

УДК 004.932.1

АЛГОРИТМ РАСПОЗНАВАНИЯ СИМВОЛЬНОЙ ИНФОРМАЦИИ, ОБЛАДАЮЩИЙ ОТНОСИТЕЛЬНОЙ ПРОСТОТОЙ ТЕХНИЧЕСКОЙ РЕАЛИЗАЦИИ

К. т. н. А. В. Садченко, О. А. Кушниренко, М. П. Пындык

Одесский национальный политехнический университет
Украина, г. Одесса
kuuk@mail.ru

Рассмотрен алгоритм распознавания символьной информации без процедур предварительной сегментации и контурной фильтрации, обладающий относительной простотой технической реализации и уменьшающий вычислительную сложность.

Ключевые слова: распознавание символьной информации, бинаризация, коэффициент корреляции, кодовое расстояние.

Задача распознавания символьной информации актуальна в системах машинного зрения. Решение этой задачи необходимо при распознавании автомобильных номеров, надписей на контейнерах и железнодорожных вагонах, анализе текстовых документов, идентификации символьной информации на печатных платах и электронных компонентах и др.

Для решения задачи распознавания символов используют стандартную методику поиска [1] и распознавания на изображениях фрагментов, соответствующих одному из множества шаблонов. Эта задача включает в себя следующие этапы:

- выделение области расположения символов;
- шумоподавление и бинаризация принятого изображения;
- сегментация и определения угла наклона полученного символа;
- сравнение символа с набором шаблонов.

Оптимальный алгоритм сравнения, обеспечивающий наибольшую вероятность правильного различения, заключается в вычислении коэффициента корреляции между распознаваемым символом и набором всех шаблонов.

На этапе сравнения сегментированного символа и шаблона кроме корреляционного можно воспользоваться и другими методами, например, алгоритмом структурных векторов Штарка [2], методом минимума кодовых расстояний или минимума норм и т. д. Однако и для этих методов число операций умножений и сложений будет достаточно большим с учетом необходимости предварительных процедур обработки изображения.

Предлагается снизить вычислительную сложность распознавания изображений с помощью алгоритма приведенного ниже.

Шаг 1. Формирование матрицы C , размерностью $(H \times D)$, соответствующей исходному размеру кадра.

Шаг 2. Преобразование цветного изображения в полутоновое изображение (градации серого).

Шаг 3. Вычисление средней яркости изображения $\bar{\zeta}_B$:

$$\bar{\zeta}_B = \frac{1}{H \cdot D} \sum_{i=1}^H \sum_{j=1}^D C_{ij}, \quad (1)$$

где H — число строк; D — число столбцов.

Шаг 4. Бинаризация матрицы C с порогом бинаризации, соответствующим средней яркости изображения $\bar{\zeta}_B$.

Шаг 5. Формирование матрицы \hat{C} согласно следующему правилу:

- если значение пиксела равно «0» — ему присваивается значение «-1»;
- если значение пиксела равно «1» — он остается без изменений.

Шаг 6. Построение сдвиговой корреляционной матрицы R для каждого шаблона в соответствии с формулой

$$R(l, k) = \sum_{l=1}^{H-n} \sum_{k=1}^{D-m} \left(\sum_{z=1}^n \sum_{s=1}^m Y(z, s) \cdot \hat{C}(l+z-1, k+s-1) \right)^2, \quad (2)$$

где \hat{C} — модифицированная матрица кадра с размерностью $(H \times D)$.

Шаг 7. Сравнение элементов корреляционной матрицы R с порогом υ , где

$$\upsilon = (nm)^2 - d_{cp}^2, \quad (3)$$

$$d_{cp} = \overline{d_{wu}} = \frac{N(N+1)}{2} \sum_{w=1}^{N-1} \sum_{u=w+1}^N d_{wu}, \quad (4)$$

где d_{cp} — усредненное кодовое расстояние между всеми шаблонами символов; N — полный размер (объем) алфавита шаблонов.

Кодовое расстояние между двумя произвольными символами вычисляется по формуле:

$$d_{wu} = \sum_{i=1}^n \sum_{j=1}^m (Y_{ij}^w \oplus Y_{ij}^u), \quad (5)$$

где $w \neq u$; w, u — номера шаблонов; \oplus — сумма по модулю 2 (операция исключающее «или»).

Шаг 8. Вычисление координат символов в составе кадра, соответствующих шаблону.

Сравнение данного алгоритма с корреляционным алгоритмом показывает снижение вычислительной сложности.

Так как коэффициент корреляции вычисляется согласно соотношению [1], вычислительная сложность распознавания одного символа с предварительной сегментацией составит $q_{\pm} = 10(n \cdot m)$, $q_{\times} = 3(n \cdot m)$ количество операций умножений и сложений. При этом вычислительная сложность данного алгоритма составляет $q_{+} = (n \cdot m)$ операций сложения на один символ и $q_{+} = (H \cdot D)$ операций сложения на кадр. Очевидным достоинством данного алгоритма является отсутствие операций умножения.

