

РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я

№ 2

1998

В. В. Сергеев

(Самара)

**ПРИМЕНЕНИЕ МЕТОДОЛОГИИ РАСПОЗНАВАНИЯ ОБРАЗОВ
В ЗАДАЧАХ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ***

Предлагается подход к локальной пространственно-инвариантной цифровой обработке изображений, обеспечивающий повышение качественных характеристик, унификацию информационных технологий, возможность создания эффективных программно-аппаратных средств анализа видеoinформации за счет нетрадиционного использования методов теории распознавания образов. Рассматриваются принципы построения и варианты реализации, приводятся некоторые результаты экспериментального исследования процедур обработки изображений, основанных на предложенном подходе.

Введение. Традиционный путь решения задач обработки и анализа изображений включает в себя подбор или полуэвристический синтез большого числа обрабатываемых процедур, что объективно обусловлено разнообразием и сложностью математических моделей формирования оптических сигналов, плохой формализацией решаемых задач, критериев качества обработки и т. д. По этой причине для современных компьютерных систем анализа видеoinформации характерна весьма развешенная и, как следствие, неудобная структура прикладного программного и аппаратного обеспечения, в рамках которой представлена широкая номенклатура алгоритмов обработки изображений, а сами алгоритмы, реализующие лишь отдельные шаги сквозных информационных технологий решения прикладных задач, обладают низкой вычислительной эффективностью и/или не обеспечивают требуемое качество обработки.

С другой стороны, заказчик обработки практически всегда может неформально указать желаемый результат, например, предъявив разработчику согласованную пару изображений, интерпретируемых как «входное» и «выходное». Поскольку сам механизм преобразования данных при этом остается неизвестным, мы имеем здесь типичную для распознавания образов ситуацию необходимости «принятия решения по прецеденту» [1, 2]. Это позволяет применить к обработке изображений универсальную методологию распознавания, т. е. свести синтез обрабатываемого алгоритма к построению решающего правила, а саму задачу обработки — к «узнаванию» результата во входных данных.

Эта идея далеко не нова, возможность подобного подхода вскользь упоминается или лежит в контексте многих работ по распознаванию зрительных образов [1—4]. Однако, насколько известно автору, ее проработка до уровня практической реализации до сих пор не осуществлялась.

* Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант № 96-01-00453).

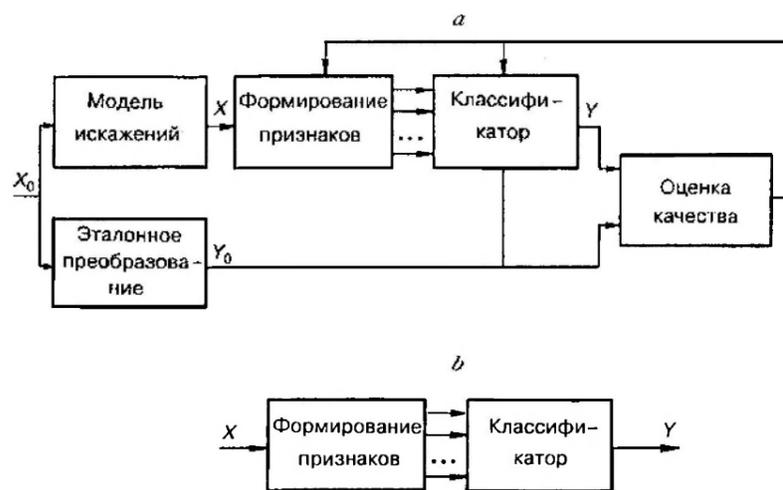


Рис. 1

1. **Общее описание преобразования данных.** Приложение методологии распознавания образов к обработке изображений можно проиллюстрировать несложными схемами, представленными на рис. 1. На входе процедуры обработки имеем изображение X , являющееся результатом искажений «идеального» изображения X_0 , недоступного для наблюдения, в информационном канале системы видеозмерений (рис. 1, *a*). Предполагается, что существует некоторое гипотетическое «эталонное» преобразование идеального изображения к требуемому выходному — Y_0 . Это преобразование может быть как сколь угодно сложным и плохо формализованным (например, при сегментации изображения, устранении мешающих деталей и фона), так и очень простым (при пороговой обработке, выделении контуров) и даже тождественным (при восстановлении изображения, искаженного в канале). В любом случае нам требуется получить результат Y , достаточно близкий к Y_0 , имея доступ только к наблюдаемому изображению X .

Если рассматривать достаточное количество известных нам согласованных пар (X, Y_0) как обучающую выборку, то можно в принципе построить процедуру обучения классификатора, ставящего в соответствие вектору признаков, вычисляемых по входному изображению X , выходное изображение Y . Набор признаков и вид классификатора при необходимости могут корректироваться по результатам оценки качества обработки (т. е. расхождения Y и Y_0), что отражено соответствующими обратными связями на схеме рис. 1, *a*.

Процедуры формирования признаков и классификации, настроенные в итоге обучения на неформально заданное преобразование данных, далее используются при обработке других изображений того же класса (рис. 1, *b*).

