

Пороговое расстояние	$P_r=53$					$P_r=75$										
	1	2	3	4	5	1	2	3	4	5	6	8	9	11	13	48
Число точек в группе																
Число групп	133	44	5	5	1	57	15	8	4	5	2	2	1	1	1	1

Для точечного процесса Пуассона можно указать среднее число групп всех размеров на единицу площади [3]:

$$C \simeq (\pi P_r^2 \lambda^2) / \exp(\pi P_r^2 \lambda - 1),$$

где  $\lambda$  — плотность точек.

Выходными параметрами программы для различных пороговых расстояний являются следующие характеристики групп: число элементов, координаты центров, средний радиус и среднее квадратичное отклонение радиуса для каждой группы, среднее расстояние между точками группы и среднее квадратичное отклонение этого расстояния.

Программа позволяет проводить анализ точечных случайных полей на групповое размещение, что имеет важное значение для оценки состояния и прогнозирования развития леса.

Таким образом, разработано математическое обеспечение первичного статистического анализа аэрофотоснимков леса, применимое и к другим изображениям точечных объектов.

Результаты работы некоторых программ иллюстрируются рис. 3—6, таблицей, в которой представлены распределения групп для различных пороговых расстояний  $P_r$ . Начальное значение может быть получено программой LESRRN или оценено как среднее значение расстояния до ближайшего соседа для случайного размещения.

#### ЛИТЕРАТУРА

1. Нестерихин Ю. Е. и др. Автоматизированный комплекс обработки изображений.— Автометрия, 1980, № 4.
2. Кендалл М., Моран П. Геометрические вероятности.— М.: Наука, 1972.
3. Getis A., Boots B. Models of Spatial Processes.— L.: Cambridge, 1978.
4. Химмельблау Д. Анализ процессов статистическими методами.— М.: Мир, 1973.
5. Ripley B. D. Modelling Spatial Patterns.— J. of the Royal Statist. Soc., 1977, vol. B39, N 2.

Поступила в редакцию 18 января 1982 г.

УДК 681.3.06

Н. С. ЯКОВЕНКО

(Новосибирск)

### ОРГАНИЗАЦИЯ СИСТЕМНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЛЕКСА ОБРАБОТКИ АЭРОКОСМИЧЕСКОЙ ИНФОРМАЦИИ

**Введение.** Данная статья является развитием работы [1]; в ее цели входит описание организации программного обеспечения автоматизированного комплекса обработки аэрокосмической информации [2], которое реализовано на ЭВМ ЕС-1010 — управляющей ЭВМ системы.

Математическое обеспечение комплекса можно подразделить на три уровня. Базовые программные модули нижнего уровня (системное про-

граммное обеспечение), написанные, как правило, на языке Ассемблер или на языке высокого уровня PLR-10, занимаются обработкой операций ввода/вывода либо простейшими (с минимальными вычислениями) операциями обработки информации, поступающей от сканирующего устройства фотометрического автомата «Зенит-2» [3]. Программные модули среднего уровня (специализированное программное обеспечение), обычно написанные на языке ФОРТРАН и осуществляющие предварительную обработку поступающих данных, оформлены в виде законченных функциональных операций, использующих модули нижнего уровня. На основе программных модулей нижнего и среднего уровней написаны или могут быть созданы прикладные программы, работающие, как правило, в диалоговом режиме или в диалоговом режиме с инициативой программы, которые реализуют конкретный алгоритм обработки информации, зарегистрированной на фотоносителях (в частности, астро- и аэроснимках). В состав прикладных программ входят также диагностические мониторные системы, предназначенные для настройки и отладки различных КАМАК-блоков управления, других функциональных частей фотометрического автомата и его специализированных внешних устройств.

