



УДК 004.852

## ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЯ В ПРОЦЕССЕ РАСПОЗНАВАНИЯ ТЕКСТА

Мищенко В.А., Коробкин А.А.

*Воронежский государственный педагогический университет, Воронеж,  
e-mail: Vadim-liski@mail.ru*

В данной статье рассмотрена задача предварительной подготовки данных для распознавания образов. Здесь будут детально рассмотрены проблемы бинаризации изображения, как первой стадии в подготовке изображения к распознаванию, а также процесс сегментации бинарного изображения с целью выделения отдельных символов для проведения распознавания. Кроме того, проведем сравнительный анализ методов бинаризации глобальной и локальной пороговой обработки изображений и определим наиболее оптимальный для практической реализации алгоритм преобразования изображения. Также будет рассмотрена оптимизация метода Берсена с помощью адаптивной пороговой обработки изображения, позволяющая значительно увеличить скорость работы данного алгоритма.

**Ключевые слова:** распознавание образов, сегментация

## PRELIMINARY PROCESSING OF THE IMAGE DURING RECOGNITION OF THE TEXT

Mishchenko V.A., Korobkin A.A.

*The Voronezh state pedagogical university, Voronezh, e-mail: Vadim-liski@mail.ru*

In given article the problem of preliminary preparation of the data for pattern recognition is considered. Here problems binaryzation images, as first stage in preparation of the image for recognition, and also process of segmentation of the binary image will be in details considered, with the purpose of allocation of separate symbols for carrying out of recognition. Besides we shall lead the comparative analysis of methods binaryzation global and local threshold processing of images and we shall define the optimal for practical realization algorithm of transformation of the image. As optimization of method Bersen with the help of adaptive threshold processing the image will be considered, allowing considerably to increase speed of work of the given algorithm.

**Keywords:** pattern recognition, segmentation

Бинаризация изображения представляет собой процесс преобразования изображения, состоящего из градации одного цвета (в нашем случае – серого), в бинарное изображение, т.е. изображение, в котором каждый пиксель может иметь только два цвета (в нашем случае это черный и белый цвета). В результате такого преобразования цвет пикселя условно считают равным нулю или единице, при этом, пиксели с нулевым значением (в данном случае это пиксели белого цвета) называют задним планом, а пиксели со значением, равным единице (черного цвета), называют передним планом. Но бинарное изображение, полученное в результате такого преобразования, искажается, по сравнению с оригиналом, что характеризуется появлением разрывов и размытостей на объектах, возникновением зашумлений изображения в однородных областях, а также потерей целостности структуры объектов.

Потеря целостности объекта, а так же разрыв объекта возникают в силу ряда причин, таких как большая неравномерность освещения объекта или касание (или наложение объектов друг на друга). Особую сложность в обработке вызывает именно наложение (или касание – как частный случай наложения), т.к., с одной стороны, изображение нескольких объектов может быть интерпретировано как один объект, а с другой стороны, алгоритмы, проверяющие

геометрическую целостность объекта, в местах наложения будут формировать разрывы, представляя эти области в виде заднего плана. Сложность обработки заключается в отсутствии теоретического решения задачи интерпретации наложения объектов, т.к. часть информации утеряна. В реализации алгоритмов на практике в качестве верного решения принимают один из указанных вариантов – либо пересечение считается продолжением текущего объекта, либо область наложения будет считаться задним планом.

В настоящее время существует большое количество методов бинаризации. Суть данного преобразования растровых изображений заключается в сравнительном анализе яркости текущего пикселя  $P(x,y)$  с неким пороговым значением  $P_T(x,y)$ : если яркость текущего пикселя превышает пороговое значение, т.е.  $P(x,y) > P_T(x,y)$ , то цвет пикселя на бинарном изображении будет белым, в противном случае цвет будет черным. Пороговой поверхностью  $P_T$  является матрица, размерность которой соответствует размерности исходного изображения.

Все методы бинаризации разделяют на два вида – это методы глобальной и локальной пороговой обработки. В методах глобальной обработки пороговая поверхность является плоскостью с постоянным порогом яркости, т.е. значение порога рассчитывается исходя из анализа гистограммы всего изобра-

жения и является одинаковым для всех пикселей исходного изображения. Глобальная пороговая обработка имеет существенный недостаток – если исходное изображение имеет неоднородное освещение, области, которые освещены хуже, целиком классифицируются как передний план. В локальных же методах пороговой обработки пороговое значение меняется для каждой точки исходя из некоторых признаков области, принадлежащей некоторой окрестности данной точки. Недостатком такого рода преобразований является низкая скорость работы алгоритмов, связанная с пересчетом пороговых значений для каждой точки изображения.