Разумеется, в общем случае практическому воплощению представленных схем препятствует чрезвычайно высокая размерность вектора признаков, с которой полностью описывающего входные видеоданные, и астрономическое число классов для выходного изображения. В такой ситуации требуемая обучающая выборка становится совершенно нереальной по объему, а процедура обработки, даже если ее и удастся построить, будет неприемлемо сложной с вычислительной точки зрения. Однако указанные препятствия в значительной степени снимаются, если ограничиться рассмотрением пространственно-инвариантной обработки изображения в режиме «скользящего окна».

2. **Обработка изображений в скользящем окне.** Как известно, ее смысл состоит в том, что некоторая ограниченная двумерная область — «окно обработки» — последовательно (например, в порядке построчной развертки) занимает все возможные положения в плоскости изображения и для каждого положения окна по значениям лежащих в нем входных отсчетов вычисляется значение только одного отсчета выходного изображения.

Пространственно-инвариантная (не зависящая от координат) обработка такого вида описывается общим соотношением:

$$y(n_1, n_2) = G\left[\{x(n_1 - m_1, n_2 - m_2)\}, \quad (m_1, m_2) \in D\right], \quad (1)$$

где $x(n_1, n_2)$, $y(n_1, n_2)$ — двумерные последовательности отсчетов входного и выходного изображения соответственно; G — оператор преобразования; D — конечное множество отсчетов, заданное относительно начала координат и определяющее форму и размеры окна обработки. В рамках предлагаемого подхода преобразование (1) для каждого положения окна реализуется через формирование K -мерного вектора признаков $\{\mu_k(n_1, n_2)\}_{k=0}^{K-1}$ и его использование в процедуре классификации каждого выходного отсчета (рис. 2).

В этом случае рассматриваются только локальные признаки изображения, вычисляемые по окну сравнительно небольшого размера, поэтому их число ограничено. Решение принимается индивидуально для каждого отсчета выходного изображения, и, следовательно, число классов мало (равно числу уровней квантования выходного отсчета). И напротив, объем обучающей выборки достаточно велик, он равен числу положений скользящего окна на изображении, по которому проводится обучение, т. е. числу самих отсчетов изображения. При необходимости выборку еще большего (теоретически неограниченного) объема можно получить с помощью компьютерного моделирования тестовых изображений, их «эталонных» преобразований и искажений в системе видеозамеров.

Первая и основная задача, которую следует решить для реализации предлагаемого подхода, состоит в разработке методов и алгоритмов формирования локальных признаков изображений в скользящем окне. Простейшими признаками могут служить сами значения отсчетов изображения в окне обработки. Однако их использование не всегда целесообразно, поскольку, во-первых, даже для небольшого окна их оказывается слишком много и, во-вторых, такие признаки сильно коррелированы между собой и, как следствие, малоинформативны.

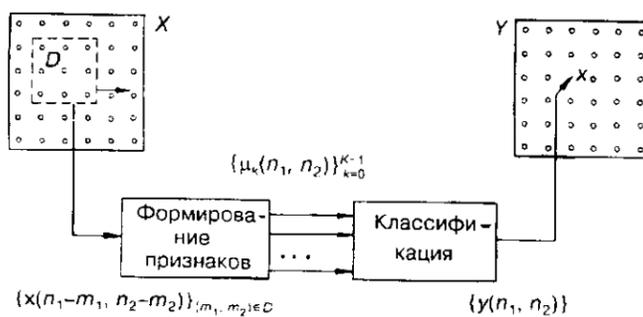


Рис. 2

Существенно более полезными являются локальные «обобщенные моменты» — линейные признаки, вычисляемые с помощью дискретных сверток изображения с конечными ядрами $\{h_k(m_1, m_2)\}_{k=0}^{K-1}$ [5—9]:

$$\mu_k(n_1, n_2) = \sum_{(m_1, m_2) \in D} h_k(m_1, m_2) x(n_1 - m_1, n_2 - m_2). \quad (2)$$

Для уменьшения объема вычислений при получении таких признаков целесообразно взять окно обработки прямоугольной формы

$$D: M_1 \leq m_1 \leq M_1 + N_1 - 1, \quad M_2 \leq m_2 \leq M_2 + N_2 - 1, \quad (3)$$

где величины M_1, M_2 задают положение окна относительно выходного отсчета, а N_1, N_2 — размеры окна, и разделимые ядра:

$$h_k(m_1, m_2) = h_k^{(1)}(m_1) h_k^{(2)}(m_2), \quad 0 \leq k \leq K - 1. \quad (4)$$

При этом формирование обобщенных моментов (2) будет выполняться каскадно, в виде двух одномерных сверток, вычисляемых по координатам n_1 и n_2 :

$$\mu_k(n_1, n_2) = \sum_{m_1=M_1}^{M_1+N_1-1} h_k^{(1)}(m_1) \left[\sum_{m_2=M_2}^{M_2+N_2-1} h_k^{(2)}(m_2) x(n_1 - m_1, n_2 - m_2) \right]. \quad (5)$$

Очевидно, что на основе оценивания локальных статистических характеристик, выделения контуров, ранговой обработки изображения и т. д. [10—12] могут быть сформированы и многочисленные нелинейные признаки изображения в скользящем окне, однако их рассмотрение выходит за рамки данной статьи.

3. Рекурсивное вычисление обобщенных моментов. Особый практический интерес представляют линейные признаки вида (2), формируемые рекурсивно, т. е. не заново для каждого положения окна, а путем внесения поправок в предыдущие значения. Вычислительная сложность соответствующих алгоритмов не зависит от размеров окна обработки, что чрезвычайно важно для случаев использования больших окон.

