С. Б. АЛАВЕРДЯН, Л. М. ДЖАВАХЯН

АЛГОРИТМЫ ВЫДЕЛЕНИЯ ГРАНИЦ, ОСТОВА И ЗАМКНУТЫХ ОБЪЕКТОВ НА БИНАРНЫХ ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ КЛЕТОЧНОЙ ЛОГИКИ

1. Введение

При решении различных прикладных задач обработки многомерных сигналов (полей) часто приходится анализировать полученные экспериментальные снимки с целью выявления определенных характеристик изображений—выделение остова и границ, нахождение замкнутых объектов и т. д.

Нарасимхан и Шерман [1] рассматривали эти процедуры в распознавании рукописных букв и в анализе фотографий пузырковой камеры, Престон [2]—в некоторых медицинских исследованиях, Мацелло [3], Кийко и Шлезингер [4]—в обработке графических изображений.

В настоящей статье предлагаются новые алгоритмы выполнения этих операций на бинарных (двухградационных) изображениях с использованием в качестве математического аппарата—клеточной логики [2, 5, 6].

Для данных операций существуют характерные условия, которые должны учитываться в процессе обработки, а именно, необходимо

1) чтобы два отдельных объекта не слились,

2) чтобы ни один объект не разделился,

3) чтобы после некоторого числа итераций изображение перестало изменяться,

4) чтобы происходило уменьшение информационного содержания изображения (т. е. сокращение количества единиц, необходимых для представления нужной информации), и т. д.

Естественного требования—максимальной скорости вычислений можно достичь при параллельной работе алгоритмов. В связи с этим представляется интересной теория клеточных автоматов, с помощью которой на изображениях с особой эффективностью параллельно выполняются такие локальные операции как выделение границ, остова и требуемых объектов, рассматриваемые в данной работе.

2. Принцип клеточных операций

Клеткой называется пара $(a \times m) \in A \times M$, где A—конечный алфавит M—множество имен с мощностью не более, чем счетной. Слово —это конечное множество клеток $w = \{(a_0, m_0), \cdots, (a_n, m_n)\}$, которое удовлетворяет условию—$m_i \neq m_j$ $(i \neq j,\ i, j = 0,1, \cdots, n)$. Век-

торы $P_{r_1}(w)=(a_0, \cdot\cdot\cdot, a_n)$ и $P_{r_2}(w)=(m_0, \cdot\cdot\cdot, m_n)$ называются соответственно первой и второй проекциями слова w.

Конфигурацией называется множество слов, у которых вторыми проекциями являются значения совокупности упорядоченных функций $(\varphi_0(x), \cdot\cdot\cdot, \varphi_n(x))$ для всех $x \in M$, при этом $\varphi_i : M \to M$, $\varphi_0(x) \neq \varphi_1(x) \neq \cdot\cdot\cdot \neq \varphi_n(x)$.

Матрице $F=(f_{ij})$ $i, j=\overline{1,N}$ изображения ставится в соответствие слово $\Gamma=\{\gamma_{(i,j)}\}_{(i,j)=\overline{1,N^2}}$ где $\gamma_{(i,j)}=(f_{ij}, [i,j])$.

$f_{i,j}=\{0,1\}$ называется значением, а $[i,j]=N(i-1)+j$ именем клетки. Γ является объектом воздействия клеточных преобразований или подстановок, которые имеют следующий вид:

$$w : S_1 * S_2 \to S_3,$$

где S_1, S_2, S_3—конфигурации. Отображение w осуществляет переход S_1 в S_3 при условии S_2. S_2 называется контекстом. В нашем случае вторая проекция меняется в зависимости от i и j по правилу $[i,j]=N(i-1)+j$ и в итоге, клеточные операции как бы „пробегают" всю плоскость изображения.

Более простым и легкореализуемым является тот случай, когда

$$P_{r_2}(S_1)=P_{r_2}(S_3)$$

Отметим, что такие подстановки называются стационарными [5].

Обычно при использовании клеточного аппарата в обработке изображений в качестве S_2 берется некоторая совокупность соседей тех клеток, которые содержатся в S_1.

