

УДК 681.391.828:681.327.12

Абламейко С.В.

АЛГОРИТМЫ ЛОГИЧЕСКОЙ ФИЛЬТРАЦИИ ДЛЯ ПОДАВЛЕНИЯ ШУМОВ НА БИНАРНОМ ИЗОБРАЖЕНИИ

Приводятся алгоритмы, использующие логические фильтры, что позволяет достаточно быстро и качественно подавить шумы, возникающие на бинарном изображении.

При автоматическом считывании черно-белой графической информации с чертежа или карты часто возникают помехи, которые могут оказывать существенное влияние на структуру и форму элементов изображения, поэтому перед дальнейшей его обработкой и необходимо устранить. Для этой цели используются простые и эффективные методы логической фильтрации [1-4].

Сущность этих методов заключается в том, что элементы изображения обрабатываются логическими фильтрами. Под логическим фильтром понимается некоторая аппаратная окрестность с определенным расположением в ней единичных и нулевых элементов. Наиболее часто используется 8-элементная окрестность. С математической точки зрения логический фильтр - это формула алгебры логики, применение которой к центральному элементу окрестности может изменить его значение или оставить без изменения.

При обработке бинарных изображений в памяти ЭВМ формируется цифровая модель графического документа. На первом этапе обработки должны подавляться шумы, которые могут повлиять на структуру и форму выделенных элементарных объектов изображения. Другими словами, данная операция должна "подготовить" изображение для последующей операции утоньшения и выделения элементарных объектов документа, соответствующих исходным.

Шумы на бинарном изображении возникают, главным образом, в виде неоднородности контура объектов, т. е. всевозможных зазубрин, выбросов, что может сильно повлиять на структуру и форму срединной линии, выделяемой на этапе утоньшения. Другими видами шумов являются изолированные темные пятна небольших размеров, пропуски и реже разрывы или слияние нескольких линий.

Одним из основных логических фильтров, используемых для подавления шумов, является фильтр, меняющий значение центрального элемента в зависимости от количества однотипных с ним элементов в 8-элементной матричной окрестности. Математически он может быть записан в виде

$$x'_0 = f(x_0, \dots, x_8) = x_0 \& \left(\bigwedge_{i=1}^{m+1} \bigwedge_{j=1}^{i+m-1} q_j \right) \vee x_0 \& \left(\bigvee_{i=1}^{m+1} \bigvee_{j=1}^{i+m-1} q_j \right), \quad (1)$$

где x_0, \dots, x_8 - элемент матрицы (рис. 1); m - порог фильтрации; $q_j \in \{x_1, \dots, x_8\}, j=1, 8$.

Алгоритм, осуществляющий последовательный просмотр всего изображения и применение фильтра (1) к каждому элементу, часто называют логическим сглаживанием [1, 2]. Он очищает объекты от зазубрин, выбросов, небольших пропусков, делает их контуры гладкими и удаляет изолированные темные пятна небольших размеров. Однако этот алгоритм обладает следующими недостатками: во-первых, последовательный просмотр всего изображения требует много времени, во-вторых, в тонких линиях изображения происходит потеря связности.

x_1	x_2	x_3
x_8	x_9	x_4
x_7	x_6	x_5

Рис. 1. 8-элементная матрица

Изображение, являющееся растровым представлением чертежа или карты, обычно состоит из конечного числа одиночных линий. Анализ качества такого изображения показал, что шумы, как правило, возникают только на его объектах (при хорошем качестве считываемого устройства). Поэтому нет необходимости применять фильтр (1) ко всем элементам изображения.

С учетом этого рассмотрим два алгоритма логической фильтрации, позволяющие устранить указанные выше недостатки.