Считая выполненными условия (3) и (4), для простоты изложения далее будем рассматривать одномерную свертку

$$\mu_k(n) = \sum_{m=M}^{M+N-1} h_k(m) x(n - m), \quad (6)$$

каскадно вычисляемую по двум координатам в соответствии с выражением (5) (здесь и ниже опустим ставшие ненужными «координатные» индексы). Как показано в [5], рекурсивное вычисление этой свертки требует описания входящего в нее ядра следующей общей формулой:

$$h_k(m) = \sum_{l=0}^{L_k-1} S_{lk} m^{\alpha_{lk}} \beta_{lk}^{-m} [u(m - \gamma_{lk}) - u(m - \delta_{lk})], \quad (7)$$

где $L_k, \alpha_{lk}, \gamma_{lk}, \delta_{lk}$ — целочисленные ($L_k \geq 1, \alpha_{lk} \geq 0, M \leq \gamma_{lk} \leq \delta_{lk} \leq M + N$), а S_{lk}, β_{lk} — произвольные вещественные или комплексные константы;

$$u(m) = \begin{cases} 1 & \text{при } m \geq 0, \\ 0 & \text{при } m < 0 \end{cases}$$

— функция единичного скачка. Интерпретируя последовательность $h_k(m)$ как импульсную характеристику «одномерного» линейного цифрового фильтра, через Z -преобразование можно найти его передаточную функцию

$$H_k(z) = \sum_{m=-\infty}^{\infty} h_k(m)z^{-m} = \sum_{i=0}^k S_{ik}(-1)^{\alpha_{ik}} z \underbrace{\frac{d}{dz} \left[z \frac{d}{dz} \left\{ \dots z \frac{d}{dz} \left[\frac{(\beta_{ik}z)^{-\gamma_{ik}} - (\beta_{ik}z)^{-\delta_{ik}}}{1 - (\beta_{ik}z)^{-1}} \right] \dots \right\} \right]}_{\alpha_{ik} \text{ раз}} \quad (8)$$

и далее, представив ее в дробно-рациональной форме (как отношение полиномов от комплексной переменной z), перейти к разностному уравнению с постоянными коэффициентами, описывающему процесс фильтрации сигнала [13].

Варьируя параметры в общей формуле (7), можно получить разнообразные частные семейства ядер, обладающих теми или иными полезными свойствами. Приведем несколько примеров.

3.1. Семейство прямоугольных ядер [5, 6, 11] состоит из функций

$$h_k(m) = u(m - \gamma_k) - u(m - \delta_k) \quad (9)$$

и соответствует случаю, когда в (7) $L_k - S_{0k} = \beta_{0k} = 1$, $\alpha_{0k} = 0$, $\gamma_{0k} = \gamma_k$, $\delta_{0k} = \delta_k$, $M \leq \gamma_k < \delta_k < M + N$.

Передаточная функция (8) записывается в виде

$$H_k(z) = \frac{z^{-\gamma_k} - z^{-\delta_k}}{1 - z^{-1}} \quad (10)$$

Из (10) следует простое разностное уравнение, описывающее процесс рекурсивного вычисления свертки (6)*:

$$\mu_k(n) = \mu_k(n-1) + x(n - \gamma_k) - x(n - \delta_k). \quad (11)$$

Для получения очередного значения последовательности $\mu_k(n)$ по формуле (11) необходимо выполнить всего две арифметические операции: сложение и вычитание. Крайняя простота формирования локальных линейных признаков, сводящаяся здесь к рекурсивному суммированию отсчетов изображения в скользящих прямоугольных окнах, является несомненным, но, возможно, единственным достоинством ядер этого семейства.

3.2. В работах [14, 15] при анализе сигналов на скользящем интервале было предложено использовать семейство ядер косинусного базиса. В [5] показано, что для наиболее быстрого рекурсивного вычисления свертки (6) такие ядра следует брать в нетрадиционной «разнормированной» форме:

$$h_k(m) = \frac{\cos \left[\frac{\pi [2(m-M) + 1] j_k}{2N} \right]}{\cos \left(\frac{\pi}{2N} j_k \right)} \left[u(m-M) - u(m-M-N) \right], \quad (12)$$

где j_k — целочисленный индекс ядра ($0 < j_k < N-1$, $j_p \neq j_q$ при $p \neq q$). Функции (12) получаются из общей формулы (7) при $L_k = 2$, $\alpha_{0k} = \alpha_{ik} = 0$:

$$S_{0k} = \frac{\exp \left[i \frac{\pi (1-2M) j_k}{2N} \right]}{2 \cos \left(\frac{\pi}{2N} j_k \right)}, \quad S_{1k} = \frac{\exp \left[-i \frac{\pi (1-2M) j_k}{2N} \right]}{2 \cos \left(\frac{\pi}{2N} j_k \right)},$$

* Здесь и ниже при записи разностных уравнений не оговариваются начальные условия, т. е. все преобразуемые последовательности считаются не ограниченными по аргументу и принимающими нулевые значения в «минус бесконечности».

