соответственно с первого по шестой.

5. Оценка эффективности исполнения параллельных алгоритмов обработки изображений на ВС МИКРОС. Рассмотренные параллельные алгоритмы обработки изображений были реализованы на макете ВС МИКРОС, состоящем из ЭВМ «Электроника-60М», объединенных с помощью системных устройств СУ-1.

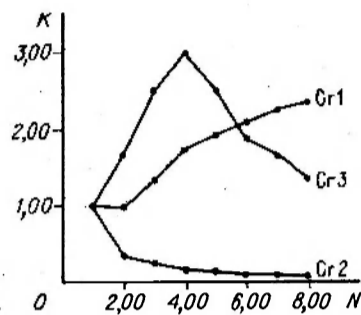


Рис. 4

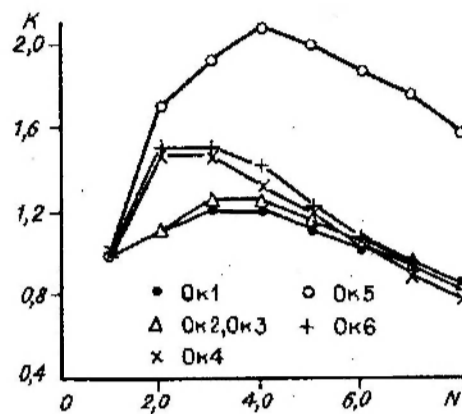


Рис. 5

Число машин в подсистеме, на которой выполнялся алгоритм, изменялось от 1 до 8. В эксперименте обрабатывались фрагменты изображений, записанных на диске, размером 32×32 с использованием ядра размером 3×3 . В ходе эксперимента определялось время исполнения алгоритмов с помощью таймера ЭВМ.

Для оценки эффективности параллельных алгоритмов используем коэффициент ускорения $k = T_1/T_N$, где T_1 — время исполнения алгоритма на одной ЭМ, T_N — время исполнения алгоритма на системе из N ЭМ. Полученные результаты приведены на рис. 4 (для алгоритмов сглаживания (Gr)) и на рис. 5 (для алгоритмов оконтуривания (Ok)).

Представленные результаты свидетельствуют об эффективности применения параллельной обработки изображений (получено ускорение в 2—3 раза). Спад коэффициента ускорения при изменении числа машин с 4 до 8 объясняется возрастающим влиянием времени передачи информации между ЭМ.

Применение нового системного устройства СУ-2, имеющего одноплатную компоновку для всех четырех направлений и позволяющего осуществлять обмены на фоне вычислений, позволит увеличить эффективность исполнения алгоритмов обработки изображений.

СПИСОК ЛИТЕРАТУРЫ

1. Миренков Н. Н. Параллельное программирование для многомодульных вычислительных систем. — М.: Радио и связь, 1989.
2. Корнеев В. В. Архитектура вычислительных систем с программируемой структурой. — Новосибирск: Наука, 1985.
3. Хорошевский В. Г. Вычислительная система МИКРОС. — Новосибирск, 1983. — (Препр. / СО АН СССР. Ин-т математики; 38).
4. Монахова Э. А. Оптимальные КАИС-структуры однородных вычислительных систем // Электронное моделирование. — 1985. — № 3.
5. Корнеев В. В., Монахов О. Г. Графы межмашинных связей однородных вычислительных систем // Изв. АН СССР. Техн. кибернетика. — 1980. — № 2.
6. Монахов О. Г. Параметрическое описание графов межмашинных связей однородных вычислительных систем // Вычислительные системы. — Новосибирск: ИМ СО АН СССР, 1979. — Вып. 80.
7. Монахов О. Г., Монахова Э. А. Организация межмашинных взаимодействий в вычислительной системе МИКРОС // Там же. — 1984. — Вып. 105.
8. Монахова Э. А., Корнеев В. В., Монахов О. Г. Организация распределенного управления заданиями в вычислительной системе МИКРОС // Управляющие системы и машины. — 1987. — № 5.

9. Корнеев В. В., Монахов О. Г. Параллельный Паскаль для вычислительных систем с программируемой структурой // Микропроцессорные средства и системы.—1990.—№ 1.
10. Розенфелд А. Распознавание изображений // ТИИЭР.—1981.—69, № 5.
11. Денисов Д. А., Низовкин В. А. Сегментация изображений на ЭВМ // Зарубеж. радиоэлектрон.—1985.—№ 10.

Поступила в редакцию 17 ноября 1992 г.

Реклама продукции в нашем журнале — залог Вашего успеха!