

УДК 681.323:681.326.3

Г. К. АФАНАСЬЕВ, В. И. ЛЕБЕДЕВ, А. Ф. ЧЕРНЯВСКИЙ, Г. Н. ЧЕРНИКОВ

АЛГОРИТМ ВЫДЕЛЕНИЯ КОНТУРА ДВОИЧНОГО ИЗОБРАЖЕНИЯ

Обычно автоматическому распознаванию изображений и анализу сцен предшествует предварительная обработка, заключающаяся в кодировании, аппроксимации, фильтрации, восстановлении и улучшении исходных изображений. В каждом конкретном случае выбор необходимых ступеней предобработки определяется качеством исходных изображений и целью их анализа. При решении ряда практических задач оказывается полезным выделение и кодирование контурных линий или границ объектов, составляющих изображения. Предложено и опробовано много методов выделения границ объектов, эвристического характера, основанных на использовании наиболее характерных особенностей границ — больших значениях производных изменения интенсивности в районе границ объектов, а также основанных на формализованных представлениях о границах. Особенности и недостатки этих подходов обобщаются в обзорной статье Девиса [1] в приложении как к параллельным, так и к последовательным методам выделения границ.

Значительно меньше внимания в литературе уделяется разработке алгоритмов выделения контуров, ориентированных на аппаратную реализацию, а также разработке соответствующих вычислительных устройств. В статье [2] предлагаются аппаратно реализуемые алгоритмы кодирования границ объектов (и промежутков между объектами) произвольных черно-белых изображений и их сжатия, но не рассматривается структура соответствующего устройства. В работе [3] описывается проблемно-ориентированный процессор многофункционального назначения, подключаемый к ЭВМ типа ЕС, который, в частности, может выполнять операцию свертки и, следовательно, может быть использован для выделения контуров [4]. Однако этот процессор нецелесообразно применять для решения отдельной задачи обработки изображений, поскольку его многофункциональность не будет использоваться. Известна аппаратная реализация градиентного алгоритма выделения контуров с использованием фильтров Уолша [5], которая обладает рядом недостатков: трудоемкость, настройки фильтров Уолша, низкое быстродействие, сложность генерации функций Уолша.

В данной статье рассматривается алгоритм выделения контура двоичного изображения, ориентированный на аппаратную реализацию средствами цифровой вычислительной техники. При этом объем аппаратуры и время обработки не зависят от структуры изображения.

Обозначим дискретное бинарное изображение как матрицу S размера $m \times n$ из элементов $s_{i,j}$, где $s_{i,j} = \{0, 1\}$. Будем называть элементы $s_{i,j} = 1$ ($s_{i,j} = 0$) черными (белыми) клетками и определим объект изображения как связанное подмножество S_1 множества черных клеток $S^* \subseteq S$,

каждый элемент которого имеет общую сторону или общую вершину хотя бы с одной соседней черной клеткой. Изолированную черную клетку также будем называть объектом. Определим контур объекта как подмножество $K \subseteq S_1$, клетки которого имеют хотя бы одну общую сторону с белой клеткой. Объединение всех подмножеств K образует контур изображения, алгоритм выделения которого (A1) состоит из следующих этапов.

1. Заменить единичные элементы $s_{i,j}$ на нулевые, если $s_{i,j-1}$ и $s_{i,j+1}$ имеют единичные значения. Во всех остальных случаях значения $s_{i,j}$ оставить без изменений.
2. Повторить пункт 1 m раз (по числу строк матрицы S).
3. Транспонировать исходную матрицу S .
4. Повторить пункт 1 для матрицы S^T n раз (по числу строк матрицы S^T).
5. Транспонировать преобразованную по пункту 4 матрицу S^T .
6. Выполнить логическое сложение матриц, полученных по пунктам 2 и 5.

Предлагаемый алгоритм можно обобщить для обработки изображений, разделенных на фрагменты. При этом алгоритм должен учитывать возможное расположение объекта изображения на нескольких фрагментах.

Пусть изображение размера $M \times N$ состоит из $h \times l$ фрагментов, где $h = \left\lceil \frac{M}{m} \right\rceil$, $l = \left\lceil \frac{N}{n} \right\rceil$. Предположим, что фрагменты размера $m \times n$ изображения считываются рецепторным полем размера $(m+2) \times (n+2)$ последовательно по строкам, начиная с верхнего левого и кончая нижним правым. При этом верхняя горизонтальная и левая вертикальная границы рецепторного поля совмещаются с соответствующими границами считываемого фрагмента. Фрагменты обрабатываются в соответствии с описанным выше алгоритмом по мере их считывания. Больший размер рецепторного поля по сравнению с размером фрагмента необходим для «сшивки» фрагментов, при которой достаточно учесть из всех возможных три ситуации:

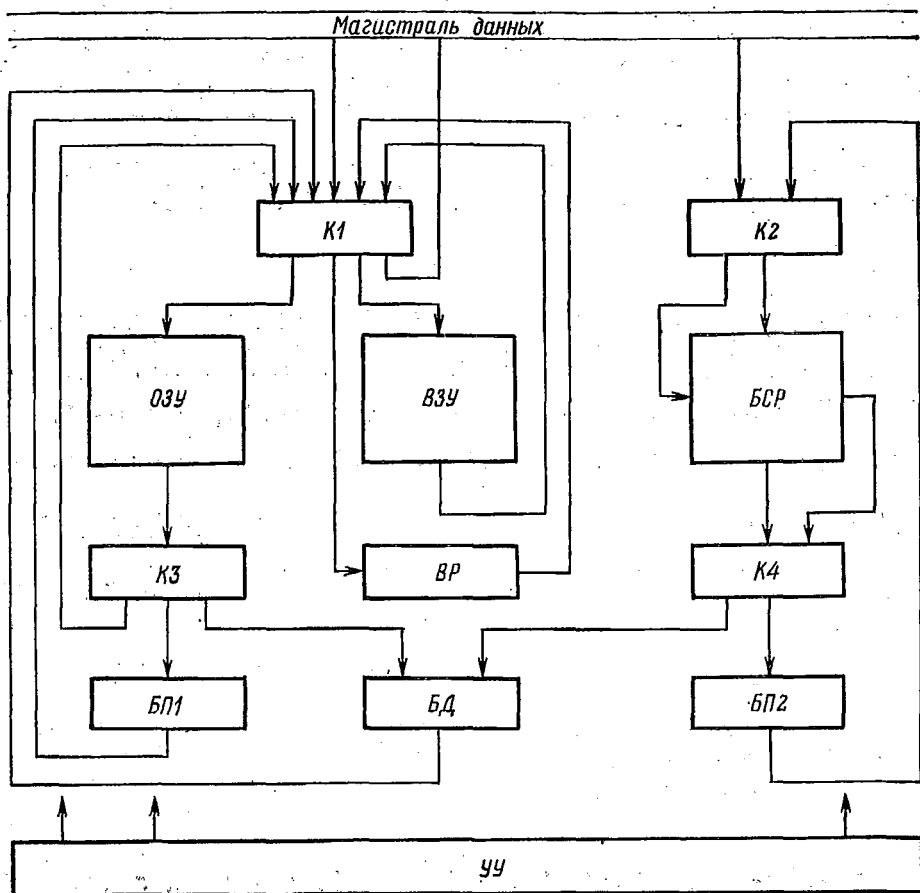
- в анализируемом фрагменте расположена часть объекта, начинающегося и заканчивающегося в смежных фрагментах;
- объект заканчивается в первой строке (столбце) анализируемого фрагмента;
- объект начинается в последней строке (столбце) анализируемого фрагмента.

Обобщенный алгоритм выделения контура (A2) состоит из следующих этапов.

1. Присоединить к текущему фрагменту $\Phi_{i,j}$ две смежные строки фрагмента $\Phi_{i+1,j}$, два смежных столбца фрагмента $\Phi_{i,j+1}$ и четыре смежных с этими строками и столбцами элемента фрагмента $\Phi_{i+1,j+1}$. Назовем полученный по этому пункту фрагмент расширенным.
2. Применить A1 к расширенному фрагменту $\Phi_{i,j}$.
3. Запомнить элементы $(m+1)$ -й строки и $(n+1)$ -го столбца расширенного фрагмента $\Phi_{i,j}$, преобразованные по пункту 2.
4. Заменить элементы первой строки и первого столбца фрагмента $\Phi_{i,j}$ элементами $(m+1)$ -й строки и $(n+1)$ -го столбца расширенных фрагментов $\Phi_{i-1,j}$ и $\Phi_{i,j-1}$ соответственно.
5. Повторить пункты 1—4 $h \times l$ раз (по числу фрагментов изображения).

Фрагмент $\Phi_{i,j}$ размера $m \times n$ после выполнения пунктов 1—4 алгоритма будет содержать контуры объектов, расположенные целиком в этом фрагменте и части контуров объектов, расположенные в нескольких фрагментах, включая фрагмент $\Phi_{i,j}$.

Структурная схема процессора (см. рисунок) для реализации обобщенного алгоритма выделения контура состоит из оперативного запоминающего устройства (ОЗУ) емкостью $m+2$ двоичных $n+2$ -разрядных слов; блока сдвиговых регистров (БСР), содержащего $n+2$ универсальных $m+2$ -разрядных регистров сдвига; двух блоков преобразо-



Пример реализации процессора

вания (БП1) и (БП2) на $n+2$ и $m+2$ разрядов соответственно; блока дизъюнкторов (БД); коммутаторов (К1—К4); вспомогательного запоминающего устройства (ВЗУ) емкостью l двоичных $n+2$ -разрядных слов; вспомогательного $m+2$ -разрядного регистра сдвига (ВР); устройства управления (УУ).

Обработка текущего фрагмента $\Phi_{i,j}$ в процессоре производится за три цикла.