$$\beta_{0k} = \exp\left(-i \frac{\pi}{N} j_k\right), \quad \beta_{1k} = \exp\left(i \frac{\pi}{N} j_k\right),$$

$$\gamma_{0k} = \gamma_{1k} = M, \quad \delta_{0k} = \delta_{1k} = M + N.$$

Подставив эти параметры в (8), находим выражение для передаточной функции формирующего цифрового фильтра:

$$H_k(z) = \frac{[z^{-M} - (-1)^{j_k} z^{-M-N}](1 - z^{-1})}{1 - 2\cos\left(\frac{\pi}{N} j_k\right) z^{-1} + z^{-2}}. \quad (13)$$

Как видно из выражения (12), значение индекса $j_k = 0$ порождает прямоугольное ядро, для которого свертка (6) рекурсивно вычисляется с помощью простейшего алгоритма вида (11). Ненулевые значения индекса требуют построения рекурсивного вычислительного процесса, исходя непосредственно из передаточной функции (13). Легко показать, что получение любых $\mu_k(n)$ здесь может быть описано следующей системой разностных уравнений:

$$\begin{cases} x_1(n) = x(n - M) - x(n - M - N), \\ x_2(n) = x(n - M) + x(n - M - N), \\ x_3(n) = x_1(n) - x_1(n - 1), \\ x_4(n) = x_2(n) - x_2(n - 1), \\ \mu_k(n) = \mu_k(n - 1) + x_1(n) & \text{при } j_k = 0, \\ \mu_k(n) = 2\cos\left(\frac{\pi}{N} j_k\right) \mu_k(n - 1) - \mu_k(n - 2) + x_3(n) & \text{при четных } j_k \neq 0, \\ \mu_k(n) = 2\cos\left(\frac{\pi}{N} j_k\right) \mu_k(n - 1) - \mu_k(n - 2) + x_4(n) & \text{при нечетных } j_k. \end{cases} \quad (14)$$

При использовании системы (14) для вычисления K значений одномерных сверток (6) с косинусными ядрами (в том числе и с ядром при $j_k = 0$) число операций сложения (вычитания) и умножения равно соответственно $(2K + 3)$ и $(K - 1)$. Иными словами, при $K \gg 1$ на одном шаге вычислительные затраты в среднем составляют примерно по два сложения и по одному умножению на каждую свертку.

3.3. Интересными свойствами обладает семейство полиномиальных ядер, в котором ядро с номером k является отрезком полинома k -го порядка от целочисленного аргумента m [7—9]:

$$h_k(m) = \sum_{j=0}^k a_{kj} m^j [u(m - M) - u(m - M - N)], \quad a_{kk} \neq 0. \quad (15)$$

Поскольку формула (15) является частным случаем (7), то и здесь применима общая методика построения разностных уравнений, описывающих параллельное рекурсивное вычисление сверток (6), через передаточные функции вида (8). Однако если эти свертки вычисляются не независимо при разных k , а сразу для всех $0 \leq k \leq K$, можно предложить еще более быстрые «каскадно-рекурсивные» процедуры [7, 8]. При произвольных коэффициентах a_{kj} полиномов (15) указанный набор одномерных полиномиальных моментов может формироваться в соответствии с разностными уравнениями

$$\mu_k(n) = \sum_{i=0}^k b_{ki} \mu_i(n - 1) + h_k(M)x(n - M) - h_k(M + N)x(n - M - N), \quad (16)$$

в которых коэффициенты b_{ki} находятся по рекуррентным формулам:

$$b_{kk} = 1, \quad (17)$$

$$b_{ki} = \frac{1}{a_{ii}} \left(\sum_{j=i}^k a_{kj} C_j^i - \sum_{j=i+1}^k b_{kj} a_{ji} \right), \quad i = k-1, k-2, \dots, 0,$$

где C_j^i — биномиальные коэффициенты. Из соотношения (16) видно, что вычисление любого полиномиального момента $\mu_k(n)$ требует вычисления и всех моментов младших порядков. При этом сложность одного шага формирования полного набора моментов при $0 \leq k \leq K$ оказывается довольно высокой, она имеет примерно квадратичную зависимость от K : число сложений (вычитаний) и число умножений равно $\frac{1}{2}(K+1)(K+4)$.

Однако, как показано в [7, 8], существуют полиномы (15), для которых обеспечивается значительное снижение объема вычислений за счет обнуления большей части коэффициентов в разностных уравнениях (16):

$$b_{kk} = b_{k(k-1)} = 1, \quad b_{ki} = 0 \quad \text{при } 0 \leq i \leq k-2. \quad (18)$$

Подставив (18) в (17), несложно получить систему линейных уравнений, которым должны удовлетворять коэффициенты таких полиномов:

$$\sum_{j=i+1}^k C_j^i a_{kj} - a_{(k-1)i} = 0, \quad 0 \leq i \leq k-1, \quad k \geq 1. \quad (19)$$

В этом случае разностные уравнения (16) записываются в упрощенном виде:

$$\begin{aligned} \mu_0(n) &= \mu_0(n-1) + a_{00} [x(n-M) - x(n-M-N)], \\ \mu_k(n) &= \mu_k(n-1) + \mu_{k-1}(n-1) + h_k(M)x(n-M) - \\ &\quad - h_k(M+N)x(n-M-N), \quad k \geq 1, \end{aligned} \quad (20)$$

и число операций сложения и умножения, необходимых для вычисления всех $\mu_k(n)$ при $0 \leq k \leq K$ на одном шаге, равно соответственно $(3K+2)$ и $(2K+1)$, т. е. примерно по три сложения и по два умножения в пересчете на одну свертку (б).