В качестве метода для решения поставленной задачи воспользуемся методом Бернсена. Метод базируется на идее сопоставления уровня яркости преобразуемого пикселя со значениями локальных средних, вычисляемых в его окружении. Пиксели изображения обрабатываются поочередно путем сравнения их интенсивности со средними значениями яркости в окнах с центрами в точках  $P_l$  ( $l = 0, 1, \dots, 7$ ) (рис. 1).

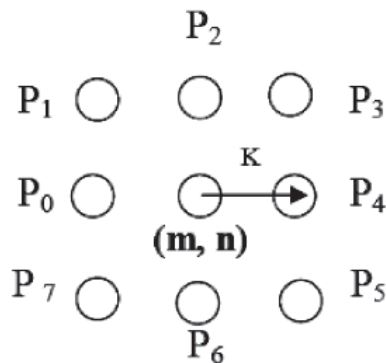


Рис. 1. Преобразование пикселя

Если символ 1 означает элемент объекта, а 0 – элемент фона в результирующем бинарном изображении, то значение преобразованного пикселя  $(m, n)$  становится равным 1 тогда, когда для всех  $l = 0, 1, \dots, 7$  выполняется условие

$$\hat{P}_l - f(m, n) > t,$$

где  $t$  – определенный параметр,

$$\hat{P}_l = \frac{1}{(2K + 1)^2} \sum_{-K \leq i, j \leq K} f(m_l + i, n_l + j) - \text{сред-}$$

няя локальная яркость,  $f(m_p, n_p)$  – яркость в точке  $P_l$  с координатами  $(m_p, n_p)$ .

Автоматическое и адаптивное определение величины локального параметра  $t$  вместо использования глобального значения позволяет устранить ошибки порогового преобразования. Параметр  $t$  вычисляется в соответствии с алгоритмом:

1. В окне  $(2K + 1) \times (K + 1)$  с центром в преобразуемом пикселе  $f(m, n)$  вычисляются значения:

$$f_{\max} = \max_{-K \leq i, j < K} f(m + i, n + j);$$

$$f_{\min} = \min_{-K \leq i, j < K} f(m + i, n + j).$$

2. Вычисляются величины:

$$\Delta f_{\max} = |f_{\max} - \hat{P}|,$$

$$\Delta f_{\min} = |f_{\min} - \hat{P}|.$$

3. Если  $\Delta f_{\max} > \Delta f_{\min}$ , то локальное окно  $(2K + 1) \times (2K + 1)$ , скорее всего, содержит больше локальных низких яркостей, поэтому

$$t = \alpha \left( \frac{2}{3} f_{\min} + \frac{1}{3} \hat{P} \right),$$

где  $\alpha$  – константа из диапазона  $[0,3; \dots; 0,8]$ .

4. Если  $\Delta f_{\max} < \Delta f_{\min}$ , то в локальном окне  $(2K + 1) \times (2K + 1)$  содержится больше локальных высоких яркостей, поэтому

$$t = \alpha \left( \frac{1}{3} f_{\min} + \frac{2}{3} \hat{P} \right).$$

5. Если  $\Delta f_{\max} = \Delta f_{\min}$ , то следует увеличить размер окна до  $(2K + 3) \times (2K + 3)$  и повторить операции, начиная с 1-го шага. Если же и в этом случае  $\Delta f_{\max} = \Delta f_{\min}$ , то пиксель  $f(m, n)$  относится к фону (или же искомым параметр выбирается как  $t = \alpha \hat{P}$ ).

Однако скорость работы реализованного алгоритма оказалась настолько низкой, что она не соответствует требованию интерактивности системы. Поэтому возникла необходимость поиска более быстродействующих решений или улучшений существующего алгоритма, существенно увеличивающих скорость его работы.

В качестве оптимизации данного алгоритма было принято решение о представлении обрабатываемого изображения в виде интегрального. Интегральное изображение может быть использовано при наличии функции  $f(x, y)$ , представляющей некую зависимость между пикселями и действительными числами (к примеру, яркость пикселей), и необходимости вычислить сумму этой функции на некоторых участках изображения.

Чтобы вычислить интегральное изображение, необходимо сохранять для каждого из участков изображения число  $I(x, y)$  – сумму всех значений  $f(x, y)$  для пикселей, расположенных левее и выше пикселя  $(x, y)$ . Для каждого пикселя верна формула:

$$I(x, y) = f(x, y) + I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1).$$

При наличии посчитанного интегрального изображения сумма функции  $f(x, y)$  для любого прямоугольника с верхним левым углом в пикселе  $(x_1, y_1)$  и нижним правым углом в пикселе  $(x_2, y_2)$  может быть за короткое время вычислена с помощью следующего выражения:

$$\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f(x, y) = I(x_2, y_2) - I(x_2, y_1 - 1) - I(x_1 - 1, y_2) + I(x_1 - 1, y_1 - 1) \quad (1)$$