**Хранение, вызов и сборка модулей.** Текст каждой программы хранится в символьном файле системы управления файлами FMS-10 [4]; имя текстового модуля совпадает с именем первой секции (если это ассемблерный модуль или модуль, написанный на PLR-10) либо с именем первой подпрограммы (если это фортрановский модуль). Символьные файлы обрабатываются при помощи системы подготовки и редактирования текстовой информации [5]. После трансляции текстового модуля, содержащего, вообще говоря, несколько секций или подпрограмм, получается объектный модуль формата RB [6], который будет иметь имя, совпадающее с именем первой секции или подпрограммы, и может быть помещен в одну из библиотек дисковой операционной системы (SL — для специализированных модулей среднего уровня, UL — для системных модулей нижнего уровня). В дальнейшем при помощи системной программы ВІВ имя объектного модуля может быть изменено, но не могут быть изменены имена секций или подпрограмм. Вызов одиночных секций не представляет затруднений при сборке редактором связей LINKD исполняемых программ формата RMI. Если же объектный модуль содержит несколько секций и из основной программы вызываются только секции, отличные от первой, то редактор связей не сможет найти их в библиотеках. Чтобы избежать таких ситуаций, в описываемых модулях первая секция служит для подгрузки всего модуля и содержит либо операции «фиксирования переменных» (см. ниже) и инициализации переменных и аппаратуры, либо просто возврат в вызывающий модуль. Вызов иницирующей секции достаточно производить один раз (в начале программы). Если исполняемая программа имеет оверлейную структуру, то вызов подобных секций можно помещать в ствол программы (однократный вызов) или в начало одной из параллельных ветвей (автоматический многократный вызов при очередной загрузке ветви в оперативную память).

Исполняемая программа может работать только в одном из двух режимов: в нормальном (SM — SLAVE MODE) либо в привилегированном (MM — MASTER MODE). К сожалению, транслятор с ФОРТРАНа не позволяет исполняемой программе работать в MM-режиме, если она имеет подпрограмму с параметрами, написанную на ФОРТРАНе. Все модули нижнего уровня содержат команды ввода/вывода и, следовательно, должны работать в MM-режиме. Таким образом, программы, использующие такие модули, должны работать в MM-режиме и не могут иметь ФОРТРАН-подпрограмм с параметрами. «Передать параметры» в MM-режиме в фортрановский модуль можно только через блок общих данных (COMMON) с вызовом без параметров. Вызов ассемблерного модуля из ФОРТРАН-программы возможен и используется обычным образом, причем, если секция имитирует подпрограмму-функцию, т. е. передает значение на регистрах и не требует аргумента, тем не менее фактический

параметр должен указываться:  $A = \text{FUNCT}(\text{DUMMY})$ . Для ускорения процесса передачи параметров применяется способ фиксации переменных. Для этого в вызове иницилирующей секции модуля с несколькими секциями указываются переменные для входных и выходных данных, с которыми будут работать остальные секции, и вызов этих секций производится без параметров (входные данные присваиваются до вызова, а результат — перед возвратом), как при работе с блоком общих данных, только переменные здесь будут находиться в локальном сегменте данных. Ускорение процесса передачи параметров происходит за счет того, что при работе с COMMON-переменными ФОРТРАН-транслятор добавляет к каждой операции команду загрузки X-регистра и следующая команда будет глобальной косвенно-индексированной вместо одной прямой локальной в описываемом способе.