В [2] операции клеточной логики определяются как «цифровые» операции преобразования некоторого массива данных $F(i, j)$ в новый массив.

Значение каждого элемента в новом массиве определяется его значением в исходном массиве и исходными значениями его ближайших «соседей».

Мы воспользуемся клеточными преобразованиями, придерживаясь этого определения и рассматривая в качестве таких соседей для каждой точки ее окрестность размером 3×3 (окрестность Мура).

3. Алгоритмы

В рассматриваемых процедурах общим является то, что некоторые элементы изображения удаляются с учетом расположения соседних элементов. Поэтому, во всех клеточных подстановках, которыми ниже опишутся предлагаемые алгоритмы, $S_1=\{(1,[i,j])\}$, $S_3=\{(0,[i,j])\}$ а S_2 состоит из клеток—соответствующих соседним элементам точки с координатой (i,j).

На бинарных изображениях граничными называются те точки, для которых хотя бы один из четырех близких соседей принадлежит фону изображения (множеству нулевых элементов). Для получения таких точек применяется следующая подстановка:

$$K : \{(1,[i,j])\} * \{(1,[i,j-1])(1,[i-1, j)(1,[i+1,j])(1,[i, j+1])\} \to \{(0,[i,j])\},$$

которая удаляет центральный элемент на рис. 1.

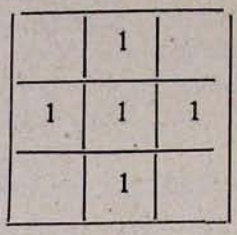


Рис 1.

Отметим, что объектом на изображении считаем каждое отдельное множество единиц, связанных по принципу восьмисвязности. Остов изображенного объекта получают выделением всех точек, равноудаленных от двух точек на границе объекта [5]. Система подстановок, применением которой получаем остов изображения, имеет такой вид:

$$
V=\begin{cases}
V_1=\{(1,[i,j])\}*\{(0,[i-1,j-1])(0,[i-1,j])(0,[i,j-1])(1,[i,j+1])(1,[i+1,j])\}\to\{(0,[i,j])\}\\
V_2=\{(1,[i,j])\}*\{(1,[i-1,j])(0,[i,j-1)(1,[i,j+1])(0,[i+1,j-1])(0,[i+1,j])\}\to\{(0,[i,j])\}\\
V_3=\{(1,[i,j])\}*\{(1,[i-1,j])(0,[i,j-1])(1,[i,j+1])(1,[i+1,j])\to\{(0,[i,j])\}\\
V_4=\{(1,[i,j])\}*\{(0,[i-1,j])(1,[i,j-1])(1,[i,j+1])(1,[i+1,j])\}\to\{(0,[i,j])\}\\
V_5=\{(1,[i,j])\}*\{(0,[i-1,j])(0,[i-1,j-1])(1,[i,j-1])(0,[i,j+1])(1,[i+1,j])\}\to\{(0,[i,j])\}\\
V_5=\{(1,[i,j])\}*\{(0,[i-1,j])(0,[i-1,j-1])(1,[i,j-1])(0,[i,j+1])(1,[i+1,j])\}\to\{(0,[i,j])\}\\
V_6=\{(1,[i,j])\}*\{(1,[i-1,j])(1,[i,j-1])(0,[i,j+1])(0,[i+1,j])(0,[i+1,j+1])\}\to\{(0,[i,j])\}\\
V_7=\{(1,[i,j])\}*\{(1,[i-1,j])(1,[i,j-1])(0,[i,j+1])(1,[i+1,j])\}\to\{(0,[i,j])\}\\
V_8=\{(1,[i,j])\}*\{(1,[i-1,j])(1,[i,j-1])(1,[i,j+1])(0,[i+1,j])\}\to\{(0,[i,j])\}
\end{cases}
$$

Легко заметить, что эти подстановки удаляют граничные элементы объектов — (центральные элементы на рис. 2).