Алгоритм фильтрации контурных шумов объекта осуществляет последовательный обход объекта изображения по контуру и применение фильтра (1) к каждому контурному элементу. Этот алгоритм используется только для обработки контуров объектов, т. е. для подавления шумов, возникающих из-за зазубрин, выбросов и т. п. (При хорошем качестве считывающего устройства и считываемого документа шумы на полученном изображении существуют в основном только на контурах объектов.) Алгоритм фильтрации контурных шумов является более быстрым, чем алгоритм логического сглаживания. Различие в скорости зависит от ширины линий объекта; при этом чем больше последняя, тем больше различие. Для более четкого описания алгоритма приведем его запись на языке АЛГОЛ:

```

begin integer m,n,i,j,l; integer array a[1:m,1:n]
      i9:=i;j9=j
M1: j1:=j-1; if a[i,j1]:=∅ then go to M2
      j:=j-1; filtr(i,j);
      if i:=i9^j:=j9 then go to fin else go to M7
M2: j1:=j-1;i1:=i+1; if a[i1,j1]:=∅ then go to M3
      j:=j-1;i:=i+1; filtr(i,j);
      if i:=i9^j:=j9 then go to fin else go to M8
M3: i1:=i+1; if a[i1,j1]:=∅ then go to M4
      i:=i+1; filtr(i,j);
      if i:=i9^j:=j9 then go to fin else go to M1
comment процедура filtr пропускает каждый элемент
изображения через фильтр (1).
M4: i1:=i+1;j1:=j+1; if a[i1,j1]:=∅ then go to M5
      i:=i+1;j:=j+1; filtr(i,j);
      if i:=i9^j:=j9 then go to fin else go to M2
M5: j1:=j+1; if a[i,j1]:=∅ then go to M6
      j:=j+1; filtr(i,j);
      if i:=i9^j:=j9 then go to fin else go to M3
M6: i1:=i-1;j1:=j+1; if a[i1,j1]:=∅ then go to M7
      i:=i-1;j:=j+1; filtr(i,j);
      if i:=i9^j:=j9 then go to fin else go to M4
M7: i1:=i-1; if a[i1,j]:=∅ then go to M8
      i:=i-1; filtr(i,j);
      if i:=i9^j:=j9 then go to fin else go to M5
M8: i1:=i-1;j1:=j-1; if a[i1,j1]:=∅ then go to M1
      i:=i-1;j:=j-1; filtr(i,j);
      if i:=i9^j:=j9 then go to fin else go to M5
fin.: end

```

На рис. 2 приводятся результаты обработки с помощью данного алгоритма исходного объекта изображения.

Алгоритм подавления шумов на объекте осуществляет последовательный просмотр всех единичных элементов объекта и прилегающих к ним нулевых элементов, а также применение к указанным элементам фильтра (1). Запись этого алгоритма на языке АЛГОЛ имеет следующий вид:

```

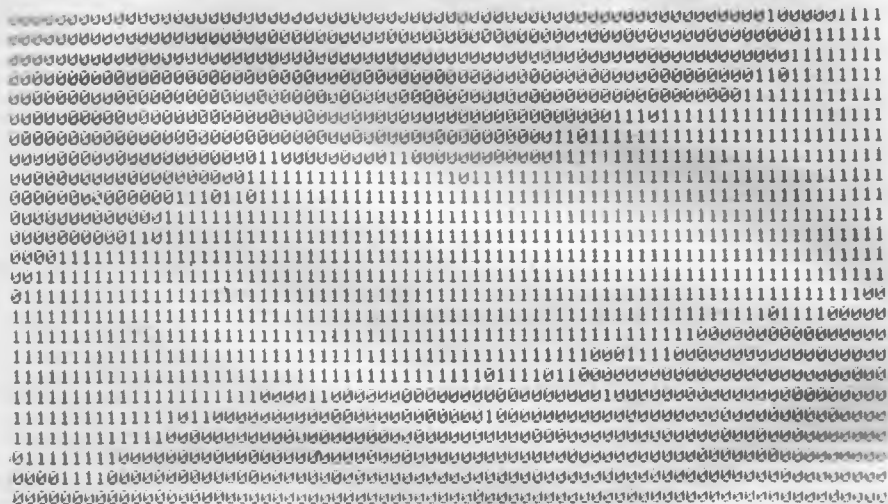
begin integer i,j,i1,j1,i2,j2,i3,m,n,k,k4,l;
  integer a[1:m,1:n],x[1:k];y[1:k];
  k:=1;k4:=1;x[k]=j;y[k]=1;
M8: for i1:=1 step 1 until 3 do
  begin
    for j1:=1 step 1 until 3 do
      begin i2:=i1+1-2;j2:=j1+1-2
        if a[i2,j2]=1 then go to M4; neighb(i,j,1);
        if l<4 then go to M3 else a[i2,j2]:=0;
M4:   for i3:=1 step 1 until k do
          begin if i2:=y[i3]∧j2:=x[i3] then go to M3
                end
          k:=k+1;x[k]:=i2;y[k]:=j2;
M3:   end j1
      end i1
      k4:=k4+1; if k4>k then go to fin
      i:=x[k4];j:=y[k4]; go to M8
      comment процедура neighb подсчитывает количество
      единичных элементов в 8-элементной окрестности;
fin: end