**Организация системных программных модулей.** Все секции исполняемой программы в зависимости от заданного параметра сборки RB-модулей редактором связей могут работать либо в SM-, либо в MM-режиме. Всякий системный модуль обслуживает какое-либо внешнее устройство и поэтому должен работать в MM-режиме. В описываемом программном обеспечении применяется три различных метода. Первый стандартный метод построения программных модулей заключается в написании хандлеров, когда для конкретного внешнего устройства (или, например, для УМСО — унифицированной магистральной системы обмена информацией [7]) программа обработки может быть логически разбита на элементарные операции ввода/вывода. Хандлер (точнее, вторая его половина) работает в MM-режиме, а программа, использующая обращения к хандлеру, как в MM-, так и в SM-режиме. Второй стандартный метод заключается в том, чтобы системные программы, использующие привилегированные команды, были оформлены в виде отдельных секций и подгружены во время генерации в качестве дополнительных секций монитора, работающего в MM-режиме. Как в первом, так и во втором случае программы могут работать в обоих режимах. Третий способ заключается в следующем. Можно написать и поместить в монитор во время его генерации две дополнительные секции монитора, которые бы меняли SM-режим на MM-режим и наоборот (посредством занесения соответствующей информации в ячейку с адресом 6 глобального сегмента данных исполняемой программы перед командой RSV), для того, чтобы программа работала в SM-режиме, а системные модули — в MM-режиме (аналогично 1-й и 2-й частям хандлера). Для передачи параметров можно применять COMMON-переменные. Если же мы хотим использовать прием параметров в ФОРТРАН-подпрограмму, то к этому моменту (или всегда) программа должна работать в SM-режиме (следствие работы ФОРТРАН-трансляторов, FORTD, по крайней мере, до 9-й версии). Поэтому при использовании первых двух методов отпадает возможность работы программ в MM-режиме, а следовательно, нельзя применять модули с привилегированными командами. В третьем способе из-за того, что редактор связей генерирует адреса с абсолютными значениями или со значениями относительно начала (G) программы для всей программы в целом, а не к каждому модулю в отдельности, и из-за трудности написания ассемблерной программы с одним типом адресов дополнительные секции, переключающие режим, тоже не могут решить проблемы передачи параметров в ФОРТРАН-модуль.

Первые два метода работы приемлемы, если системная часть обработки в целом больше не меняется, так как процесс отладки хандлеров и секций супервизора вместе с процессом генерации является достаточно сложной процедурой. Третий же метод применим с особой осторожностью к способам адресации для написания системных программ. Поэтому, за исключением описанных отдельно случаев при написании ФОРТРАН-программ, сборка редактором связей осуществляется в MM-режиме, а для передачи параметров используются либо COMMON-переменные, либо описанный выше способ фиксации переменных.

Система обычных и быстрых прерываний, принятая в архитектуре ЕС-1010, позволяет обрабатывать запросы, поступающие от внешних устройств ЭВМ независимо друг от друга в порядке их приоритета. В действующем комплексе на базе фотометрического автомата «Зенит-2» можно выделить следующие уровни прерываний и запросы на обработку:

запрос на считывание оптической плотности от сканирующего устройства автомата (обрабатывается при помощи системы быстрых прерываний)\*;

запрос на окончание обработки «зонного» сканирования (поступает в ЭВМ после того, как в режиме интерполяции будет окончен обход прямоугольника с оцифровыванием оптической плотности в каждой внутренней точке прямоугольника);

запрос на прекращение дальнейшей работы в связи со сбоем КАМАК-аппаратуры (нет сигнала-ответа «X»);

запрос на анализ координат (по X и Y) подвижной каретки стола (поступает при прохождении  $\epsilon$ -окрестности точки назначения);

сбой автоматической подстройки частоты или амплитудной характеристики по обеим координатам;

прерывание по управлению в системе УМСО;

прерывание по данным в системе УМСО.

Два уровня прерываний, отведенных для работы с УМСО, обслуживаются стандартным для ЕС-1010 образом за тем исключением, что любое стандартное внешнее устройство обслуживается одним уровнем прерывания. Анализ и обработка поступающих запросов оформлены в виде стандартной программы (хандлер УМСО), которая в процессе генерации монитора подсоединяется к остальным хандлерам системы. Все хандлеры являются резидентными программами, т. е. всегда находятся в памяти машины. Обращение к хандлеру УМСО из программы происходит при помощи заполнения стандартным образом управляющего блока, куда заносится вид приказа и характер его исполнения. Исполняемая программа может работать как в MM-, так и в SM-режиме.

Прерывания от КАМАК-аппаратуры управления «Зенитом-2» и его окружения (КАМАК-дисплей, цветной монитор и монитор серого тона) обслуживаются нестандартным для ЭВМ ЕС-1010 способом. Это вызвано главным образом ограниченностью оперативной памяти ЭВМ и применением оверлейной структуры исполнения программ, когда различные уровни прерываний и связанные с ним базовые программные модули могут лежать в параллельных ветвях. Если в данной программе не участвует какое-либо функциональное звено, то соответствующий модуль обслуживания не будет находиться в оперативной памяти, освобождая ее для других целей. Инициация работы базовых программных модулей, управляющих соответствующими блоками аппаратуры автомата (заполнение контекстов уровней прерывания, слов дезактивации прерываний, соответствующих таблиц монитора и т. д.), осуществляется вызовом иницирующих секций из модулей более высокого уровня или из программ пользователя.

