5. **Кильдишев Г. С., Аболенцев Ю. И.** Многомерные группировки.— М.: Статистика, 1978.
6. **Елисеева И. И., Рукавишников В. О.** Группировка, корреляция, распознавание образов.— М.: Статистика, 1977, с. 125—135.
7. **Тюмиков Д. К., Кацюба О. А.** О выборе доминантных переменных в задаче идентификации химико-технологических объектов.— В кн.: Автоматизация химических производств. М.: НИИТЭХИМ, 1981, вып. 3.
8. **Дуб Дж.** Вероятностные процессы.— М.: ИЛ, 1956.

*Поступила в редакцию 16 марта 1981 г.;
окончательный вариант — 10 октября 1982 г.*

УДК 681.32.05

И. В. БЕЛАГО, А. И. ПИНЧУК, М. А. СТАРКОВ

(Новосибирск)

АНАЛИЗАТОР БИНАРНЫХ ИЗОБРАЖЕНИЙ

В последнее время появилось направление синтаксического (языкового) описания изображений в задачах распознавания образов. Рассматриваемый ниже алгоритм по своей структуре может быть отнесен к алгоритму синтаксического анализа изображений, однако результатом его работы является таблица описания изображений, которая дает информацию для управления процессом обработки в автоматическом и диалоговом режимах.

Введем определения. Пару чисел (j, k), где j и k — целые, будем называть точкой, а множество точек $\{(j-1, k), (j+1, k), (j, k-1), (j, k+1)\}$ — окрестностью (j, k)-й точки. Последовательность (j_1, k_1), $(j_2, k_2), \ldots, (j_n, k_n)$ назовем путем, соединяющим точки (j_1, k_1) и (j_n, k_n), если любая рядом стоящая пара точек последовательности принадлежит окрестности друг друга, а значение изображения во всех этих точках равно единице. Две точки будем считать связными, если существует хотя бы один соединяющий их путь. Множество точек назовем объектом (или связной областью), если любые две точки объекта связны.

Определим «дыру» в объекте. Для этого исключим на матрице все объекты, кроме данного. Полной окрестностью (j, k)-й точки будет объединение обычной окрестности и множества $\{(j-1, k-1), (j-1, k+1), (j+1, k-1), (j+1, k+1)\}$. Сохраним определения пути, связности и объекта для нового определения окрестности, потребовав равенства нулю всех точек пути. «Дырой» назовем связную область, состоящую из нулей и не имеющую общих точек с границей матрицы (в новом определении окрестности).

Сформулируем задачу. Пусть автомату предъявляется бинарная матрица поэлементно слева направо, сверху вниз. Требуется наделить автомат такой структурой, чтобы после предъявления последней точки изображения в его памяти была сформирована таблица описания объектов (ТО), в которой каждому объекту соответствовал бы столбец, содержащий следующие данные: a, l — номера строки и столбца, в которых объект встречается впервые; b — номер строки, в которой объект заканчивается; c, d — соответственно крайний левый и крайний правый столбцы, в которых встречается объект; S — площадь объекта (число составляющих его точек); r — число «дыр» в объекте; x, y — координаты центра тяжести, определяемые по формулам

$$x = \frac{1}{S} \sum_{(j,k)} j, \quad y = \frac{1}{S} \sum_{(j,k)} k; \qquad (1)$$

t_x, t_y, t_{xy} — компоненты тензора инерции объекта, вычисляемые из выражений

$$t_x = \frac{1}{S}\sum_{(j,k)}(j-x)^2; \quad t_y = \frac{1}{S}\sum_{(j,k)}(k-y)^2; \quad t_{xy} = \frac{1}{S}\sum_{(j,k)}(j-x)(k-y), \quad (2)$$

причем суммирование в (1) и (2) проводится по всем (j, k), принадлежащим описываемому объекту.

Для определенности будем считать, что если значение изображения в точке равно единице, то эта точка принадлежит какому-либо объекту, в противном случае значение в точке равно нулю.

Назовем j, k-м сечением матрицы последовательность ее элементов $S_{j,k} = (j, 1), (j, 2), \ldots, (j, k), (j-1, k), (j-1, k+1), \ldots, (j-1, n)$, состоящую из $n+1$ элементов, где n — длина строки. Заметим, что если (j, k) — последний предъявленный автомату элемент, то $S_{j,k}$ разделяет матрицу на две части: предъявленную и непредъявленную.