В первом цикле расширенный фрагмент $\Phi_{i,j}$ через К1 и К2 записывается одновременно в ОЗУ и БСР в виде матриц Φ и Φ^T соответственно. При этом новая строка матрицы Φ записывается в младшие разряды регистров БСР. Такт записи автоматически совмещается с тактом сдвига ранее записанной информации вправо на один разряд.

Во втором цикле каждая строка матрицы Φ из ОЗУ поступает через К3 и БП1, а каждый столбец этой матрицы (строка матрицы Φ^T) — из БСР через К4 в БП2, где они преобразуются согласно пунктам 1, 4 алгоритма А1.

Из БП1 через К1 преобразованная строка матрицы Φ записывается в исходную ячейку ОЗУ, а преобразованная строка матрицы Φ^T из БП2 через К2 — в исходный регистр БСР.

В третьем цикле строки и столбцы преобразованных матриц Φ и Φ^T считываются одновременно из ОЗУ и старших разрядов регистров БСР соответственно и через К3 и К4 подаются в БД, где производится их логическое сложение согласно пункту 6 алгоритма А1. Из БД через К1 преобразованная матрица Φ (за исключением 1-й, $(m+1)$ -й, $(m+2)$ -й строк

и 1-го, $(n+1)$ -го, $(n+2)$ -го столбцов) записывается в ОЗУ или выводится из процессора для дальнейшей обработки. При этом первые строка и столбец этой матрицы заменяются $(m+1)$ -й строкой и $(n+1)$ -м столбцом расширенных фрагментов $\Phi_{i-1, j}$ и $\Phi_{i, j-1}$ из ВЗУ и ВР соответственно.

Легко показать, что число тактов, необходимое для обработки одного фрагмента в процессоре, равно $2(m+2) + \max\{m+2, n+2\}$.

ЛИТЕРАТУРА

1. Davis L. S.—Computer Graphics and Image Processing, 1975, v. 4, № 3, p. 248.
2. Morrin T. H.—Computer Graphics and Image Processing, 1976, v. 5, № 2, p. 172.
3. Щиглик В. Д., Налбандян Ж. С.—Докл. АН Арм. ССР, 1976, т. 12, № 2, с. 100.
4. Фу К. Структурные методы в распознавании образов. Пер. с англ.—М., 1977, с. 20.
5. Епифанцев Б. Н., Забишта Ю. П.—Изв. Томского политехн. ин-та, 1975, т. 248, с. 35.

Поступила в редакцию
27.12.77.

Кафедра электронных
математических машин

УДК 861.142.01

А. А. КОЛЯДА

АЛГОРИТМЫ АРИФМЕТИКИ ОБОБЩЕННЫХ СОК

Рассмотрим нормированные обобщенные системы остаточных классов (ОСОК) первого рода [1], в которых все арифметические операции сводятся к выполнению некоторых действий над целыми числами из множества

$$P(r_{\min}, r_{\max}) = \left\{ A \in Z \mid A = \sum_{i=1}^n P_i \tilde{\alpha}_i - r_A P; r_A \in [r_{\min}, r_{\max}] \right\}, \quad (1)$$

$$0 \leq \tilde{\alpha}_i < P_i,$$

где Z — множество целых чисел; $P = \prod_{i=1}^n p_i$; $p_1, p_2, \dots, p_n$ — система попарно взаимно простых модулей; $P_i = \frac{P}{p_i}$, $\tilde{\alpha}_i = |\alpha_i| P_i^{-1} |_{p_i}$, $\alpha_i = |A|_{p_i}$ ($i = 1, 2, \dots, n$); $[r_{\min}, r_{\max}] \supset [0, r_{\max}^n + 1]$, $r_{\max}^n$ — максимальное значение нормированного ранга чисел из $[0, P)$, определяемое соотношением

$$r_{\max}^n = n - 1 - \left[\sum_{i=1}^n \frac{1}{p_i} \right] \leq n - 1. \quad (2)$$

Согласно (1), максимальным и минимальным числами множества $P(r_{\min}, r_{\max})$ являются соответственно:

$$A_{\max} = \left| \sum_{i=1}^n P_i (p_i - 1) \right|_P + (r_{\max}^n - r_{\min}) P; A_{\min} = -r_{\max} P. \quad (3)$$

На практике чаще всего оперируют с симметричными множествами. Будем называть $P(r_{\min}, r_{\max})$ «симметричным», если $A_{\max}$ и $A_{\min}$ находятся в одинаково удаленных от 0 интервалах вида $[jP, (j+1)P)$ ($j = 0, \pm 1, \pm 2, \dots$). Согласно (3), это требование приводит к условию $r_{\max} = r_{\max}^n + 1 - r_{\min}$.

Прежде всего остановимся на важном вопросе об оценке интервального индекса (ИИ) результатов арифметических операций. Требующиеся оценки ИИ можно получить непосредственно из формул для них путем исследования на экстремум соответствующих выражений [2, 3]. Однако в некоторых случаях к более простым решениям приводит другой подход,