Если в программе, не имеющей оверлейной структуры, используется один базовый модуль, работающий, например, с быстрыми прерываниями, то его иницирующая секция вызывается один раз в начале программы. Если же структура программы оверлейная и мы по каким-либо причинам не хотим иницирующую секцию помещать в ствол программы, то ее необходимо располагать в начале каждой ветви, применяющей остальные секции данного базового модуля. Если в программе используются два (или более) различных базовых модуля, работающих, например, с быстрыми прерываниями, то иницирующую секцию необходимо помещать всякий раз, когда происходит смена обращений к секциям другого базового модуля.

\* Остальные запросы, перечисленные в порядке уменьшения приоритета, обрабатываются при помощи системы обычных прерываний.

**Считывание оптической плотности.** Базовый программный модуль сбора цифровой информации со снимка IZENIT, обеспечивающий функционирование сканирующего устройства, позволяет выполнять следующие операции: установить луч в заданную точку области сканирования и измерить (среднюю) оптическую плотность в ней; собрать в буфер оперативной памяти для дальнейшей обработки значения плотностей на некотором фрагменте области сканирования; определить среднюю плотность на всем фрагменте или вдоль одного направления; передать в буфер гистограмму плотностей с фрагмента для определения верхней и нижней границ, дисперсии и т. п.

**КАМАК-монитор и полутоновые дисплеи.** Для оперативного просмотра участка поля сканирования в составе внешних специализированных устройств «Зенита-2» имеется устройство индикации полутоновой информации на аналоговом КАМАК-мониторе с телевизионной трубкой размером  $256 \times 256$  точек. Монитор может работать только параллельно со сканирующим устройством, которому задается режим зонного сканирования. Программные модули нижнего уровня, обслуживающие КАМАК-монитор с использованием подвижной каретки (см. ниже), обеспечивают динамический просмотр снимков, а также определение координат объектов на снимке.

Дисплеи обладают автономной памятью [2], позволяющей хранить информацию о телевизионном кадре, состоящем из  $384 \times 256$   $N$ -разрядных слов. Для дисплея серого тона  $N = 8$ , а для цветного  $N = 9$ . Интерпретация оптической плотности на дисплее серого тона: 0 — светлый участок снимка, 255 — темный. В случае синтезирования цветного изображения для каждого из трех основных цветов ( $R$  — красный,  $G$  — зеленый,  $B$  — синий) интерпретация аналогична черно-белому изображению: 0 — светлый участок соответствующего цвета, 255 — темный. В случае использования цветного монитора с целью повышения контрастности изображения применяется техническое раскрашивание (т. е. придание данной оптической плотности какого-либо оттенка цвета) при помощи одной из четырех специально подобранных цветовых гамм (клиньев). Имеется возможность во время работы программы в диалоговом режиме произвольно изменить клин. Основной недостаток — отсутствие возможности считывать содержимое памяти — в необходимых случаях восполняется программно: при записи в память информация записывается также и в рабочий файл для последующего считывания.

**Непосредственный доступ к изображению.** Обеспечиваемый автоматом «Зенит-2» непосредственный доступ к произвольному участку обрабатываемого изображения для широкого класса задач цифровой обработки изображений предоставляет существенные преимущества перед последовательным считыванием изображения до начала обработки (например, в задачах отслеживания линий уровня, границ образований и т. п.).