Отрезком сечения будем считать множество подряд стоящих единиц в последовательности $S_{j,k}$. Два отрезка полагаем связными, если между ними существует путь, проходящий в предъявленной части матрицы. Отсюда следует, что если два отрезка связны, то они принадлежат одному объекту. Обратное утверждение неверно, так как связь может образоваться в непредъявленной части матрицы. Таким образом, чтобы решить вопрос о принадлежности множества точек одному объекту, автомату достаточно при предъявлении очередной точки матрицы следить за связностью отрезков во вновь образовавшемся сечении $S_{j,k+1}$. Для кодирования связности отрезков будем применять следующие обозначения: «0» — для отрезка, не связного ни с одним отрезком сечения, «(» — для первого в группе связных отрезков, «)» — для последнего в группе связных отрезков, «1» — если и слева, и справа от отрезка есть отрезки, с ним связные. Последовательность символов, соответствующих отрезкам сечения (в том порядке, в котором они следуют в $S_{j,k}$), назовем словом состояния сечения и обозначим через $H_{j,k}$.

Состояние автомата характеризуем значением следующих переменных. Вышеопределенная ТО представляет собой матрицу размерностью $13 \times N$. Из определения ТО следует, что для описания объекта достаточно 12 строк, 13-я строка вводится для удобства реализации алгоритма, в ней записывается признак завершения описания объекта. Величина N определяется максимальным числом объектов, пересекаемых сечением, и равна $\left]\frac{n+1}{2}\right[$.

В случае переполнения ТО инициируется программа, которая просматривает 13-ю строку; она находит объекты, описание которых закончено, и выводит их на внешний носитель. Таблица КС (код слова) состоит из двух строк. В 1-й строке записывается $H_{j,k}$, во 2-й — адрес объекта (номер столбца ТО), которому принадлежит закодированный в 1-й строке отрезок. Такая организация таблицы КС удобна тем, что группа связных отрезков имеет одинаковую адресную часть.

Назовем (j, k)-е состояние автомата правильным, если ТО верно отражает описание всех объектов (такое, каким оно было бы при равенстве непросмотренной части матрицы нулю).

Правила перевода автомата из (j, k)-го состояния в $(j, k+1)$-е зависят от реализации значений в следующей четверке точек: (j, k), $(j, k+1)$, $(j-1, k)$, $(j-1, k+1)$ *.

Очевидно, что комбинации $\begin{matrix}00\\00\end{matrix}, \begin{matrix}01\\00\end{matrix}, \begin{matrix}11\\00\end{matrix}, \begin{matrix}00\\10\end{matrix}, \begin{matrix}01\\10\end{matrix}, \begin{matrix}10\\10\end{matrix}$ не изменяют состояния автомата. Рассмотрим комбинацию $\begin{matrix}11\\11\end{matrix}$. Нетрудно видеть, что КС не изменяется, однако к объекту по адресу, указанному во 2-й строке,

---

* **Старков М. А., Трофимов О. Е., Фризен Д. Г.** Автомат для подсчета числа плоских объектов.— Автоматика и телемеханика, 1976, № 5.

присоединилась еще одна точка, поэтому переменные в ТО должны быть пересчитаны по следующим формулам:

$$S := S + 1, \tag{3}$$

$$\left.\begin{aligned} x &:= x + j, \\ y &:= y + k + 1, \end{aligned}\right\} \tag{4}$$

$$t_x := t_x + j^2;\ t_y := t_y + (k+1)^2;\ t_{x,\,y} + (k+1)j \tag{5}$$

(приведение ТО к окончательному виду, определенному выражениями (1), (2), осуществляется после предъявления всей матрицы).

Комбинации $\begin{matrix}01\\01\end{matrix}$ и $\begin{matrix}11\\01\end{matrix}$ приводят к аналогичным изменениям в ТО. Коррекции может потребовать также переменная b, а поскольку в ТО всегда $b \leqslant j$, достаточно сделать присвоение

$$b := j. \tag{6}$$

Для комбинаций $\begin{matrix}00\\11\end{matrix}$ и $\begin{matrix}10\\11\end{matrix}$ ТО корректируется выражениями (3)—(5); $(j,\ k+1)$-я точка может оказаться самой правой точкой объекта, поэтому необходимо выполнить следующий оператор:

