ГОСУДАРСТВЕННЫЙ НАУЧНЫЙ ЦЕНТР РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНСТИТУТ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ

ИФВЭ 97–89
ОМВТ

С.В.Матвеев

ПРЕДСТАВЛЕНИЕ ИЗОБРАЖЕНИЯ В ВИДЕ УКРУПНЕННОЙ БЛОЧНОЙ СТРУКТУРЫ

Протвино 1997

УДК 519.62.622.2:539.18.182 М–24

Аннотация

Матвеев С.В. Представление изображения в виде укрупненной блочной структуры: Препринт ИФВЭ 97–89. – Протвино, 1997. – 9 с., 5 рис., 1 табл., библиогр.: 5.

Предлагается метод обработки полутоновых изображений. Исходное изображение разбивается на последовательность двоичных сечений. К каждому сечению применяется алгоритм сегментации блоков. Форма и размеры блоков определены заранее. Выделение блоков осуществляется на основе анализа “меток”, присваиваемых пикселам изображения. Разработан однопроходный алгоритм присвоения “меток” элементам изображения.

Abstract

Matveyev S.V. Image Representation as Enlarged Blocks Structure: IHEP Preprint 97–89. – Protvino, 1997. – p. 9, figs. 5, tables 1, refs.: 5.

In the present paper we propose a technique for processing a gray-scale image. The original image is presented as sequence of binary slices. A block segmentation algorithm is applied to earch slice. The set of blocks (shape and size) is predefined. The technique of blocks extraction is based on pixel’s “label” examination. One-path algorithm of “labeling” the pixels is developed.

1. Обработка изображений. Основные понятия и алгоритмы

К 2-мерным скалярным данным могут быть отнесены черно-белые (бинарные) или полутоновые фотографии, данные, полученные сканированием материала на определенной глубине (сечения), результаты расчетов и т.п. В физике высоких энергий сюда можно отнести снимки событий в пузырьковых камерах. К задачам, решаемым при обработке изображений, относятся: выделение областей с определенной плотностью (цветом) на фотоснимках, выделение границ областей с заданными свойствами, упрощение **(simplification)** и хранение данных.

Важным элементом обработки изображений является процедура сегментации **(image segmentation)** исходного изображения. Сегментация означает идентификацию границ изображения. Другими словами — это алгоритмы представления изображения в виде набора неперекрывающихся областей, объединение которых даст исходное изображение.

Пусть оригинальное полутоновое изображение состоит из $M \times N$ элементов (пикселов), цвет каждого из которых равен $f(x, y)$. Как правило, для большинства практических задач достаточно 256 различных цветов. На графических станциях этот режим реализован с использованием индексов, ссылающихся на элементы палитры, которая содержит описание цвета, соответствующего данному индексу. Такие изображения удобно описывать в палитре, состоящей из оттенков серого цвета (когда изменению индекса от $T_{min} = 0$ до $T_{max} = 255$ соответствует плавное изменение цвета от черного до белого). В этом случае $T_{min} \leq f(x, y) \leq T_{max}$ будет обозначать уровень серого цвета в пикселе, находящемся в позиции (x, y). Кроме того, пусть $h(x, y)$ будет обозначать результат применения какого-либо алгоритма к оригинальному изображению. Так, использование пороговой обработки **(thresholding)** может быть описано следующим образом:

$$h_k(x, y) = \begin{cases} 1 & \text{if} \quad T_k \leq f(x, y) < T_{k+1}, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

где $T_{mix} \leq T_k \leq T_{max}$ — величина порога, и k — номер порогового уровня.

Пиксел со значением равным 1 представляет объект, а 0-е значение соответствует фону.

Процедура пороговой обработки превращает оригинальное полутоновое изображение в набор двоичных изображений.

Математическая морфология **(Mathematical Morphology)** — еще один из методов, используемых в обработке изображений. Это набор полезных алгоритмов, обеспечивающих анализ *структуры* изображения. Как нелинейная модель обработки изображений и сигналов она стала эффективным прорывом по сравнению с линейными методами. Теория математической морфологии [1],[2] в последние годы получила широкое развитие. Математическая морфология описывается в терминах теории множеств. Множества в 2-мерном евклидовом пространстве означают области принадлежащие объекту (пикселы равные 1 для двоичных изображений).

Морфологические операции — это хорошо известные методы сегментации и улучшения **(enhancement)** изображений, а также распознавания **(recognition)** объектов. Рассмотрение основных операций представлено в терминах двоичных изображений.