Базовый программный модуль перемещения области сканирования, помимо сброса, установки и чтения абсолютных координат, по существу отрабатывает один алгоритм перехода из текущей точки  $(X_0, Y_0)$  в заданную  $(X_1, Y_1)$ . Для этого в соответствующие регистры загружается векторная разность  $(X_1 - X_0, Y_1 - Y_0)$ , которую модули управления электродвигателями начинают обрабатывать, тем самым перемещая каретку к точке  $(X_1, Y_1)$ . Имеется два способа окончания отработки перемещения. Однозначного сообщения о том, что «рассогласование»  $(X_1 - X_0, Y_1 - Y_0)$  отработано, от КАМАК-модуля нет, поэтому, после того как каретка оказывается в некоторой окрестности точки  $(X_1, Y_1)$ , программным способом осуществляется временная задержка (первый способ), затем проверяется истинное местонахождение каретки, и, в случае если разность не больше допустимой, переход считается осуществленным (в противном случае задержка повторяется). Второй способ заключается в подсчете количества перемен знака регистра приращення до заданной величины по обеим координатам.

**Заключение.** Система базовых программных модулей (системное программное обеспечение) является совместимой, легко расширяемой, недублирующей написание сходных по конечному результату программ; она доступна программисту-пользователю, реализующему свой алгоритм на языке высокого уровня (FORTRAN-IV, PLR-10). Модули написаны на Ассемблере и оптимизированы либо по объему занимаемой оперативной памяти, либо по быстродействию; они обеспечивают удобство работы с внешними устройствами в диалоговом режиме. Программист-пользователь освобожден от необходимости подробно знакомиться с КАМАК-аппаратурой и командами управления, слежения и коррекции автомата и его окружения, так как он может использовать программные модули, реализующие конкретные функции ввода/вывода и предварительной обработки данных на уровне вызова с параметрами, с фиксированием переменных либо через общие (COMMON) переменные. На основе этих модулей созданы и создаются диалоговые и полностью автоматизированные системы обработки изображений, использующие все возможности фотометрического автомата, его окружения и дисковой операционной системы ЕС-1010. Возможность применения оверлейной структуры исполняемых программ, применяющих описанные модули нижнего и среднего уровней, позволяет в известной степени уменьшить влияние ограниченности оперативной памяти.

#### ЛИТЕРАТУРА

1. Яковенко Н. С. Базовое программное обеспечение комплекса обработки аэрокосмической информации.— Автометрия, 1980, № 4.
2. Бурый Л. В. и др. Обработка изображений.— Автометрия, 1980, № 3.
3. Нестерихин Ю. Е., Пушной Б. М. О системе автоматической обработки изображений.— Автометрия, 1977, № 3.
4. Система обслуживания файлов на диске FMS-10: Руководство пользователя. Видеотон, 1975.
5. Яковенко Н. С. Система подготовки и редактирования текстовой информации для ЕС-1010.— Новосибирск: изд. ИАиЭ СО АН СССР, 1979. (Препринт/АН СССР, Сиб. отд-ние, ИАиЭ; № 110).
6. Редакторы связей: Руководство пользователя. Видеотон, 1974.
7. Бредихин С. В. и др. Программирование средств связи и управления вводом/выводом в унифицированной магистральной системе обмена (УМСО).— В кн.: Автоматизация эксперимента. Новосибирск: изд. ИАиЭ СО АН СССР, 1976.

*Поступила в редакцию 4 февраля 1982 г.*

УДК 621.378.9 : 778.4

**А. Г. КОЗАЧОК**  
(Новосибирск)

### ВОПРОСЫ АВТОМАТИЗАЦИИ ПРОЧНОСТНЫХ ИССЛЕДОВАНИЙ НА ОСНОВЕ МЕТОДОВ КОГЕРЕНТНОЙ ОПТИКИ И ГОЛОГРАФИИ

Исследование процессов механического разрушения предполагает экспериментальное определение большого числа характеристик на макро- и микроуровнях. Внимание, которое привлекают когерентно-оптические методы для проведения таких экспериментов, в значительной степени объясняется тем, что они являются комплексными и позволяют в рамках единой измерительной системы получить разностороннюю информацию о процессе разрушения.

На рис. 1 приведена блок-схема, показывающая взаимоотношения параметров прочности и методов голографической интерферометрии. Как