$$\text{если } d < k+1, \text{ то } d := k+1. \tag{7}$$

Перечисленные выше комбинации не изменяли слова состояния $H_{j,\,k}$, в чем нетрудно убедиться из рассмотрения сечений $S_{j,\,k}$ и $S_{j,\,k+1}$. Приступим к исследованию более сложных ситуаций. Проанализируем комбинацию $\begin{matrix}00\\01\end{matrix}$. Нетрудно видеть, что в новом сечении появился несвязный отрезок, который может быть первой точкой нового объекта, поэтому по соответствующему адресу в ТО сделаем такие присвоения:

$$\left.\begin{aligned} &a := j,\ b := j,\ c := k+1, \\ &d := k+1,\ e := k+1,\ S := 1, \\ &r := 0,\ x := j,\ y := k+1, \\ &t_x := j^2,\ t_y := (k+1)^2,\ t_{x,y} := j\,(k+1). \end{aligned}\right\} \tag{8}$$

Слово $H_{j,\,k} = \ldots\alpha\beta\ldots$ перейдет в слово $H_{j,\,k+1} = \ldots\alpha 0\beta\ldots$. Описанную процедуру назовем оператором ОТКРОЙ ОБЪЕКТ.

В комбинации $\begin{matrix}1\ 0\\0\ 0\end{matrix}$ отрезок, состоящий из одной «1» в старом сечении, в новом — исчезает. Если ему в $H_{j,\,k}$ соответствует код «0», то это является необходимым и достаточным условием того, что $(j-1)$-я точка — последняя в объекте. Слово $H_{j,\,k} = \ldots\alpha 0\beta\ldots$ преобразуется в $H_{j,\,k+1} = \ldots\alpha\beta\ldots$, коды остальных отрезков не изменяются. При программной реализации алгоритма в служебной строке записывается признак окончания объекта, по которому он может быть переписан на внешний носитель.

Если отрезку в $H_{j,\,k}$ соответствует код «1», то слово $H_{j,\,k} = \ldots\alpha 1\beta\ldots$ преобразуется в $H_{j,\,k+1} = \ldots\alpha\beta\ldots$. В случае если код отрезка равняется «(», справа от него необходимо найти ближайший к нему связный отрезок (по равенству адресных частей) и заменить «1» на «(» или «)» на «0». Для отрезка, имевшего код «)», слева от него нужно найти ближайший к нему связный и заменить «1» на «)» или «(» на «0».

Для комбинации $\begin{matrix}1\ 0\\0\ 1\end{matrix}$ слово $H_{j,\,k} = \ldots\alpha\gamma\beta\ldots$ преобразуется в $H_{j,\,k+1} = \ldots\alpha' 0\beta'\ldots$, коды остальных отрезков в зависимости от значения γ изменяются так же, как и в комбинации $\begin{matrix}1\ 0\\0\ 0\end{matrix}$, далее выполняется оператор ОТКРОЙ ОБЪЕКТ.

В комбинации $\begin{smallmatrix}1 & 1\\ 1 & 0\end{smallmatrix}$ слово $H_{j,k} = \ldots\alpha\gamma\beta\ldots$ преобразуется в $H_{j,k+1} = \ldots\alpha\gamma_1\gamma_2\beta\ldots$, величины γ_1 и γ_2 определяются значением γ:

$$\left.\begin{array}{l}\text{если } \gamma = 0, \text{ то } \gamma_1 := (,\ \gamma_2 :=);\\ \text{если } \gamma = 1, \text{ то } \gamma_1 := 1,\ \gamma_2 := 1;\\ \text{если } \gamma = (\ , \text{ то } \gamma_1 := (\ ,\ \gamma_2 := 1;\\ \text{если } \gamma =)\ , \text{ то } \gamma_1 := 1,\ \gamma_2 :=).\end{array}\right\} \qquad (9)$$

Рассмотрим последнюю комбинацию $\begin{smallmatrix}0 & 1\\ 1 & 1\end{smallmatrix}$, в которой два отрезка сливаются в один. Здесь могут возникнуть два варианта: либо отрезки принадлежат одному объекту, либо разным, что легко выяснить сравнением их адресных частей. В первом варианте ТО должна быть изменена по формулам (3)—(5). Слово $H_{j,k} = \ldots\alpha\gamma_1\gamma_2\beta\ldots$ преобразуется в $H_{j,k+1} = \ldots\alpha\gamma\beta\ldots$. В табл. 1 показаны все возможные реализации γ_1 и γ_2 для этого варианта и значение γ, которое должно им соответствовать. Из табл. 1 видно, что во всех случаях существует замкнутый путь, идущий из точки $(j,\ k+1)$ через $(j,\ k)$ к точке $(j-1,\ k+1)$, внутри которого находится как минимум одна точка $(j-1,\ k)$ с нулевым значением. Следовательно, в ТО надо скорректировать число «дыр» и сделать присвоение $r := r+1$.