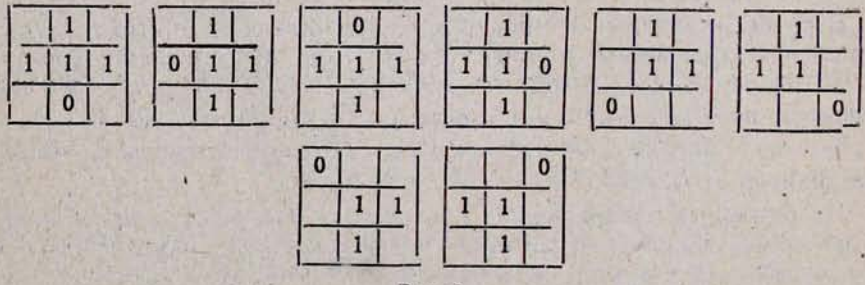


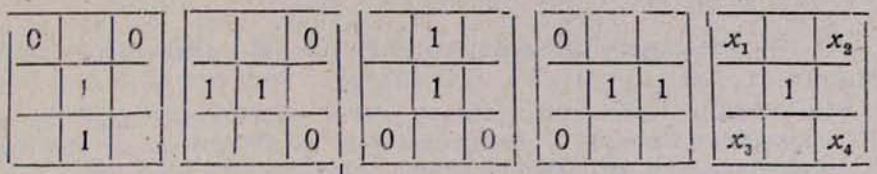
Рис. 2

Удаление при этом выполняется симметрично во всех направлениях. Во время одной итерации подстановки работают в следующем порядке

а) V_1, V_2, V_3,
б) V_1, V_5, V_4,
в) V_5, V_6, V_7,
г) V_2, V_6, V_8,

а на каждом шаге все V_l работают параллельно.

После применения алгоритма V, полученные на изображении линии остова имеют толщину равную единице. Из этих линий замкнуты и непрерывны те, каждый элемент которых в своей окрестности размером имеет, по меньшей мере, два других элемента, принадлежащих этим же линиям. Следовательно, для получения таких линий надо освободиться от ситуаций, показанных на рис. 3, что осуществляет следующая система:



$$\left(\sum_{l=1}^{4} x_l = 1\right)$$

Рис. 3

$$Z=\begin{cases} z_1=\{(1,[i,j])\}*\{(1,[i-1,j])(0,[i+1,j-1])(0,[i+1,j+1])\}\to\{(0,[i,j]\} \\ z_2=\{(1,[i,j])\}*\{(1,[i,j-1])(0,[i-1,j+1])(0,[i+1,j+1])\}\to\{(0,[i,j])\} \\ z_3=\{(1,[i,j])\}*\{(1,[i+1,j])(0,[i-1,j-1])(0,[i-1,j+1])\}\to\{(0,[i,j])\} \\ z_4=\{(1,[i,j]\}*\{(1,[i,j+1])(0,[i-1,j-1])(0,[i+1,j-1])\}\to\{(0,[i,j])\} \\ z_5=\{(1,[i,j])\}*\{(0,[i-1,j-1])(0,[i-1,j+1])(0,[i+1,j-1])(0,[i+ \\ \quad +1,j+1])\}\to\{(0,[i,j])\} \\ z_6=\{(1,[i,j])\}*\{(1,[i-1,j-1])(0,[i-1,j+1])(0,[i+1,j-1])(0,[i+ \\ \quad +1,j+1])\}\to\{(0,[i,j)\} \\ z_7=\{(1,[i,j])\}*\{(0,[i-1,j-1])(1,[i-1,j+1])(0,[i+1,j-1])(0,[i+1, \\ \quad j-1])(0,[i+1,j+1])\}\to\{(0[i,j])\} \\ z_8=\{(1,[i,j])\}*\{(0,[i-1,j-1])(0,[i-1,j+1])(1,[i+1,j-1])(0,[i+1, \\ \quad j+1])\}\to\{(0,[i,j])\} \\ z_9=\{(1,[i,j])\}*\{(0,]i-1,j-1])(0,[i-1,j+1])(0,[i+1,j-1])(1,[i+ \\ \quad +1,j+1])\}\to\{(0,[i,j])\} \end{cases}$$