Приведенная техника адаптивной пороговой обработки является простым расширением метода Бернсена, основной идеей которого является сравнение каждого пикселя со средним арифметическим окружающих его пикселей. По мере продвижения по изображению вычисляется аппроксимирующее подвижное среднее последних  $s$  пикселей. Если значение текущего пикселя на  $t$  процентов ниже среднего, то на соответствующем бинарном изображении оно устанавливается как черное, иначе – белое. Проблемой метода Бернсена является то, что он зависит от порядка прохода пикселей. Кроме того, перемещающееся среднее является не слишком хорошей аппроксимацией окружающих пикселей на каждом шаге, потому что соседние пиксели распределены неравномерно по всем направлениям. Используя интегральное изображение, мы получаем такое важное улучшение алгоритма, как значительное сокращение времени работы (так как требуется всего два прохода по всему изображению), а также метод свободен от приведенных недостатков метода Бернсена. Предложенный метод позволяет избежать

резких контрастных линий и игнорирует небольшие градиентные изменения.

На первом проходе по изображению вычисляется интегральное изображение. На втором проходе вычисляем среднее в прямоугольнике размером  $s \times s$ , используя интегральное изображение для каждого пикселя, а затем осуществляется сравнение. Если значение текущего пикселя на  $t$  процентов меньше, чем среднее, оно устанавливается в черный цвет на бинарном изображении, иначе – в белый.

Сегментация изображения является вторым шагом в процессе распознавания знаков почтового индекса. Суть данного процесса сводится к нахождению отдельных символов индекса на всей плоскости изображения. В качестве такого метода сегментации изображения воспользуемся принципом, включающим в себя построение диаграмм Вороного (рис. 2).

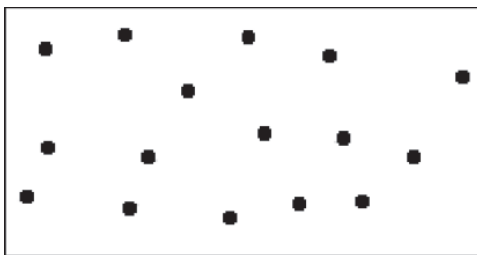
Диаграмма Вороного представляет собой набор конечного множества точек на плоскости, при котором образуется такое разбиение точек на плоскости, при котором каждая область разбиения находится ближе к одному из элементов множества, чем к другому, т.е. если  $S$  – конечное множество точек на плоскости, то диаграмма Вороного  $V(S)$  будет выглядеть, как разбиение множества  $S$  на следующие подмножества:

$$V(s_i) = \{p : \forall j \neq i, d(p, s_i) < d(p, s_j)\}, \quad s_i \in S,$$

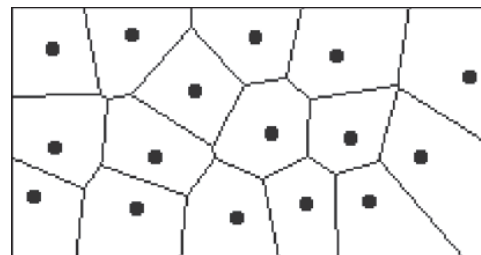
$$V(S) = \bigcup_{i=1}^N V(s_i),$$

где  $d(p, q)$  – евклидова метрика.

Здесь  $V()$  – так называемые ячейки Вороного, а точки  $s_i \in S, i = 1 \dots N$  – генераторы.



а



б

Рис. 2. Множество генераторов и диаграмма Вороного: а – множество генераторов; б – диаграмма Вороного

Как видно из рис. 2, ячейки диаграммы являются выпуклыми многоугольниками, вершины многоугольников называются вершинами диаграммы Вороного, а отрезки, соединяющие вершины, – ребрами диаграммы.

Определим основные свойства, которыми обладает диаграмма Вороного.

1. Каждая вершина диаграммы Вороного, полученной для множества  $N$  генераторов, является точкой пересечения 3-х ребер диаграммы (при  $N > 2$ ).

2. Каждая ближайшая точка  $p \in S$  к точке  $q \in S$ , определенная условием

$$d(p, q) = \min_{r \in S \setminus \{q\}} d(r, q),$$

задает ребро ячейки Вороного, т.е. множество точек  $\{r: d(p, r) = d(q, r)\}$ .

3. Многоугольник  $V(s_i)$  является неограниченным тогда и только тогда, когда точка  $s_i$  лежит на границе выпуклой оболочки множества  $S$ .

4. Диаграмма Вороного, построенная для множества  $N$  точек, имеет не более  $2N - 5$  вершин и  $3N - 6$  ребер.

Воспользуемся принципом диаграммы Вороного для выделения цифр индекса. Для этого обозначим каждый символ как множество связанных точек. Таким образом, значение индекса будет определено как совокупность связанных множеств точек  $C_1, \dots, C_N$ .