Операция эрозии (erosion) изображения определяется следующим образом:

$$h(x,y) = f(x,y) \ominus m(i,j) = \begin{cases} 1 & \text{if } \forall_{i,j} m(i,j) = 1 \bigwedge f(x+i, y+j) = 1, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

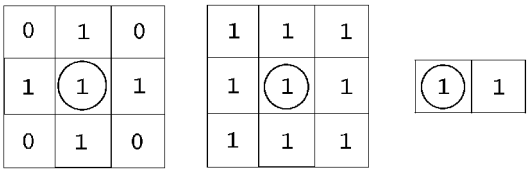

Рис. 1. Маски для морфологических операций.

где 1 — пикселы, принадлежащие объекту; 0 — фон; $m(i,j)$ — маска эрозии **(mask of the erosion)**. Уравнение означает, что, если маска помещена поверх изображения и всем пикселам маски со значением 1 соответствуют пикселы изображения со значением 1, тогда соответствующий пиксел результирующего изображения будет равен 1. В противном случае результат равен 0. Различные типы масок представлены на рис. 1. Здесь пиксел, к которому применяется морфологическая операция, отмечен окружностью.

Операция расширения (dilation) используется для заполнения или расширения области. Операция определяется как

$$f(x,y) \oplus m(i,j) = f(x,y)^c \ominus m(i,j))^c, \tag{3}$$

где $f(x,y)^c$ — обратное или комплиментарное для $f(x,y)$ изображение. Таким образом, операция расширения фона эквивалентна операции эрозии объекта и наоборот. Прямое определение операции расширения:

$$h(x,y) = f(x,y) \oplus m(i,j) = \begin{cases} 1 & \text{if } \forall_{i,j} m(i,j) = 1 \bigvee f(x+i, y+j) = 1. \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Операции эрозии и расширения дуальные, но не инверсные друг по отношению к другу, т.е. выполнение эрозии после расширения и наоборот не восстанавливает исходного изображения.

Операция **закрыть (close)** определяется как операция эрозии, следующая за операцией расширения:

$$h(x,y) = f(x,y) * m(i,j) = (f(x,y) \oplus m(i,j)) \ominus m(i,j). \tag{5}$$

Аналогично, операция **открыть (open)** означает операцию расширения после операции эрозии:

$$h(x,y) = f(x,y)circm(i,j) = (f(x,y) \ominus m(i,j)) \oplus m(i,j). \tag{6}$$

На рис. 2 показано, как морфологические операции влияют на исходное изображение.

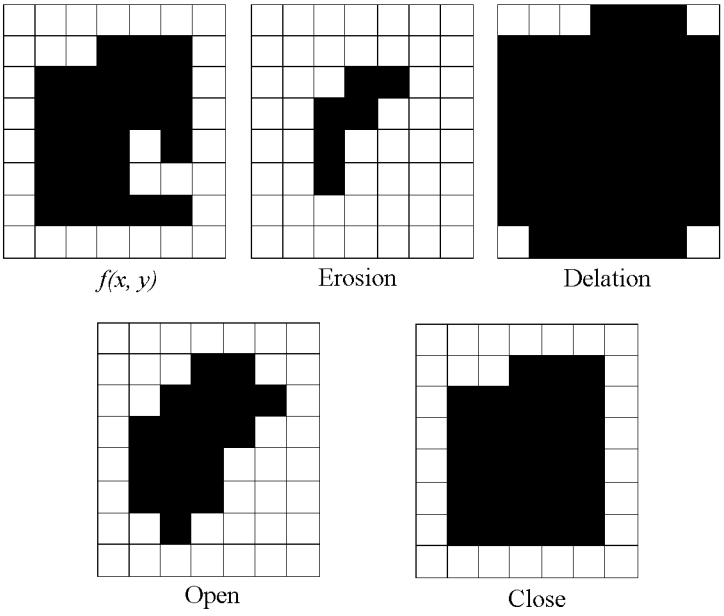

Рис. 2. Пример основных морфологических операций.

Операция **закрыть** заполняет “дыры” и соединяет разрывы объекта. Операция **открыть** используется для устранения мелких деталей.

Выбор и использование методов обработки изображений определяется как конечной целью (конечным результатом), так и ограничениями, которые накладываются на способы работы с данными (изображением).

2. Постановка задачи

Для задачи, решаемой в данной работе, были поставлены следующие цели и ограничения:

- Данные являются полутоновым изображением.
- Результат должен быть представлен в виде, удобном для последующей обработки, т.е. в виде связанных областей заранее определенной формы.
- Каждый элемент данных обладает информацией, необходимой для последующей обработки.