Представим Γ и полученный в итоге Γ' в виде

$$\Gamma=\left\{\bigcup_{l=1}^{k}\Gamma_p\right\}\cup\Gamma_0, \qquad \Gamma'=\left\{\bigcup_{z=1}^{m}\Gamma'_z\right\}\cup\Gamma'_0$$

где Γ_p $(l=\overline{1,k})$ соответствуют объектам в F, Γ'_z—в F', а Γ_0 и Γ'_0—фонам F и F'. Изложенные в первом пункте условия терминами клеточной логики можно выразить так;

1) Из равенства $P_{r_2}(\Gamma_p)\cap P_{r_2}(\Gamma_e)=\varnothing\,(l,p=\overline{1,k})$ следует, что не существует $\Gamma'_z\subset\Gamma'$ такое, что $P_{r_1}(\Gamma'_z)\cap P_{r_1}(\Gamma_p)=\varnothing$, $P_{r_2}(\Gamma'_z)\cap P_{r_2}(\Gamma_e)=\varnothing$.

2) Для любого $\Gamma_p \subset \Gamma(p=\overline{1,k})$ существует $\left.\bigcup_{t=1}^{T} \Gamma'_{p_t}\right\} \subset \Gamma'\ (T<m)$

такое, что

$$\rho_{r_1}(\Gamma_p) \cap \rho_{r_1}(\Gamma'_{p_1}) \neq \varnothing,$$
$$\rho_{r_2}(\Gamma_p) \cap \rho_{r_2}(\Gamma'_{p_2}) \neq \varnothing,$$
$$\cdots\cdots\cdots$$
$$\rho_{r_3}(\Gamma_p) \cap \rho_{r_3}(\Gamma'_{p_T}) \neq \varnothing.$$

3) Существует i такое, что $\Phi(\Gamma^{i-1}) = \Gamma^{i-1}$.

4) $\sum_{t=1}^{T} \left|\Gamma'_{p_t}\right| \leqslant \left|\Gamma_p\right|$

Легко убедиться, что алгоритмы V, K, и Z удовлетворяют этим условиям (см. рис. 5 –8).

Как операции клеточной логики, эти алгоритмы обладают свойством непротиворечивости, необходимым для их параллельной работы и для синтезирования реализующей эту работу сети клеточных автоматов [5, 6].

4. Последовательная реализация алгоритмов

4.1. *Выделение границ.* Строятся вспомогательные матрицы A и B a_{ij}, $b_{ij} \in \{0, 1, 2, 3, 4\}$ по такому принципу:

если $f_{ij}=0$ то $a_{ij}=b_{ij}=0$,

если $f_{ij}\neq 0$ то

$$a_{ij} = \begin{cases} 1 \text{ при } f_{i,j-1}=0 \text{ и } f_{i,j+1}=0 \\ 2 \text{ при } f_{i,j-1}=0 \text{ и } f_{i,j+1}\neq 0 \\ 3 \text{ при } f_{i,j-1}\neq 0 \text{ и } f_{i,j+1}=0 \\ 4 \text{ при } f_{i,j-1}\neq 0 \text{ и } f_{i,j+1}\neq 0 \end{cases}$$

$$b_{ij} = \begin{cases} 1 \text{ при } f_{i-1,j}=0 \text{ и } f_{i+1,j}=0 \\ 2 \text{ при } f_{i-1,j}=0 \text{ и } f_{i+1,j}\neq 0 \\ 3 \text{ при } f_{i-1,j}\neq 0 \text{ и } f_{i+1,j}=0 \\ 4 \text{ при } f_{i-1,j}\neq 0 \text{ и } f_{i+1,j}\neq 0 \end{cases}$$