Во втором варианте, когда два объекта сливаются вместе, один из столбцов ТО должен быть вычеркнут. Другой столбец необходимо скорректировать следующим образом:

$$\left.\begin{array}{l}\text{если } d_2 > d_1, \text{ то } d_1 := d_2;\\ S_1 := S_1 + S_2 + 1;\ r_1 := r_1 + r_2;\\ x_1 := x_1 + x_2 + j;\ y_1 := y_1 + y_2 + k + 1;\\ t_{x_1} := t_{x_1} + t_{x_2} + j^2;\ t_{y_1} := t_{y_1} + t_{y_2} + (k+1)^2;\\ t_{x_1y_1} := t_{x_1y_1} + t_{x_2y_2} + j\,(k+1).\end{array}\right\} \qquad (10)$$

Операторы (10) следует выполнять тогда, когда вычеркивается объект с большим адресом. Вычеркивание объекта с меньшим адресом приводит к увеличению числа выполняемых операций, в чем легко убедиться, рассмотрев возникающие ситуации. Во 2-й строке таблицы КС следует везде заменить адрес A_2 на адрес A_1. Код слова $H_{j,k} = \ldots\alpha\gamma_1\gamma_2\beta\ldots$ преобразуется в $H_{j,k+1} = \ldots\alpha'\gamma\beta'\ldots$. Все возможные значения γ_1 и γ_2 и соответствующие им значения γ показаны в табл. 2. В комбинациях, отмеченных звездочкой, необходимо найти справа символ «)», отвечающий γ_2, и заменить его на «1», в комбинациях, отмеченных двумя звездочками, следует найти слева символ «(» и заменить его на «1».

Для правильной работы алгоритма к изображению слева нужно присоединить нулевой столбец, а снизу и сверху — нулевые строки.

Рассмотрим применение алгоритма для выделения прямолинейных границ на изображении, представленном цифровым массивом 1024×1024 элемента (рис. 1). Исходное многоградационное изображение переводилось в бинарное с помощью следующей процедуры. На исходном изображении выбирались четверки точек с координатами $\{(j,\ k),\ (j+1,\ k),\ (j,\ k+1),\ (j+1,\ k+1)\}$, значения в которых сравнивались между собой. Если на четверке точек наблюдались два различных значения, то $(j,\ k)$-му элементу бинарной матрицы присваивалось значение, равное единице. В случае реализации на опорной четверке одного, трех или четырех

Таблица 1

γ_1	γ_2	γ
1	1	1
1	)	)
(	1	(
(	)	0

Таблица 2

γ_1	γ_2	γ
0	δ	δ
δ	0	δ
1	(	1 *
(	(	(*
)	1	1 **
)	)	) **
)	(	1

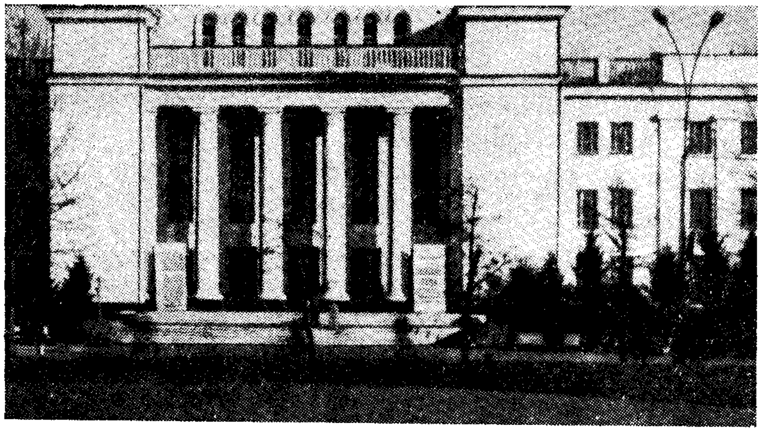

Рис. 1.

