

АЛГОРИТМ УДАЛЕНИЯ ПОМЕХ ПРИ ОБРАБОТКЕ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ

В задачах обработки изображений часто возникает необходимость выделения связанных областей (объектов) на изображении для подсчета объектов в поле зрения, измерения их геометрических параметров, структурного анализа формы объектов [1] и т. п. Затем на основании результатов измерений принимаются те или иные решения о выделенных объектах, например, классификация их по площади или выделение объектов, имеющих заданную форму.

В данной статье предлагается алгоритм удаления помех. Задача заключается в том, чтобы измерить габаритные размеры связанных областей на изображении и удалить те области, размеры которых оказались меньше заданного порога.

Укажем несколько известных алгоритмов выделения связанных областей. Здесь и далее будем рассматривать двухградационные (графические) изображения, т. е. такие, элементы которых принимают значения 1 — «черное» и 0 — «белое».

1. Параллельные растровые алгоритмы. Здесь все черные клетки изображения нумеруются в произвольном порядке. Затем на каждой итерации рассматриваются всевозможные квадраты размером 3×3 элемента раstra, центральный элемент которых — черный, и этому элементу присваивается наименьший из номеров черных клеток, входящих в данное окно. Процедура повторяется до тех пор, пока при очередном просмотре ни одна из клеток раstra не изменит своего номера. Недостатком таких алгоритмов является заранее неизвестное (и в основном большое) количество итераций. Эффективно использовать такие алгоритмы можно лишь в специализированных вычислительных устройствах типа матричных процессоров, позволяющих распараллеливать обработку изображения.

2. Построчные растровые алгоритмы. Пусть изображение задано на растре размером $m \times n$. Обозначим $x(i, j)$ ($i=1 \div n, j=1 \div m$) черноту клетки с координатами (i, j) , причем $x(i, j) \in \{1, 0\}$. Обозначим $k(i, j)$ метку, присваиваемую клетке (i, j) , для которой $x(i, j)=1$, причем $k(i, j) \in \{1, \dots, p\}$, где p — количество различных меток. До начала работы алгоритма $k(i, j)=0, i=1 \div n, j=1 \div m; p=0$. Итерация алгоритма состоит из последовательного просмотра всех клеток раstra слева направо и снизу вверх. На каждом шаге рассматривается клетка (i, j) и две соседние с ней клетки $(i-1, j)$ и $(i, j-1)$. При этом возникают следующие ситуации: а) если $x(i, j)=0$ (клетка «белая») — то переход на следующую клетку; б) если $x(i, j)=1$ и $x(i-1, j)=x(i, j-1)=0$, то вводится новая метка $p^*=p+1$, где p — максимальная текущая метка, и $k(i, j) := p^*, p := p^*$, т. е. клетке (i, j) присваивается новая метка p ; в) если $x(i, j)=1$ и $x(i-1, j) \neq x(i, j-1)$, то $k(i, j) := k(i^*, j^*)$, где (i^*, j^*) — координаты той клетки из пары $\{(i-1, j), (i, j-1)\}$, для которой $x(i^*, j^*)=1$, т. е. клетке (i, j) присваивается метка, равная метке соседней черной клетки; г) если $x(i, j)=x(i-1, j)=x(i, j-1)=1$ и $k(i-1, j)=k(i, j-1)=p'$, то $k(i, j) := p'$; д) если $x(i, j)=x(i-1, j)=x(i, j-1)=1$ и $k(i-1, j) \neq k(i, j-1)$, то $k(i, j) := \min \{k(i-1, j), k(i, j-1)\}$

Пусть для определенности $k(i-1, j)=k_1$ а $k(i, j-1)=k_2$ и $k_1 < k_2$. Тогда необходимо просмотреть все клетки (i', j') ; $i'=1, \dots, i$; $j'=1, \dots, j$ и для всех клеток, у которых $k(i', j')=k_2$, заменить метку на k_1 .

Таким образом, данный алгоритм требует многократных просмотров изображения и его трудоемкость пропорциональна N^2 , где N — количество черных точек.

3. Многократных просмотров изображения можно избежать, расплачиваясь за это введением дополнительной памяти — так называемого массива переиндексации M . Первоначально значениями элементов данного массива являются их номера, т. е. $M(i)=i$. В случае возникновения ситуации д) из предыдущего алгоритма, вместо просмотра изображения в ячейку $M(k_2)$ заносится значение k_1 . Это означает, что метки k_2 нужно будет заменить на метки k_1 . После окончания просмотра изображения необходимо проанализировать массив M , исключая из него многократную переиндексацию. Например, если $M(r)=s$ и $M(s)=t$, то нужно $M(r) := t$. Затем метки на черных клетках изображения меняются в соответствии с массивом M .