На рис. 4 показан пример построения этих матриц

$$F = \begin{vmatrix} 000000000000000000000 \\ 000000000010000000000 \\ 000000000111000000000 \\ 000000001111100000000 \\ 000000011111110000000 \\ 000000111111111000000 \\ 000001111111111100000 \\ 000011111111111110000 \\ 000111111111111111000 \\ 001111111111111111100 \\ 011111111111111111110 \\ 000000000000000000000 \end{vmatrix}$$

$$A=\begin{vmatrix}
0000000000 & 0000000000\\
0000000000 & 1000000000\\
0000000002 & 4300000000\\
0000000024 & 4430000000\\
0000000244 & 4443000000\\
0000002444 & 4444300000\\
0000024444 & 4444430000\\
0000244444 & 4444443000\\
0002444444 & 4444444300\\
0024444444 & 4444444430\\
0000000000 & 0000000000
\end{vmatrix},\qquad
B=\begin{vmatrix}
0000000000 & 0000000000\\
0000000000 & 2000000000\\
0000000002 & 4200000000\\
0000000024 & 4420000000\\
0000000244 & 4442000000\\
0000002444 & 4442000000\\
0000024444 & 4442000000\\
0000244444 & 4442000000\\
0002444444 & 4442000000\\
0024444444 & 4442000000\\
0133333333 & 3333333310\\
0000000000 & 0000000000
\end{vmatrix}$$

Рис. 4.

Введением этих матриц каждому элементу f_{ij} ставится в соответствие пара a_{ij}, b_{ij}, которая дает информацию о значениях четырех смежных соседей этого элемента.

После построения матриц A и B, как граничные интерпретируются те элементы, для которых $a_{ij}\neq 4$ или $b_{ij}\neq 4$ (см. рис. 6).

4.2. *Выделение остова.* Работа каждого из четырех шагов одной итерации этого алгоритма реализуется одной подпрограммой.

а). В матрице A значение «2» соответствует левым граничным элементам (см. рис. 4). В работе первой подпрограммы обсуждаются только эти элементы и им присваивается значение «0» если

$$b_{ij}=4 \vee b_{ij}=3 \& f_{i+1,\,j-1}=0 \vee b_{ij}=2 \& f_{i-1,\,j-1}=0$$

б). Вторая подпрограмма удаляет верхние граничные точки($b_{ij}=2$), если для них

$$a_{ij}=4 \vee a_{ij}=3 \& f_{i-1,\,j+1}=0 \vee f_{i-1,\,j-1}=0 \&\ a_{ij}=2$$

в). Третья подпрограмма удаляет правые граничные элементы ($a_{ij}=3$), если

$$b_{ij}=4 \vee b_{ij}=3 \& f_{i+1,\,j+1}=0 \vee a_{ij}=2 \&\ f_{i+1,\,j+1}=0$$

г). Четвертая подпрограмма удаляет нижние граничные точки ($b_{ij}=3$)

$$a_{ij}=4 \vee a_{ij}=3 \&\ f_{i+1,\,j+1}=0 \vee a_{ij}=2 \&\ f_{i+1,\,j-1}=0.$$

С помощью специальных параметров проверяется, изменила ли изображение та или иная подпрограмма. Эти параметры меняют свое начальное нулевое значение, если соответственно подпрограммы а), б), в), г) удаляют хотя бы один элемент $f_{ij}=1$ изображения. Работа считается законченной при условии, что сумма этих параметров равна нулю (см. рис. 7).

4.3. *Выделение замкнутых линий.* В результате работы предыдущего алгоритма получается изображение, для каждого элемента которого в паре a_{ij}, b_{ij} существует хотя бы одна единица, кроме тех точек, в которых пересекается несколько линий. Этот факт учитывается в реализации этого алгоритма. В нем элемент $f_{ij}=1$ получает значение «0», если $a_{ij}=1$ и

$$b_{ij}=3 \&\ f_{i+1,\,j-1}=0 \&\ f_{i+1,\,j+1}=0 \vee b_{ij}=2 \&\ f_{i-1,\,j-1}=0 \&\ f_{i-1,\,j+1}=0,$$

или $b_{ij}=1$ и $a_{ij}=2 \&\ f_{i-1,\,j-1}=0 \&\ f_{i+1,\,j-1}=0 \vee f_{i+1,\,j+1}=0 \& a_{ij}=3 \&\ f_{i+1,\,j+1}=0$

или $a_{ij}=b_{ij}=1$ и $f_{i-1,\,j-1}+f_{i-1,\,j+1}+f_{i+1,\,j-1}+f_{i+1,\,j+1}\leqslant 1$

Здесь тоже есть параметр, который определяет завершение работы (см. рис. 8).