В качестве основных параметров для сегментации определяют два типа расстояний. Пусть множество отрезков  $\{l_1, l_2, \dots, l_m\}$  – граница между двумя связанными множествами  $C_i$  и  $C_j$ , тогда расстояние от каждого из компонентов до границы будет определяться формулой:

$$md_s(C_i, C_j) = \min_{1 \leq k \leq m} (g(l_k, C_i), g(l_k, C_j)),$$

где  $g(l_k, \bullet)$  – расстояние от прямой, содержащей  $l_k$ , до соответствующего компонента.

Пусть отрезки  $\{e_1, e_2, \dots, e_{k_i}\}$  – ребра ячейки Вороного, тогда расстояние от компонента  $C_i$  до его границы, заданной ребрами ячейки, будет определено по формуле:

$$md_c(C_i) = \min_{1 \leq k \leq k_i} g(e_k, C_i).$$

Исходя из определенных выше значений, можно сказать, что если

$$md_s(C_i, C_j) \leq 2 \cdot \min(md_c(C_i), md_c(C_j))$$

верно для смежных множеств  $C_i$  и  $C_j$ , то символы, соответствующие этим множествам, объединяются в слово. Если же данное условие не выполняется, то граница ячейки Вороного, разделяющая эти два множества, также является границей слов, которым принадлежат символы, соответствующие множествам  $C_i$  и  $C_j$ . Однако такой метод будет достаточно затратным с вычислительной точки зрения. Для выделения индекса и символов индекса воспользуемся следующим алгоритмом.

Пусть каждому связанному множеству  $S_i$  соответствует

$$c_i = \frac{1}{|S_i|} \sum_{s \in S_i} s, \quad i = 1 \dots N.$$

Множество точек  $C = \{c_i, i = 1 \dots N\}$  будем рассматривать как множество генераторов диаграммы Вороного. Таким образом, получим диаграмму Вороного, построенную по центрам масс символов (рис. 3). Прямолинейные границы ячеек диаграммы Вороного будут достаточно грубым приближением диаграммы, однако, они сохранят информацию о расстояниях между символами и словами.

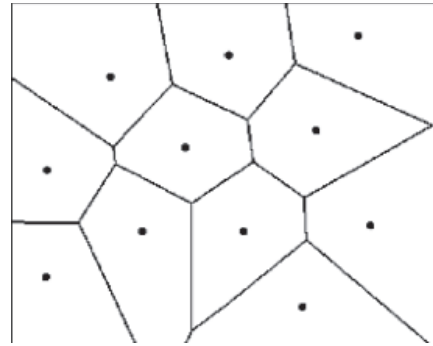


Рис. 3. Диаграмма Вороного, построенная по центрам масс символов

Пусть генератору  $c$  соответствует ячейка, ограниченная множеством прямых отрезков  $E$ . Тогда наименьшее расстояние от генератора до границы ячейки  $V(c)$  будет определено как

$$md_c(c) = \min_{e \in E} d(c, e),$$

где  $d(c, e)$  – евклидово расстояние от точки  $c$  до отрезка  $e$ . Пусть также,  $md_s(c, p)$  – расстояние для двух смежных генераторов до общей границы ячеек Вороного. Данная величина будет равняться половине евклидова расстояния между точками-генераторами.

Исходя из введенных обозначений, можно выделить признаки принадлежности символов, заданных центрами масс  $c = (c_x, c_y)$  и  $p = (p_x, p_y)$  к одному слову и одной строке:

– символы будут находиться в одной строке, если выполнено условие

$$|p_y - c_y| < \alpha \min(md_c(p), md_c(c)),$$

где  $\alpha > 0$  – заранее определенная константа; – для символов находящихся в одном слове будет верно следующее:

$$md_s(p, c) < \beta \min(md_c(p), md_c(c)),$$

где  $\beta > 0$  – заранее определенная константа.

#### Список литературы

1. Чулоков В.А., Астахова И.Ф., Потапов А.С. Основы искусственного интеллекта. – Воронеж: ВГПУ, 2006. – 304 с.
2. Яхнева Г.Э. Нечеткие множества и нейронные сети. – М.: Интернет-Университет информационных технологий; БИНОМ. Лаборатория знаний, 2006. – 316 с.: ил., табл. – (Серия «Основы информационных технологий»).

#### Рецензенты:

Кургалин С.Д., д.ф.-м.н., профессор, декан ФПК, ГОУ ВПО «Воронежский государственный университет», г. Воронеж;

Шашкин А.И., д.ф.-м.н., профессор, декан факультета ПММ, зав. кафедрой математического и прикладного анализа ГОУ ВПО «Воронежский государственный университет», г. Воронеж.

Работа поступила в редакцию 28.04.2011.