Рассмотрим пример работы предложенного алгоритма по распознаванию изображения. Для простоты пусть изображение содержит только цифры от 0 до 9.

На рис. 1. представлен вид исходного изображения (кадра).

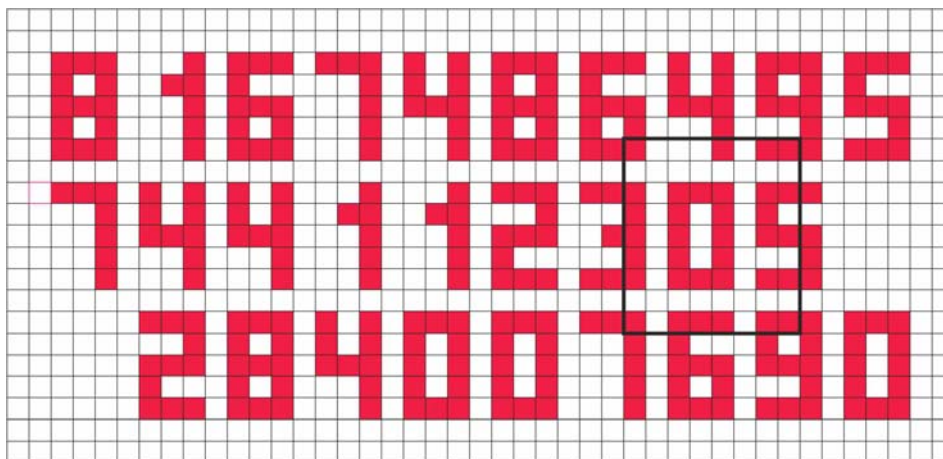


Рис. 1. Исходный кадр

В результате одного прохода алгоритма (для шаблона цифры «0») были вычислены координаты символа «0» в составе кадра (рис. 2). Здесь четыре символа соответствующие шаблону «0», имеют координаты:

1. 15 строка — 19 столбец
2. 15 строка — 23 столбец
3. 15 строка — 39 столбец
4. 9 строка — 31 столбец

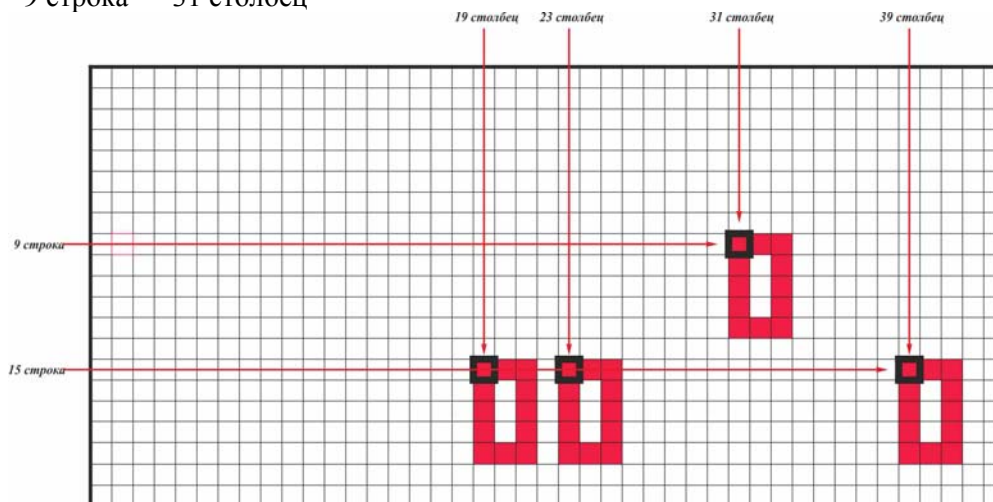


Рис. 2. Вычисленные координаты символа «0» в составе кадра

Таким образом, показано, что основным достоинством данного алгоритма является отсутствие необходимости предварительной сегментации. За один проход алгоритма можно выявить все координаты символов соответствующих данному шаблону. Вычислительная сложность уменьшается из-за отсутствия операций умножения и снижения общего числа операций сложения.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений.— Москва: Техносфера, 2005.
2. Штарк Г. Применение вейвлетов для ЦОС.— Москва: Техносфера, 2007.
3. Глумов Н. И., Мясников Е. В., Копенков В. Н., Чичева М. А.. Метод быстрой корреляции с использованием тернарных шаблонов при распознавании объектов на изображениях // Компьютерная оптика. — 2008.— Т. 32, № 3.

A. V. Sadchenko, O. A. Kushnirenko, M. P. Pindik

A low-complexity algorithm for symbolic information recognition.

The authors consider an algorithm of symbolic information recognition without procedures of preliminary segmentation and contour filtration. The algorithm has a relatively simple technical implementation and reduces the computational complexity.

Keywords: *symbolic information recognition, binarization, correlation coefficient, code distance.*