Следует заметить, что при $0 \leq k \leq K$ система (19) состоит из $\frac{1}{2}K(K+1)$ уравнений с $\frac{1}{2}(K+1)(K+2)$ неизвестными, т. е. является недоопределенной и имеет бесконечное множество решений. Иначе говоря, условия (19) задают целый класс полиномиальных ядер с $(K+1)$ степенями свободы, которыми можно воспользоваться, в частности, для дополнительного повышения вычислительной эффективности рекурсивного формирования сверток (б).

Используем одну степень свободы этого класса полиномиальных ядер для того, чтобы соблюдалось условие

$$h_0(n) = a_{00} = 1, \quad (21)$$

а остальные K — для обнуления одного (для определенности последнего) коэффициента во всех разностных уравнениях (20), кроме первого:

$$h_k(M+N) = 0, \quad 1 \leq k \leq K. \quad (22)$$

В данном случае из формул (19), (21) и (22) следует рекуррентная процедура

$$a_{k(i+1)} = \frac{1}{i+1} \left[a_{k(i)} - \sum_{j=i+2}^k C_j^i a_{kj} \right], \quad (23)$$

$$a_{k0} = - \sum_{j=1}^k a_{kj}(M + Ny), \quad i = k - 2, k - 1, \dots, 0, \quad 1 \leq k \leq K.$$

Подставив (21) и (22) в (20), получаем наиболее простой алгоритм вычисления обобщенных моментов с полиномиальными ядрами:

$$\mu_0(n) = \mu_0(n-1) + x(n-M) - x(n-M-N), \quad (24)$$

$$\mu_k(n) = \mu_k(n-1) + \mu_{k-1}(n-1) + h_k(M)x(n-M), \quad 1 \leq k \leq K.$$

Число операций сложения и умножения для всех $\mu_k(n)$ на одном шаге здесь составляет $(2K+2)K$ (в среднем примерно по два сложения и по одному умножению на одну свертку (6)), что эквивалентно соответствующим оценкам, полученным ранее для косинусных ядер.

В экспериментах, результаты которых будут продемонстрированы ниже, использованы ядра прямоугольного и полиномиального типов.

4. Классификация отсчетов изображения. Задачи синтеза и обучения алгоритмов классификации отсчетов выходного изображения по заданному вектору локальных признаков, вычисляемых в скользящем окне, лежат в русле традиционной проблематики теории распознавания образов и в принципе могут иметь стандартные решения [1—3, 16]. Тем не менее предлагаемый подход к обработке видеoinформации требует выполнения классификации в весьма специфичных условиях. Поскольку в его рамках делается попытка построения унифицированного средства обработки изображений, при синтезе классификатора нет оснований использовать важные упрощающие предположения о линейной разделимости классов, гауссовых распределениях признаков и т. п., можно рассчитывать лишь на компактность классов в признаковом пространстве. В такой ситуации наиболее естественным было бы применение какой-либо достаточно хорошей аппроксимации байесовского классификатора [16]. Однако тот факт, что в данном случае классифицируется каждый отсчет выходного изображения, налагает нетипично жесткие ограничения на сложность вычислительных процедур и ведет к необходимости разработки новых быстрых алгоритмов классификации.

Как следует из сути описываемого подхода, номер класса здесь одновременно является значением выходного отсчета, что автоматически определяет понятие расстояния между классами и порождает специфичный вид штрафной функции, получающей смысл погрешности описания отсчетов. При этом оказывается целесообразным аппроксимировать не дискриминантные функции для каждого класса, как это обычно делается в классификаторах, а непосредственно функцию решений.

Вопрос построения алгоритма классификации отсчетов, наилучшим образом приспособленного к данной ситуации и отвечающего сформулированным требованиям, остается открытым. На данном этапе исследований был использован классификатор, реализованный в иерархической (древовидной) форме при простейшей кусочно-постоянной аппроксимации функции решений.

Процесс синтеза (обучения) такого классификатора, а точнее «аппроксиматора» функции решений, заключается в следующем. Область определения указанной функции, представляющая собой K -мерный гиперкуб,

последовательно делением пополам разбивается по осям, что порождает в памяти ЭВМ древовидную структуру. В каждой из областей, полученных в результате разбиения, определяется аппроксимирующее (среднее) значение функции решений. Области с малой ошибкой аппроксимации принимаются за терминальные вершины дерева. Те области, в которых ошибка велика, подвергаются дальнейшему разбиению. Процедура обучения завершается либо при достижении заданной точности аппроксимации, либо при исчерпании ресурсов памяти ЭВМ, выделенных для хранения древовидной структуры вместе со значениями аппроксимирующих констант в терминальных вершинах дерева.

Далее, при обработке изображений, если это необходимо, найденные аппроксимирующие константы функции решений округляются до целочисленных (квантованных) значений выходных отсчетов.

Как показали эксперименты, подобный классификатор при небольшой размерности признакового пространства достаточно быстро обучается и далее демонстрирует хорошие результаты распознавания при малых вычислительных затратах.

5. Результаты экспериментов. Для проверки работоспособности и эффективности предлагаемого метода было проведено его моделирование применительно к различным задачам локальной обработки изображений. Краткое описание рассмотренных примеров и полученных экспериментальных результатов представлено ниже.