Использование традиционных методов сегментации изображений обычно позволяет выделить области произвольной формы. Однако некоторые операции последующей обработки или анализа изображения будут проходить быстрее, если выделенные области будут иметь определенную форму. Так, в работе [3] описывается однопроходный рекурсивный алгоритм по применению основных морфологических операций к бинарным изображениям, состоящим из *линейных* элементов.

Представление изображения в виде структурных элементов [4] может быть использовано для хранения изображений. Такой способ позволяет существенно ускорить операции увеличения или уменьшения разрешения, с которым изображение выводится на дисплей.

Смит и Чанг [5] предложили алгоритм представления изображений в виде квадрантных деревьев (**quad-tree)**. Структура дерева получается последовательным делением начального узла (изображения) на четыре дочерних. Для каждого из дочерних узлов (фрагментов изображения) эта процедура повторяется до тех пор, пока в узле не будут содержаться элементы только одного типа.

3. Реализация алгоритма

Предлагаемый алгоритм состоит из трех фаз:

I. Сначала исходное изображение подвергается процедуре пороговой обработки. В результате получается набор двоичных сечений, каждое из которых соответствует определенному интервалу полутонов (цветов) (см. формулу (1)).

II. Вторая фаза — алгоритм присвоения пикселам меток, определяемых соседними пикселами.

III. Извлечение блочных областей с заранее определенной формой происходит в третьей фазе.

Фазы II и III применяются последовательно ко всем двоичным сечениям.

```
for  each binary slice
{
    do pixels “labelling” phase (II)
    do blocks segmentation phase (III)
}
```

3.1. Присвоение меток

Фаза присвоения "*меток*" — это однопроходный алгоритм линейного сканирования двоичного изображения. "Метка" пиксела $f(x, y)$ определяется наличием в трех соседних позициях пикселов, принадлежащих объекту (см. рис. 3).

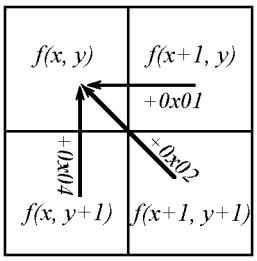

Рис. 3. Присвоение "метки" пикселу.

В терминах теории множеств определим следующую морфологическую операцию:

$$h(x, y) = f(x, y) \sum_{i,j} f(x + i, y + j) m(i, j), \qquad (7)$$

где маска представляется матрицей $m(i, j) = \begin{pmatrix} (1) & 1 \\ 4 & 2 \end{pmatrix}$.

Все возможные случаи присвоения "меток" показаны в таблице.

Таблица

Hex	Bin	Phase I	Phase II
0x00	0000	0 0 0 0	0 0 0 0
0x01	0001	1 0 0 0	1 0 0 0
0x02	0010	1 1 0 0	2 1 0 0
0x03	0011	1 0 0 1	3 0 0 1
0x04	0100	1 1 0 1	4 1 0 1
0x05	0101	1 0 1 0	5 0 1 0
0x06	0110	1 1 1 0	6 1 1 0
0x07	0111	1 0 1 1	7 0 1 1
0x08	1111	1 1 1 1	8 1 1 1

Из данных, приведенных в таблице, видно, что для хранения информации о "метке" достаточно четырех бит. Алгоритм присвоения меток реализован в виде следующей процедуры:

```
algorithm labelling (binary image f)
for each line of binary slice
    {
    for each pixel f(x, y) into current line
        {
         if f(x, y) equal 1 than
           {
           if f(x+1, y) equal 1 than label_1 = 0x01
           else label_1 = 0
           if f(x+1, y+1) equal 1 than label_2 = 0x02
           else label_2 = 0
           if f(x, y+1) equal 1 than  label_3 = 0x04
           else label_3 = 0
           h(x, y)=0x01 + label_1 + label_2 + label_3
           }
    else h(x, y) = 0x00
        }
    }
```

3.2. Тестирование на принадлежность блоку

Используется стандартный набор блоков в соответствии с требованиями постпроцессора. Это может быть набор с ограниченным числом типов (формы) блоков или ограниченным типом и размером. Например, можно использовать блоки в форме квадратов с произвольными или ограниченными размерами, в форме горизонтальных, вертикальных или наклонных линий. Набор блоков может быть любой произвольной формы, которая должна быть описана в процедурах извлечения блоков. Каждому набору блоков соответствует набор функций, которые осуществляют проверку на принадлежность пиксела какому-либо блоку из набора. Тест на принадлежность горизонтальной линии имеет следующий код:

```
function HorLineTest(x, y)
if ( f(x,y) == 0x02 or f(x,y) == 0x04 or f(x,y) == 0x06)
    than return(TRUE)
else return(FALSE)
```

Аналогично определяются другие функции тестирования. Следует заметить, что тестирование пикселов должно осуществляться с проверки на принадлежность сначала к блокам, имеющим более сложную топологию. Если сначала извлечь из изображения все линейные элементы, то ни одного 2-мерного элемента после этого не останется.

