

ОБРАБОТКА ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ ПРОЦЕССОРОВ ТАБЛИЧНЫХ ПРЕОБРАЗОВАНИЙ

При аппаратной реализации алгоритмов первичной обработки изображений (АПОИз) целесообразно применение табличных преобразований, позволяющих реализовать широкий класс этих алгоритмов.

При цифровой обработке растровых изображений используется их представление в виде прямоугольной $N \times M$ матрицы чисел - элементов функции яркости изображения $f(m, n)$, где индексы $m \in M$, $n \in N$ - образы физических координат исходного изображения. Преобразование изображений может касаться значений элементов изображения $f(m, n)$ или их координат (индексов). В настоящей статье рассматривается реализация алгоритмов первой группы. Преобразование элементов изображения может выполняться над матрицей в целом (алгоритмы общего вида), группой элементов, попавших в окно перемещающегося по полю изображения (локальная обработка), или над каждым элементом в отдельности (позлементная обработка). Заметим, что функция яркости изображения f принимает конечное число квантованных значений $\text{Im} f \subseteq 2^F$, где 2^F - число уровней квантования, F - небольшое целое число (от 1 до 16). Именно это обстоятельство позволяет применять табличные преобразования для обработки изображений.

Очевидна табличная реализация простейшего вида цифровой обработки - алгоритма поэлементного преобразования изображений. Здесь значение каждого элемента f переводится в новое значение g в соответствии с функциональной зависимостью $f \xrightarrow{\varphi} g$. Функцию преобразования проще всего задать таблицей преобразования, которая должна содержать 2^F элементов (по числу возможных значений функции яркости исходного изображения), а сами значения яркости f можно интерпретировать как адреса таблицы с выходными значениями g . Аппаратная реализация такой таблицы, в отличие от постоянного запоминающего устройства, предлагаемого в работе [1] - оперативное запоминающее устройство (ОЗУ), с организацией $2^F \cdot G$ - разрядных слов ($\{g\} = \text{Im } g = \text{Im}(\varphi \circ f) \subseteq 2^G$). Это объясняется более высоким быстродействием современных ОЗУ, а также тем, что АПОИз часто меняются. Условное графическое обозначение (УГО) ОЗУ таблицы представлено на рис. 1.

В режиме *write* (запись) ОЗУ надо заполнить значениями $\varphi(f)$, где $\varphi \in 2^F$ "пробегают" адресное пространство ОЗУ (множество входов таблицы преобразования) через информационные входы DI_1, DI_2, \dots, DI_G . В режиме *read* (чтение) ОЗУ надо использовать как таблицу преобразования. По этой схеме реализованы устройства, описанные в [2].

Несмотря на свою простоту, метод позволяет в реальном времени решать довольно много прикладных задач улучшения качества и анализа изображений (контрастирование, препарирование, преобразование гистограмм, пороговая обработка). Однако поэлементная обработка не учитывает пространственную структуру изображения и, как следствие, не позволяет решать задачи подчеркивания контуров, сглаживания шумов, низкочастотной или высокочастотной фильтрации и т.д. С другой стороны, оптимальные алгоритмы, решающие названные выше задачи, являются алгоритмами общего вида, весьма сложными в аппаратной реализации (например винеровский фильтр). Поэтому в устройствах обработки изображений реального времени предпочтение отдается квазиоптимальным алгоритмам, основанным на локальной обработке изображений "окном" малых размеров.

В настоящее время популярны алгоритмы локальной обработки, которые вырождаются в суперпозицию (возможно многократную) линейной обработки окном и поэлементного преобразования ($\psi = \varphi_0 \circ \varphi_1 \circ \dots \circ \varphi_S \circ \dots \circ \varphi_R$, где φ_S - алгоритм одного из указанных двух классов). Поскольку поэлементная обработка рассмотрена ранее, рассмотрим линейную обработку "окном". В общем виде оператор преобразования φ имеет вид

$$g(m,n) = \varphi[f,m,n] = \sum_{k,l \in W} d(k,l) \cdot f(m+k, n+l) \quad (1)$$

- f - функция яркости исходного изображения;
- g - функция яркости обработанного изображения;
- $f(m,n), g(m,n)$ - их значения;
- W - область "окна";
- d - линейная маска;
- $a(k,l)$ - значения коэффициентов линейной маски.

Соотношение (1) описывает двумерную свертку функции яркости исходного изображения f с импульсной характеристикой a некоторой двумерной КИХ-системы. Выбирая определенным образом коэффициенты маски $a(k,l)$ и размеры окна, можно выполнить различные операции по улучшению качества изображения: повышение резкости, подчеркивание контуров, сглаживания шумов и т.д.

Для эффективной аппаратной реализации оператора (1) предлагается перейти к свертке поэлементных преобразований элементов окна. Причем, поэлементное преобразование предлагается применить для исключения операции перемножения $a(k,l) \cdot f(m+k, n+l)$ и замены ее операцией: $a(k,l) \cdot f(m+k, n+l) = y_{k,l} [f(m+k, n+l)]$.

Таким образом, процедура локальной обработки принимает вид

$$\varphi[f,m,n] = \sum_{k,l \in W} y_{k,l} [f(m+k, n+l)],$$

что позволяет исключить не только трудоемкую операцию умножения, но и существенно расширить класс решаемых задач.

Если используется прямоугольное окно W с размерами $K \times L$, то маска y содержит $K \times L$ таблиц поэлементных преобразований $y_{k,l}$. Для адресации этих таблиц необходимо $E = \lceil \log_2(K \cdot L) \rceil$ двоичных разрядов. Таким образом, маску поэлементных преобразований можно помещать в ОЗУ с организацией $2^E \times K \times L$ G -разрядных слов, где G, F имеют тот же смысл, что и для поэлементных преобразований (см. рис. 2).