5.1. Очевидно, что обычная обработка изображения с целью выделения контуров может быть интерпретирована как классификация его отсчетов на контурные и фоновые. Так, простейший вариант процедуры контурной обработки [10]

$$f(n_1, n_2) = \sqrt{(x(n_1, n_2) - x(n_1 - 1, n_2))^2 + (x(n_1, n_2) - x(n_1, n_2 - 1))^2}, \quad (25)$$

$$y(n_1, n_2) = \begin{cases} 1, & f(n_1, n_2) > f_0, \\ 0, & f(n_1, n_2) \leq f_0, \end{cases}$$

где f_0 — пороговое значение отсчетов «градиентного» изображения $f(n_1, n_2)$, можно представить в виде квадратичного классификатора, использующего в качестве признаков три соседних отсчета входного изображения:

$$\begin{aligned} \mu_0(n_1, n_2) &= x(n_1, n_2), & \mu_1(n_1, n_2) &= x(n_1 - 1, n_2), \\ \mu_2(n_1, n_2) &= x(n_1, n_2 - 1). \end{aligned} \quad (26)$$

На рис. 3, а представлено исходное (незашумленное) тестовое изображение, а на рис. 3, б — результат его обработки в соответствии с формулой (25). Порог f_0 был выбран здесь по гистограмме градиентного поля из априорно принятого условия, что к контурным относится 20 % отсчетов изображения. К сожалению, такая обработка оказывается весьма непомехоустойчивой: даже небольшое добавление шума к исходным данным приводит к резкому снижению ее качества (рис. 3, с, d).

Как альтернативу обычной контурной обработке попробуем использовать предлагаемый метод. При его настройке (обучении) возьмем в качестве эталонного результата «чистые» контуры (см. рис. 3, б), а признаки (26) будем вычислять по зашумленному изображению (см. рис. 3, с). Применение обученного классификатора для обработки последнего изображения дает результат, показанный на рис. 3, е.

Обучающая выборка в данном эксперименте была невелика, она формировалась из единственного кадра размером 256×256 отсчетов, однако результат обучения классификатора оказался уже достаточно устойчивым. Действительно, как видно из рис. 3, f, приемлемое качество обработки здесь сохраняется и для повернутого (т. е., по сути, другого) изображения с новой реализа-



Рис. 3

цией шума. При увеличении объема обучающего материала качественное различие двух последних изображений быстро исчезает.

Этот простой пример, в частности, демонстрирует достаточно высокую скорость и точность аппроксимации функции решений при обучении классификатора в ситуации, когда аргументы этой функции (локальные признаки изображения) заранее известны.

5.2. Рассмотрим теперь другую задачу — *восстановление бинарного (двухградационного) изображения* типа печатного текста. На рис. 4, *a* приведено исходное (неискаженное) изображение, а на рис. 4, *b* — оно же после искажения дефокусировкой и аддитивным белым шумом. В эксперименте сравнивались три метода восстановления изображения:

1) простая пороговая обработка искаженного изображения (ее результат показан на рис. 4, *c*);

2) оптимальная линейная винеровская фильтрация изображения (рис. 4, *d*) с последующей пороговой обработкой (рис. 4, *e*);

3) предлагаемый метод, основанный на использовании семи линейных признаков: центрального отсчета изображения в скользящем окне и двумерных обобщенных моментов с прямоугольными осесимметричными ядрами размерами 1×3 , 3×1 , 3×3 , 3×5 , 5×3 и 5×5 ; обучение метода проводилось по изображениям на рис. 4, *a* и *b*, результат его применения представлен на рис. 4, *f*.

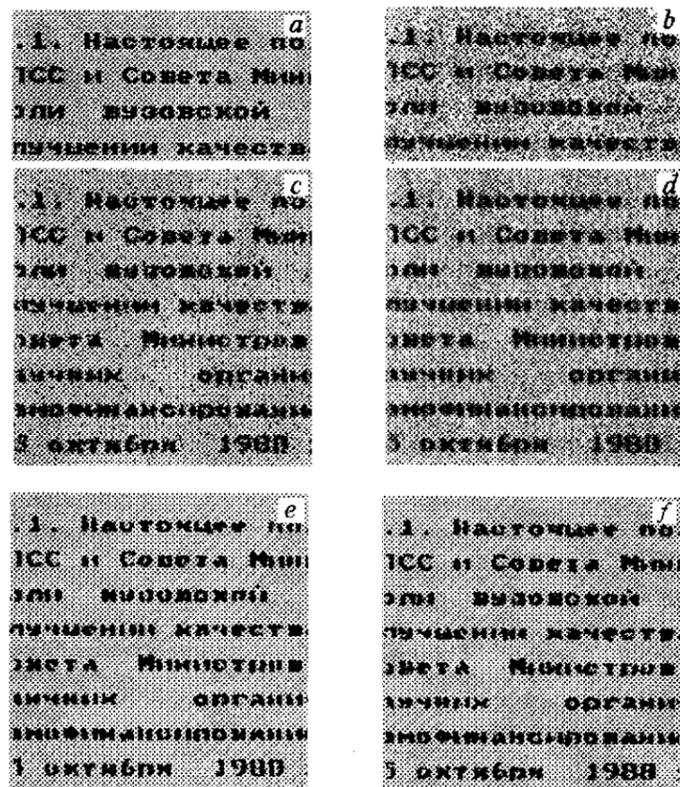


Рис. 4

В табл. 1 приведены количественные оценки эффективности указанных методов — вероятности (относительные частоты) ошибочного восстановления отсчетов в зависимости от коэффициента шум/сигнал по мощности (отношения дисперсий аддитивного шума и неискаженного изображения).