Недостатком данного алгоритма является введение дополнительной памяти, размер которой заранее не известен и зависит от изображения.

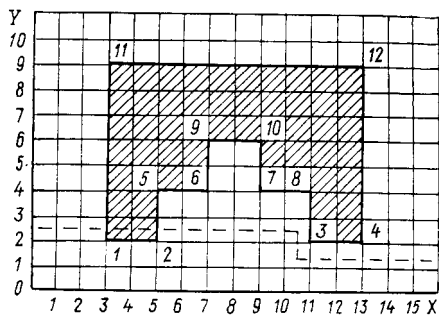
4. Алгоритмы, основанные на выделении контуров. В большинстве случаев более экономным (по сравнению с растровым) описанием изображения является задание его в виде контуров [2]. Контуром называется замкнутая последовательность вертикальных и горизонтальных отрезков, составляющих границу между черными и белыми клетками. Контура задаются при помощи списочной структуры, состоящей из трех синхронных массивов X , Y и A , где $X(i)$ и $Y(i)$ — координаты начала i -го отрезка контура, а $A(i)$ — адрес (номер) начала отрезка, следующего за данным. В этом случае процедура удаления помех будет состоять из трех этапов: выделения контуров, обхода всех контуров для измерения их габаритных размеров, обхода всех контуров для удаления помех.

Недостатком такого алгоритма является наличие массива адресов следующих A , длина которого заранее не известна и равна количеству контурных отрезков на изображении, и необходимость трехкратного просмотра всего изображения.

В данной статье предлагается однопроходный построчный алгоритм удаления помех, основанный на кодировании изображений при помощи так называемых особых мест. Показан ряд преимуществ нового алгоритма по сравнению с известными.

Экономное кодирование. Растровое описание изображений требует огромных затрат машинной памяти. Поэтому чертежно-графические изображения следует экономно кодировать упомянутыми особыми местами. Суть метода состоит в следующем.

Пусть i, j — координаты элемента раstra, а $x(i, j) \in \{1, 0\}$ — чернота этого элемента. Пара (i, j) называется особым местом, если $||x(i, j) - x(i+1, j)| - |x(i, j+1) - x(i+1, j+1)|| = 1$. Так изображение на рисунке кодируется двенадцатью особыми местами. Экономное кодирование изображений заключается в составлении списка особых мест.



Совокупность особых мест является таким же исчерпывающим описанием изображения, как и растровое, однако, для чертежно-графической информации является в десятки и даже сотни раз экономнее.

Описание алгоритма. Предлагаемый алгоритм убирания помех на графических изображениях состоит из двух этапов: 1) выделение связанных областей с измерением их габаритных размеров; 2) удаление выделенных областей, размеры которых меньше заданного порога.

1. Выделение связанных областей. Пусть изображение задано при помощи упорядоченного списка особых мест. Сведения о строке изображения, содержащей особые места, представлены в следующем формате: $Y_i, x_1, \dots, x_k, \langle \text{КС} \rangle$, где Y_i — номер i -й строки, $x_j, j=1, \dots, k$, — координаты x особых мест в строке, расположенные в порядке возрастания, $\langle \text{КС} \rangle$ — признак конца строки. Информация о строках располагается в порядке возрастания номеров строк.

Особое место с координатами (i, j) будем считать расположенными в левом нижнем углу клетки раstra с координатами (i, j) . При кодировании особыми местами предполагается, что чернота клеток $x(1, j), x(n, j) (j=1, \dots, m)$ и $x(i, 1), x(i, m) (i=1, \dots, n)$ равна 0, т. е. граничные точки поля зрения — белые. Тогда количество особых мест в строке всегда четное.

Алгоритм работает по тактам. На каждом такте рассматривается пара соседних особых мест, расположенных в одной строке, причем первое особое место нечетное, а второе — четное. Пусть на очередном такте рассматривается пара особых мест, первое из которых имеет координаты (x^*, y^*) . Тогда просмотренной частью изображения назовем ту часть клеток раstra, координаты Y которых $y < y^*$ и те клетки строки y^* , у которых $x < x^*$. Граница просмотренной части проходит через середины клеток раstra с координатами $(x_k, y^*), x_k=1, \dots, x^*-1; (x_l, y^*-1), x_l=x^*-1, \dots, N$, где N — длина строки изображения. На рисунке пунктирной линией обозначена граница просмотренной части изображения после обработки особых мест 1 и 2.