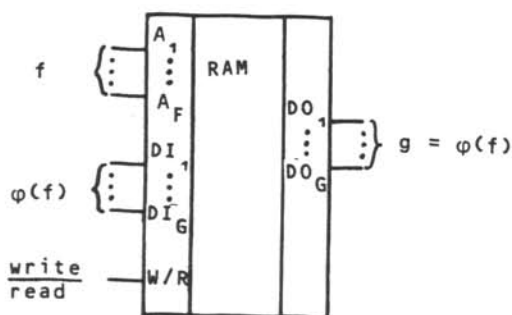


Рис. 1

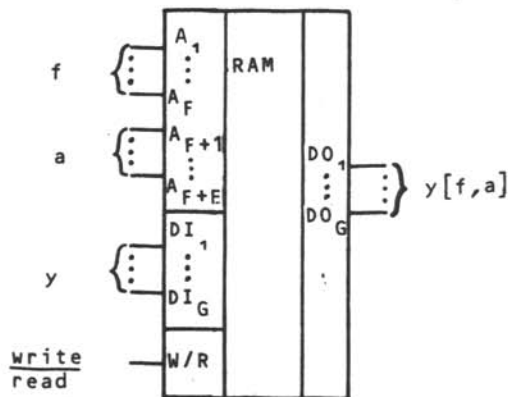


Рис. 2

Часто для преобразования изображений используется совместное поэлементное преобразование двух полей в соответствии с соотношением

$$f_3(m,n) = d \cdot f_1(m,n) \cdot f_2(m,n) + b \cdot f_1(m,n) + c \cdot f_2(m,n);$$

где: $f_1(m,n)$, $f_2(m,n)$ - элементы соответственно первого и второго входных полей;
 $f_3(m,n)$ - результат преобразования (элемент входного поля);
 d, b, c - коэффициенты преобразования.

Используя табличные преобразования, можно эффективно реализовать эту процедуру следующим образом: все произведения вида $const \cdot f(m,n)$ заменить поэлементными преобразованиями

$$f_3(m,n) = g_0[f_1(m,n)] \cdot f_2(m,n) + g_1[f_1(m,n)] + g_2[f_2(m,n)].$$

Такая замена кроме упрощения реализации процедуры значительно расширяет ее возможности. Для умножения, суммирования здесь необходимо использовать матричные перемножитель и сумматор соответственно.

При работе с бинарными изображениями эффективно применение логического фильтра, который представляет собой перемещающееся по изображению "окно" малых размеров. Центральный элемент "окна" заменяется логической функцией отсчетов, попавших в "окно":

$$g(m,n) = \underset{W}{\text{logic}}[f(m,n)] = \text{logic}[f(m+k, n+l): k, l \in W],$$

где:
 $g(m,n)$ - обработанное изображение (бинарное);
 $f(m,n)$ - исходное изображение (бинарное);
 logic - логическая функция по "окну" W .

Для "окна" 3×3 элемента объем памяти, требуемый для записи логической функции от всех элементов, попавших в окно, равен 2^9 бит, для окна 5×5 элементов - $2^{25} = 32$ Мбит. Поэтому "окно" 5×5 не используют. Однако возможно компромиссное решение: повернутое "окно", диагонали которого содержат по пять элементов (рис. 3, где а - нормальное "окно", б - повернутое "окно" 5×5 по диагонали). Требуемый объем памяти для него $2^{13} = 8$ кбит, что позволяет реализовать высокоэффективный логический фильтр на одной БИС ОЗУ (рис. 4).

Таким образом, благодаря табличным преобразованиям становится возможной эффективная реализация широкого класса алгоритмов в реальном времени при использовании простых серийных средств электронной промышленности.

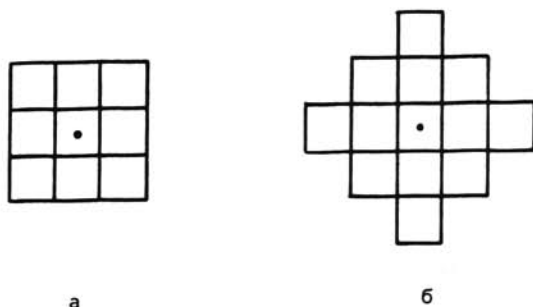


Рис. 3

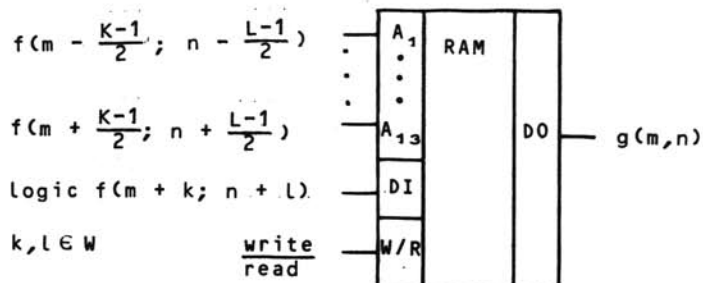


Рис. 4

ЛИТЕРАТУРА

1. Балашов Е.П., Пузанков Д.В. Проектирование информационно-управляющих систем. М.: Радио и связь, 1987. - 255 с.
2. Арефьев Е.Ю., Сергеев В.В., Тихонов Д.Н. Программно-аппаратный комплекс первичной обработки изображений. В кн.: Тезисы докладов II Всесоюзной конференции "Автоматизированные системы обработки изображений (АСОИз-86)", М.: Наука, 1986, с. 238-240.