3.3. Сегментация блоков

Следующая стадия — это разработка функций извлечения блоков заданной формы и удаления этих блоков из изображения. При удалении пиксела $f(x, y)$ из изображения необходимо изменить "метки" соседних пикселов (см. рис. 4).

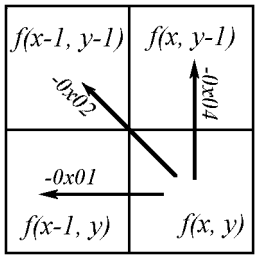

Рис. 4. Изменение "меток" при удалении пиксела.

Следующий код функции удаляет пиксел из изображения.

```
function remove(x, y)
if (f(x-1,y)≠0x00 and x≠0                    than h(x-1,y)     = f(x-1, y)    - 0x01
if (f(x-1, y-1)≠0x00 and x≠0 and y≠0) than h(x-1, y-1) = f(x-1, y-1) - 0x02
if (f(x, y-1)≠0x00 and y≠0)                  than h(x, y-1)    = f(x, y-1)    - 0x04
    h(i, j) = 0x00
```

Внутри этой функции проводится проверка наличия у соседних пикселов метки фона и выхода за границы изображения. Ниже приводится код рекурсивной функции, которая извлекает из изображения горизонтальную линию, удаляет ее и изменяет "метки" соседних пикселов.

```
function HorLineRecursive()
  for all lines from top to down
  {
  for  all pixels f(x, y) from left to right
      {
       if (HorLineTest(x,y) == TRUE )
         {
          length = 1                    \\ a length of a horizontal line
         x_initial = x                  \\initial points have to
         y_initial = y                  \\return to main program
         remove(x, y)
         x = x + 1                      \\ go to the next pixel
         while HorLineTest(x,y) == TRUE
             {
              remove(x,y)
              length =length + 1
              x = x + 1                 \\ go to the next pixel
             }
         length =length + 1             \\ if HorLineTest(x, y) == FALSE
         remove(x+1, y)                 \\ stop recursion on next pixel
         SaveOrPrepareLine(x_initial, y_initial, length)
         }
      }
  }
```

Заключение

Предлагаемый алгоритм позволяет представить исходное изображение в виде "укрупненной" блочной структуры (см. рис. 5). Это важно для тех постпроцессоров, которые обрабатывают блоки за одну операцию. Данный алгоритм был реализован в комплексе программ по созданию 2D-голограмм высокого разрешения и киноrрамм.

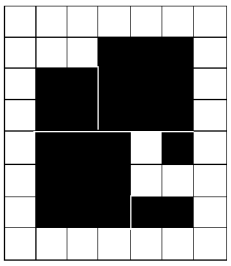

Рис. 5. Пример работы алгоритма (исходное изображение на рис. 2).

Список литературы

[1] Serra J. Image analysis and mathematical morphology. — London, Academic Press, 1982.

[2] Haralick R.M., Stenberg S.R. and Zhuang X. Image Analysis Using Mathematical Morphology. // IEEE-Trans PAMI. July 1987, v. 9, № 4, p. 532-550.

[3] Nadadur Desika C. and Haralick R.M. Recursive Morphology Using Line Structuring Elements. – Proc. of the ISMM'96, Atlanta, May 11-13, 1996.

[4] Ford G.E., Estes R.R. and Chen H. Space Analysis for Image Sampling and Interpolation. – Proc. of the International Conf. on Acoustic, Speech and Signal Processing, 1992, v. III, p. 165-168.

[5] Smith J.R., Chang Shin-Fu. Quad-Tree Segmentation for Texture-Based Image Query. – Proc. of the 2nd Annual ACM Multimedia Conf., San Francisco, CA, Oct. 1994.

Рукопись поступила 25 декабря 1997 г.

С.В.Матвеев.
Представление изображения в виде укрупненной блочной структуры.

Оригинал-макет подготовлен с помощью системы LaTeX.
Редактор Л.Ф.Васильева. Технический редактор Н.В.Орлова.

Подписано к печати 30.12.97. Формат 60 × 84/8. Офсетная печать.
Печ.л. 1,12. Уч.-изд.л. 0,86. Тираж 150. Заказ 131. Индекс 3649.
ЛР №020498 17.04.97.

ГНЦ РФ Институт физики высоких энергий
142284, Протвино Московской обл.

Индекс 3649

П Р Е П Р И Н Т 97–89, И Ф В Э, 1997