Из этих экспериментальных данных видно, что предложенный метод дает качество восстановления не хуже, чем винеровский фильтр. Причем следует заметить, что этот пример является чисто демонстрационным, здесь (так же, как и в следующем примере) не ставилась задача подбора оптимальной систе-

Таблица 1

Шум/сигнал	Метод 1	Метод 2	Метод 3
0,1	0,033	0,022	0,021
0,2	0,041	0,027	0,026
0,5	0,071	0,036	0,035
1,0	0,124	0,045	0,043

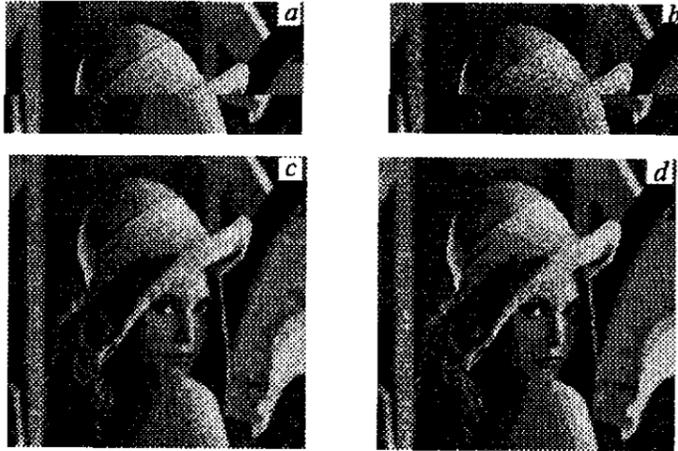


Рис. 5

мы признаков, в результате решения которой эффективность метода, очевидно, может быть дополнительно повышена.

5.3. С подобным же качественным эффектом решается и задача *фильтрации полутонного (многоградационного) изображения*. В отличие от предыдущих случаев здесь число классов (уровней квантования выходных отсчетов) существенно больше двух. В нашем конкретном примере обрабатывалось входное изображение (рис. 5, *a*), квантованное на восемь уровней, по той единственной причине, что пробная версия программы обучения классификатора, использованная в эксперименте, имела жесткие ресурсные ограничения по памяти. На рис. 5, *b* представлено искаженное (зашумленное) изображение, а на рис. 5, *c* и 5, *d* — изображения, обработанные соответственно: 1) методом линейной винеровской фильтрации; 2) предлагаемым методом при четырех линейных признаках: центральном отсчете изображения в окне и двумерных обобщенных моментах с прямоугольными осесимметричными ядрами размерами 3×3 , 5×5 и 7×7 .

В табл. 2 приведены значения относительной остаточной ошибки фильтрации (отношения дисперсий ошибки и входного изображения) в зависимости от коэффициента шум/сигнал по мощности.

Полученные экспериментальные результаты еще раз демонстрируют работоспособность предлагаемого метода и его достаточно высокую эффективность — примерно на уровне винеровского фильтра. Отметим, что

Таблица 2

Шум/сигнал	Метод 1	Метод 2
0,1	0,490	0,453
0,2	0,341	0,346
0,5	0,199	0,218
1,0	0,132	0,143

специальный подбор признаков здесь не производился, а было взято несколько простейших.

5.4. Еще одна задача анализа изображений, имеющая важное прикладное значение, — обнаружение локальных двумерных объектов. Продемонстрируем ее решение на примере обнаружения объекта, показанного на рис. 6, *a*. Объект является достаточно протяженным и вписывается в окно обработки, имеющее размеры намного больше, чем в предыдущих случаях, — 31×31 . Будем считать, что объект может занимать произвольное положение на плоскости и иметь случайную ориентацию. Для обучения классификатора синтезируем тестовое изображение большого числа развернутых объектов, «врезанных» в случайный коррелированный фон, с искажениями типа дефокусировки и шума (рис. 6, *b*).

Известно, что в подобной ситуации в качестве признаков целесообразно использовать геометрические параметры объекта, инвариантные к его поворотам [6, 17]; простейшими из них являются площадь, расстояние до центра тяжести и момент инерции в системе координат, связанной со скользящим окном. Указанные три признака-инварианта рассчитываются по шести локальным двумерным степенным моментам младших порядков: $(0, 0)$, $(1, 0)$, $(0, 1)$, $(2, 0)$, $(1, 1)$ и $(0, 2)$. В свою очередь, этот набор степенных моментов однозначно связан с обобщенными моментами с полиномиальными ядрами тех же порядков [7]. С учетом этой связи сразу возьмем в качестве признаков обобщенные моменты с полиномиальными ядрами указанных порядков. Дополнительным аргументом в пользу такого выбора является наличие алгоритма их рекурсивного вычисления (см. п. 3.3), особенно эффективного для больших скользящих окон.

Одновременно с полями указанных признаков, вычисляемыми по тестовому изображению, при обучении классификатора последнему предъявляется и бинарное изображение, размеченное точками известного положения объектов, — требуемый результат обработки (рис. 6, *c*).