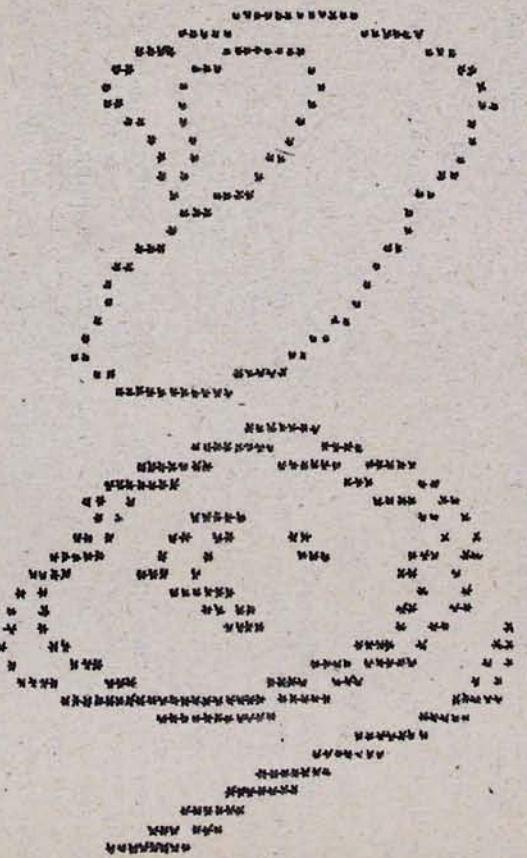


Рис. 5. Исходное изображение

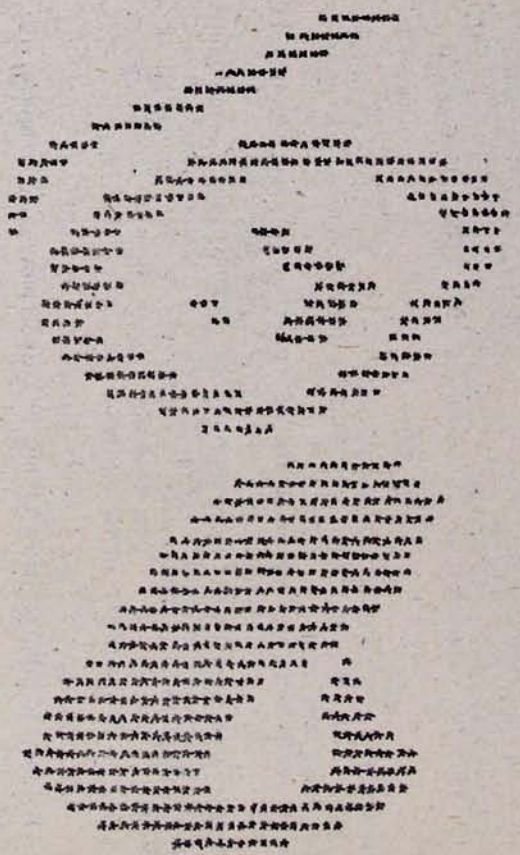
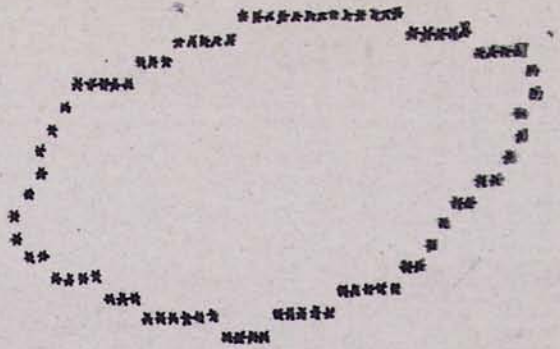


Рис. 6. Выделение границ.