различных значений соответствующему элементу бинарной матрицы присваивалось значение нуля. При этом два значения a и b считались равными, если $a \in [b-\delta,\ b+\delta]$, где δ — константа, определяемая контрастностью снимка. Таким образом, на изображении выделялось множество квадратов, с наибольшей вероятностью содержащих границу. Квадраты, в которых наблюдались пересечения двух и более границ, из рассмотрения исключались. Далее бинарная матрица обрабатывалась в соответствии с изложенным алгоритмом, в результате чего получены таблицы описаний связных областей. По ТО определялись диагонали прямоугольников, описывающих связные области, и выделялись те из них, площади которых были соизмеримы с диагоналями. Для найденных

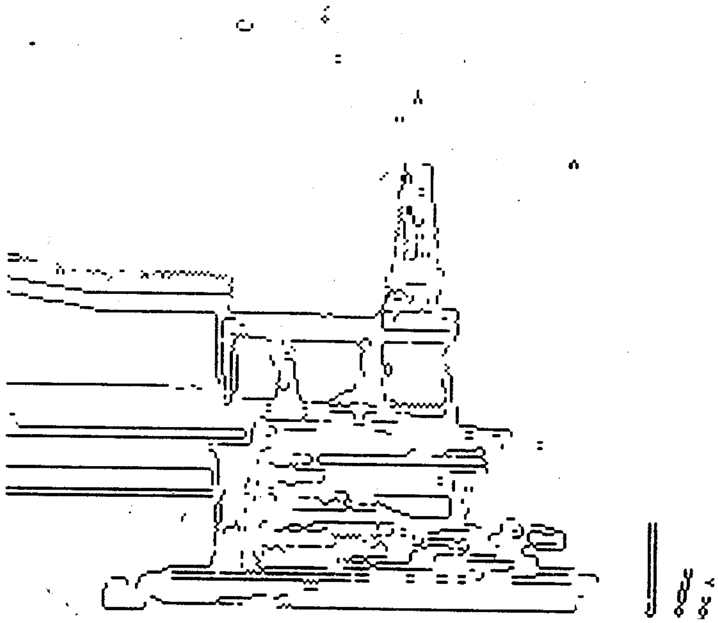

Рис. 2.

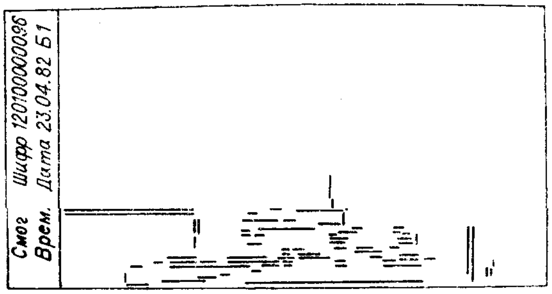

Рис. 3.

таким образом областей вычислялись главные направления и собственные числа тензора инерции, и если отношения собственных чисел превышали заданный порог, то область идентифицирована инерции, так что ее длина лимитировалась габаритами связной области.

Бинарная матрица представлена на рис. 2, на рис. 3 показано множество выделенных границ, вычерченных графопостроителем. Алгоритм был реализован на языке ФОРТРАН ЭВМ БЭСМ-6, вся процедура потребовала менее 2 мин процессорного времени. На рисунке показан фрагмент изображения.

С помощью указанного алгоритма описания связных областей можно полностью решить ряд задач, относящихся к обработке некоторых узких классов изображений. К ним относятся также задачи по обработке биологических структур, треков и т. д. Однако основным назначением алгоритма следует считать предварительную обработку изображений по приведенной выше схеме, т. е. бинаризацию изображений и описание связных областей.

Алгоритм бинаризации может быть модифицирован следующим образом. Элементу бинарной матрицы будем присваивать нуль тогда и только тогда, когда все значения опорной четверки совпадают. Таким образом, контрастным объектом изображения удается поставить в соответствие связную область бинарной матрицы. Объект изображения получает в ТО идентификатор в виде номера столбца, а также параметры, позволяющие судить о его форме, размерах и т. д. Оператор может вызвать объект по номеру или признаку на экран дисплея (весь прямоугольник или замаскированный связной областью бинарной матрицы), присвоить ему имя, выбрать программу для его обработки из пакета прикладных программ и проделать ряд других процедур. Возможен и такой вариант: если программа не может идентифицировать объект, она выводит его на экран для распознавания оператором. Таким образом, получаем язык для диалоговой обработки изображений. Коррекция изображения возможна на уровне редактирования бинарной матрицы.

*Поступила в редакцию 10 декабря 1981 г.;
окончательный вариант — 16 июля 1982 г.*