```

На рис. 3, *a*, *b* показаны исходный и обработанный объекты, в которых устранены разрывы между ними шириной в один элемент раstra и удалены изолированные белые пятна. Алгоритм логического сглаживания нарушает связность данного объекта (рис. 3, *b*).

Для обработки автоматически считанной графической информации и преобразования ее в векторную форму разработан пакет программ, реализованный на мини-ЭВМ СМ-4. Обработка ведется по кадрам изображения размером 128x1024. Первым этапом этого процесса преобразования является подавление шумов изображения, для чего используются рассмотренные выше алгоритмы.

Предлагаемые алгоритмы запрограммированы на языке ФОРТ-РАН. Экспериментальная проверка показала их эффективность и хорошее качество полученных результатов.



a



b

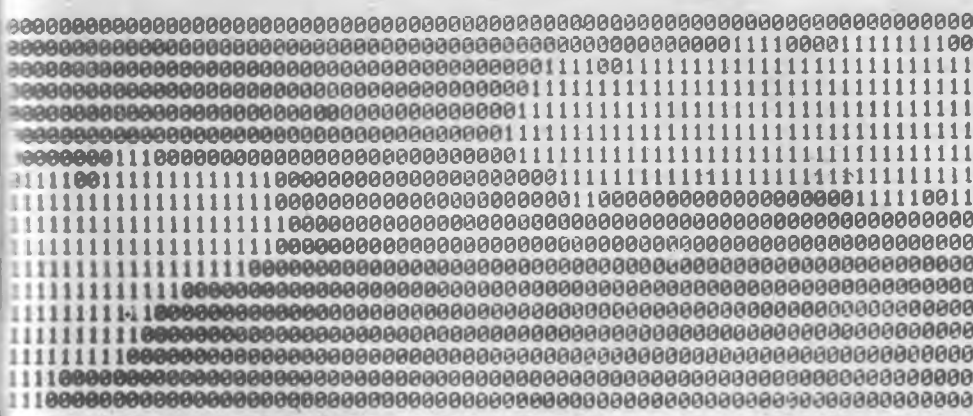
Рис. 2. Результат обработки исходного объекта изображения (а) алгоритмом фильтрации контурных шумов (б)



α



β



γ

Рис. 3. Результаты обработки исходного объекта изображения (α) алгоритмом подавления шумов (β) и алгоритмом логического сглаживания (γ)

Л и т е р а т у р а

1. Тин Д., Прасада Б. Методы цифровой обработки для кодирования графической информации. - ТИИЭР/Пер. с англ., 1979, т. 68, № 7, с. 6-21.
2. Дуда Р., Харт П. Распознавание образов и анализ сцен. - М.: Мир, 1976. - 511 с.
3. Абламейко С.В. Метод выделения средней линии объектов бинарных изображений. - В кн.: Автоматизация процессов проектирования: (Направление: Вычислительная техника в машиностроении; выш. 4). - Минск: ИТК, 1982, с. 128-133.
4. Ковшов В.К., Пчелинцев И.П. Аппарат логического анализа бинарных изображений. - В кн.: Обработка изображений и дистанционные исследования; Тезисы докладов региональной конференции. - Новосибирск, ВЦ СО АН СССР, 1981, с. 29.

Статья поступила в январе 1983 г.