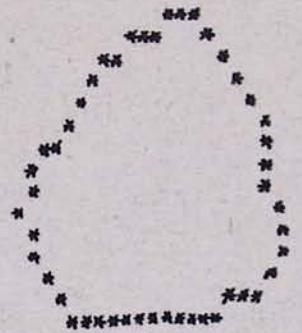
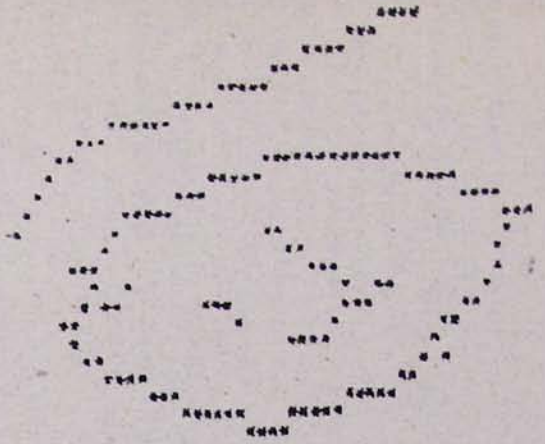


Рис. 7. Замкнутые линии.



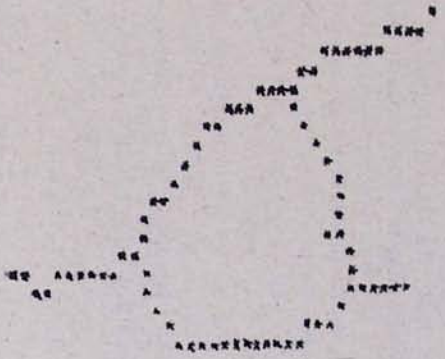


Рис. 8. Осевые линии.

ЛИТЕРАТУРА

1 Автоматический анализ сложных изображений.—М., 1969.
2 *Престон К. и др.* Основы КЛ с приложениями к обработке изображений в медицине. ТИИЭР, т. 67, № 5, 1979, с. 149—185.
3 *Мацелло В. В.* Выделение длинных горизонтальных и вертикальных границ на изображении. В кн.: Математические и технические средства робототехники и распознание образов. АН УССР, ИК, Киев, 1981.
4 *Кийко В. М., Шлезингер М. И.* Алгоритм выделения отрезков осевой линии на графических изображениях. Киев, 1983, (препринт АН УССР, ИК 83—19).
5 *Анишев П. А., Ачасова С. М., Бадман О. Л. и др.* Методы параллельного микропрограммирования. Новосибирск, Наука, 1981.
6 *Сергеев С. Н.* Распознавание непротиворечивости алгоритмов стационарных подстановок. В кн.: Архитектура вычислительных систем с программируемой структурой. Вычислительные системы, вып. 82, Новосибирск, 1980, 18—25.

Ս. Բ. ԱԼԱՎԵՐԴՅԱՆ, Լ. Մ. ՋԱՎԱԽՅԱՆ

ՎԱՆԴԱԿԱՅԻՆ ՏՐԱՄԱԲԱՆՈՒԹՅԱՆ ՄԻՋՈՑՈՎ ԵՐԿՄԱԿԱՐԴԱԿ ՊԱՏԿԵՐՆԵՐՈՒՄ ՍԱՀՄԱՆԱՅԻՆ ԳԾԵՐԻ, ԿՄԱԽՔՆԵՐԻ ԵՎ ՓԱԿ ՏԻՐՈՒՅԹՆԵՐԻ ԱՌԱՆՁՆԱՑՄԱՆ ԱԼԳՈՐԻԹՄՆԵՐ

Ամփոփում

Աշխատանքում առաջարկվում են երկմակարդակ (բինար) պատկերներում սահմանային գծերի, կմախքների և փակ տիրույթների առանձնացման նոր ալգորիթմներ։

Ստացված լոգարիթմները հնարավորություն են տալիս ԷՀՄ-ի միջոցով պատկերների մշակման ժամանակ սահմանային գծերը, կմախքները և փակ տիրույթները առանձնացնող գործողությունները կատարել ղուգահեռ։

Ալգորիթմները հեշտությամբ կարելի է տարածել նաև բազմամակարդակ պատկերների վրա։