Алгоритм заключается в заполнении пяти массивов, каждый из которых имеет длину N , равную длине строки изображения. Эти массивы имеют следующий формат.

Индексный массив $I(r)$, $r=1, \dots, N$. Элементы этого массива принимают значения из множества $\{1, 0\}$. Причем $I(r)=1$, если на данном такте вертикальный отрезок контура с координатой $x=r$ пересекает границу просмотренной части. Так после обработки особых мест 1 и 2 (см. рис.), $I(3)=1$ и $I(5)=1$.

Массив связности XH . Пусть $I(r_1)=I(r_2)=1$. Элемент массива $XH(r_1)=r_2$ в том случае, если на просмотренной части изображения существует связный кусок контура, один конец которого пересекает границу просмотренной части в точке с координатой $x=r_1$, а другой конец — в точке с координатой $x=r_2$. Одновременно $XH(r_2)=r_1$. На том же примере (см. рис.) $XH(3)=5$, $XH(5)=3$.

Массивы габаритных координат $MINX$, $MAXX$, $MINY$. Для каждого r^* , для которого $I(r^*)=1$, в соответствующие ячейки массивов $MINX(r^*)$, $MAXX(r^*)$, $MINY(r^*)$ заносятся максимальные и минимальные координаты особых мест, которые принадлежат куску контура, лежащего на просмотренной части и пересекающего границу ее в точке, с координатой $x=r^*$.

Работа алгоритма на каждом такте заключается в модификации пяти описанных выше массивов.

Пусть на очередном такте рассматривается пара особых мест, лежащих в строке с координатой Y^* , левое из которых имеет координату x , равную x_l , а правый — x_p . При этом возможна одна из пяти ситуаций. Опишем работу алгоритма при возникновении каждой из них.

1. $I(x_l)=0$, $I(x_p)=0$. Это значит, что данные особые места не являются концами отрезков контура, лежащих на просмотренной части (это соответствует парам особых мест 1, 2 и 3, 4 на рис.). Такую ситуацию назовем «новый контур». При этом необходимо выполнить следующие действия: $I(x_l):=1$, $I(x_p):=1$; $XH(x_l):=x_p$, $XH(x_p):=x_l$, $MINX(x_l)=MINX(x_p):=x_l$, $MAXX(x_l)=MAXX(x_p):=x_p$, $MINY(x_l)=MINY(x_p):=Y^*$.

2. $I(x_l)=1$, $I(x_p)=0$. Такая ситуация соответствует паре особых мест 5, 6 на рисунке и называется «наращивание вправо». В данной ситуации выполняются следующие действия: $I(x_l):=0$, $I(x_p):=1$, $XH(x_p):=XH(x_l)$, $XH(XH(x_l)):=x_p$, $MAXX(XH(x_p))=MAXX(x_p):=\max\{MAXX(x_l), x_p\}$.

3. $I(x_l)=0$, $I(x_p)=1$. Эта ситуация соответствует паре особых мест 7, 8 на рисунке. Она является симметричной ситуации 2 и называется «наращивание влево». При этом выполняются следующие действия: $I(x_l):=1$, $I(x_p):=0$, $XH(x_l):=XH(x_p)$, $XH(XH(x_p)):=x_l$, $MINX(XH(x_l))=MINX(XH(x_p)):=\min\{MINX(x_p), x_l\}$.

4. $I(x_l)=1$, $I(x_p)=1$, $XH(x_l) \neq x_p$, $XH(x_p) \neq x_l$. Эта ситуация соответствует паре особых мест 9, 10 (см. рис.) и называется «слияние». При этом сливаются два участка контура, находящихся на просмотренной части изображения. В этой ситуации необходимо выполнить следующее: $I(x_l):=0$, $I(x_p):=0$, $XH(XH(x_l)):=XH(x_p)$, $XH(XH(x_p)):=XH(x_l)$. $MINX(XH(x_l))=MINX(XH(x_p)):=\min\{MINX(x_l), MINX(x_p)\}$, $MAXX(XH(x_l))=MAXX(XH(x_p)):=\max\{MAXX(x_l), MAXX(x_p)\}$, $MINY(XH(x_l))=MINY(XH(x_p)):=\min\{MINY(x_l), MINY(x_p)\}$.

5. $I(x_l)=1$, $I(x_p)=1$, $XH(x_l)=x_p$, $XH(x_p)=x_l$. Эта ситуация соответствует паре особых мест 11, 12 (см. рис.) и называется «замыкание». Необходимые действия: $I(x_l):=0$, $I(x_p):=0$.