На рис. 6, *d* представлены результаты последующего применения «обученной» процедуры обработки к тестовому изображению, они демонстрируют

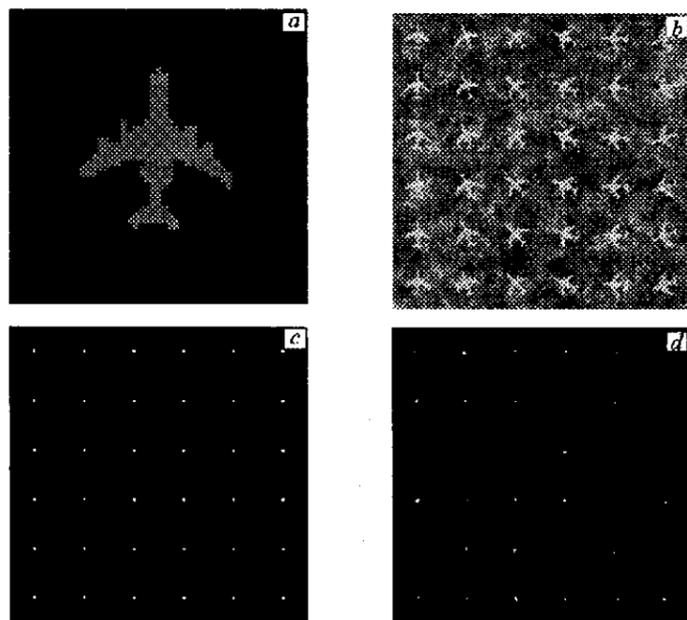


Рис. 6

весьма качественное обнаружение и локализацию (определение координат) объектов.

Заключение. Подводя итог проведенному исследованию, можно сделать вывод, что для широкого круга задач обработки изображений имеется возможность построения унифицированной обрабатывающей процедуры, достаточно легко настраиваемой на конкретное применение. При этом традиционный «пользовательский» подход, основанный на работе с множеством алгоритмов, заменяется на работу с множеством признаков и заранее обученных классификаторов, что в ряде ситуаций может оказаться более удобным. Применение указанной процедуры во многих случаях может дать повышение скорости и/или качества обработки по сравнению с использованием известных «частных» алгоритмов.

Важно отметить и то, что предложенный подход открывает новое поле деятельности специалистам по распознаванию образов «внутри» проблематики цифровой обработки сигналов, требует нетрадиционного использования и, следовательно, переосмысления многих известных результатов теории распознавания.

СПИСОК ЛИТЕРАТУРЫ

1. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов. М.: Наука, 1974.
2. Журавлев Ю. И., Гуревич И. Б. Распознавание образов и распознавание изображений // Распознавание, классификация, прогноз. Математические методы и их применение. М.: Наука, 1989. Вып. 2. С. 5.
3. Ковалевский В. А. Методы оптимальных решений в распознавании изображений. М.: Наука, 1976.
4. Шлезингер М. И., Гимельфарб Г. Л. Возможности современного распознавания образов в прикладном распознавании изображений // Управляющие системы и машины. 1987. № 6. С. 21.
5. Сергеев В. В. Параллельно-рекурсивные КИХ-фильтры для обработки изображений // Компьютерная оптика. 1992. Вып. 10–11. С. 186.
6. Глумов Н. И., Коломиец Э. И., Сергеев В. В. Информационная технология обнаружения объектов на изображении в режиме скользящего окна // Научное приборостроение. 1993. № 1. С. 72.
7. Glumov N. I., Myasnikov V. V., Sergeyev V. V. Polynomial bases for image processing in a sliding window // Pattern Recognition and Image Analysis. 1994. N 4. P. 408.
8. Глумов Н. И., Мясников В. В., Сергеев В. В. Применение полиномиальных базисов для обработки изображений в скользящем окне // Компьютерная оптика. 1995. Вып. 14–15. Ч. I. С. 55.
9. Глумов Н. И., Мясников В. В., Сергеев В. В. Параллельно-рекурсивные алгоритмы вычисления полиномиальных признаков изображения в скользящем окне // Тр. 5-го Междунар. семинара «Распределенная обработка информации». Новосибирск, 1995. С. 272.
10. Прэтт У. К. Цифровая обработка изображений. М.: Мир, 1982. Кн. 2.
11. Ярославский Л. П. Цифровая обработка сигналов в оптике и голографии: Введение в цифровую оптику. М.: Радио и связь, 1987.
12. Gurevitch I. V., Zhuravlev Yu. I., Polikarpova N. S. On image features in recognition problems: P. 1 // Pattern Recognition and Image Analysis. 1995. N 2. P. 204.
13. Оппенгейм А. В., Шафер Р. В. Цифровая обработка сигналов. М.: Мир, 1979.
14. Зеленков А. В. Алгоритмы дискретного косинусного преобразования для спектрального анализа на скользящем интервале // Автометрия. 1982. № 6. С. 34.
15. Виткус Р. Ю., Ярославский Л. П. Адаптивные линейные фильтры для обработки изображений // Адаптивные методы обработки изображений. М.: Наука, 1988.
16. Фукунага К. Введение в статистическую теорию распознавания образов. М.: Наука, 1979.
17. Майтра С. Моментные инварианты // ТИИЭР. 1979. № 4. С. 197.

Поступила в редакцию 10 декабря 1997 г.