На этом заканчивается обработка одной связной области на изображении. Причем, координаты ее габаритных точек следующие: $X_{\min}=MINX(x_l)$, $X_{\max}=MAXX(x_l)$, $Y_{\min}=MINY(x_l)$, $Y_{\max}=Y^*$. На основании этих данных принимается решение о том, является ли данная область помехой, либо нет.

2. Удаление помех. После того, как связная область признана помехой, ее нужно удалить, т. е. исключить из описания изображения сведения об особых местах, относящихся к данной области. Для этого воспользуемся техникой выделения контуров.

Контур — это замкнутая последовательность горизонтальных и вертикальных отрезков — границ между черными и белыми клетками раstra. Так как особые места являются концами отрезков контура, то контур можно описать в виде замкнутой последовательности особых мест. Для этого, после выбора направления обхода контура (например, такое, чтобы черные клетки находились справа от линии контура [2]), нужно для каждого особого места указать адрес другого особого места, следующего за данным по контуру. При этом необходима дополнительная память, длина которой равна количеству особых мест в описании изображения.

Заключение. Предложенный алгоритм удаления помех обладает рядом положительных свойств.

1. Алгоритм основан на экономном кодировании изображения при помощи особых мест. Такое кодирование позволяет не только компактно хранить изображение, но и создавать быстродействующие построчные алгоритмы обработки. Трудоемкость данного алгоритма оценивается как kN , где N — количество особых мест на изображении. В случае, если N значительно меньше количества точек раstra, алгоритм работает значительно быстрее, чем известные построчные алгоритмы обработки растровых изображений (см. [3]).

2. Алгоритм выделения связных областей и измерения их параметров является однопроходным. В случае удаления помех требуется еще один обход контуров, признанных помехами. При этом оценка трудоемкости алгоритма kN не меняется.

3. Дополнительная память алгоритма не зависит от количества строк изображения. Требуется 5 массивов длиной в строку плюс ограниченный массив для выделения контуров.

1. Мацелло В. В., Шлезингер М. И. Синтаксический анализ изображений в процессе построчного сканирования // Техн. кибернетика.—1982.—№ 3.—С. 173—179.
2. Тырский С. Я. Алгоритм выделения контура // Матем. и техн. средства робототехники и распознавания образов.— Киев: Ин-т кибернетики АН УССР, 1981.—С. 42—48.
3. Автоматизация анализа и распознавания изображений: Методы и средства.— Рига: Зинатне, 1979.—267 с.

*НТК «Ин-т кибернетики им. В. М. Глушкова»
АН УССР, Киев*

*Получено
05.12.85*

АВТОМАТИКА

НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

ОСНОВАН В 1956 г.

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД

№ 2 • МАРТ
АПРЕЛЬ • 1986

КИЕВ

НАУКОВА ДУМКА

СОДЕРЖАНИЕ

<i>Цыпкин Я. З.</i> Идентификация нестационарных динамических систем	34
Методы обработки информации	
<i>Ивахненко Н. А., Семина Л. П., Чихрадзе Т. А.</i> Модифицированный алгоритм объективной кластеризации данных	10
Идентификация сложных систем	
<i>Данич В. Н., Семесенко М. П.</i> Идентификация процессов обобщенными экспоненциальными квазиполиномами. Часть 1	20
<i>Ивахненко А. Г., Ивахненко Л. П.</i> Самоорганизация математических моделей при создании искусственного интеллекта	25
<i>Михайлов В. И., Сидоров Ю. Е.</i> Методы верификации моделей для идентификации характеристик объекта и прогноза по зашумленным и неполным данным исследований	36
<i>Степашко В. С.</i> Селективные свойства критерия непротиворечивости моделей	40
<i>Бюттнер Т., Мюллер И. А., Козубовский С. Ф.</i> Применение теории самоорганизации для анализа и прогнозирования демографических процессов	49
Динамика систем управления	
<i>Бирюков В. Ф., Воронов Е. М., Карпенко А. П.</i> О применении принципа сложности и гарантирующих решений в задаче программного управления в условиях неопределенности	53
Адаптивное управление	
<i>Васильев В. И., Горелов Ю. И., Усманов З. Д.</i> О возможности построения управления плохо формализуемыми объектами	61
Краткая информация	
<i>Долгополов В. И.</i> Модель динамики магнитного поля активных областей Солнца с определением функции источника по МГУА	68
<i>Зелинский Д. И.</i> О влиянии неустойчивости частотных позиций сигнала на помехоустойчивость частотно-фазовой системы передачи данных	71
<i>Козубовский С. Ф.</i> Определение оптимального множества запаздывающих аргументов для разностной прогнозирующей модели с помощью корреляционного анализа	74