

Bozhenyuk Alexander Vitalievich – Scientific and Technical Center «Information Technologies» of Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education «Southern Federal University»; e-mail: avb002@yandex.ru; 4, Oktyabrskaya Square, Taganrog, 347922, Russia; phone: +79198799621; dr. of eng. sc.; professor.

Gerasimenko Evgeniya Michailovna – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: e.rogushina@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: + 79885315343; junior scientific researcher.

Rozenberg Igor Naymovich – Public corporation “Research and development institute of railway engineers”; e-mail: I.kudreyko@gismps.ru; 27/1, Nizhegorodskaya, Moscow, 109029, Russia; phone: +74959677701; dr. of eng. sc.; deputy director.

УДК 681.3.06: 681.323 (519.6)

Я.Е. Ромм, А.С. Дзюба

ИДЕНТИФИКАЦИЯ РУКОПИСНЫХ СИМВОЛОВ С ПРИМЕНЕНИЕМ ПОДСТАНОВКИ ИНДЕКСОВ ПРИ СОРТИРОВКЕ ПОЛЯРНЫХ КООРДИНАТ

Представлен метод распознавания рукописных символов на основе подстановки индексов при сортировке полярных координат точек изображения. Дан алгоритм нахождения признаков рукописного символа с использованием трехкратного рекурсивного прореживания всех точек изображения посредством идентификации экстремальных элементов координат на основе сортировки. Описана классификация и целочисленная идентификация рукописных символов с использованием фильтрации и экстремальных признаков. Согласно программному эксперименту различимы все рукописные символы из алфавита русского языка и почерк. Представлены алгоритмы и результаты программного экспериментов.

Распознавание рукописных символов; сортировка подсчетом; подстановка; целочисленные идентификаторы; экстремумы полярных координат изображения.

Ya.E. Romm, A.S. Dzuba

IDENTIFICATION OF HANDWRITTEN SYMBOLS WITH USING SUBSTITUTION OF INDEXES WHEN SORTING POLAR COORDINATES

There was presented a method for recognition of handwritten symbols that based on the substitution indexes when sorting the polar coordinates of pixels. Proposed an algorithm to finding signs of handwritten symbols. It achieved by decimation of all image points with using triple recursion. There were described the classification and integers identifiers of handwriting symbols using filtration and extreme signs. According to the program's experiment, handwritten symbols from the Russian alphabet and handwriting were distinct. The algorithm and the results of program's experiments are presented as well.

Recognition of handwritten symbols; the sorting by counting; substitution; integers identifiers; extremes of polar coordinate of image.

Постановка вопроса. Выбраны четыре образца каждого из рукописных символов русского алфавита, три из которых написаны одним человеком, а четвертый – другим. Наряду с тем рассматриваются символы шрифтов ms windows, сходные по написанию с рукописными символами. Все образцы представлены в растровом изображении, при этом образцы одного класса сходны между собой, но не идентичны. Цвет написания каждого рукописного символа темный, фон – светлый. Ставится задача классификации, распознавания и компьютерной идентификации

рукописных символов данного вида при помощи целочисленных идентификаторов. Подход к ее решению строится на основе сортировки полярных координат точек изображения символа, формируемой на ее основе подстановки индексов и идентификации с ее помощью экстремальных признаков изображения.

Описание метода. Запрограммированный алгоритм [4] решения задачи конструируется следующим образом. Вначале растровое изображение с рукописным символом монохромизируется. Предполагается, что точка принадлежит рукописному символу, если она имеет черный цвет, и не принадлежит, – если она белого цвета. Для каждой точки изображения символа считываются ее координаты в декартовой системе, при этом ось абсцисс направлена в ту же сторону, что и в классической декартовой системе координат, а ось ординат – противоположно оси классической системы. Считывание производится последовательно: слева направо и сверху вниз. Сначала рассматривается левая верхняя точка изображения, затем точка под ней. После нижней левой точки, положение считывания перемещается на вторую точку слева вверх. Абсциссы и ординаты считанных точек записываются в два отдельных массива. Индекс элемента в каждом из этих массивов – порядковый номер точки при считывании. На основе сохраненных координат вычисляется центр рукописного символа, который в дальнейшем будет принят за центр новой декартовой системы с классическим расположением осей. За ординату центра принимается среднее арифметическое всех сохраненных ординат точек символа, за абсциссу – среднее арифметическое всех сохраненных абсцисс. Определенный таким образом центр для рукописных символов одного класса всегда оказывается примерно в одинаковом положении относительно всех точек символа. Для некоторого класса рукописных символов центр попадает в точку изображения символа или располагается в некоторой малой окрестности от точки изображения, для другого – на значительно большем расстоянии от точки изображения, что в дальнейшем используется при классификации. Поэтому центр изображения символа определяется как разновидность вторичной характеристики (о других возможных вторичных характеристиках будет упомянуто в дальнейшем). Найденный центр принимается за центр полярной системы координат, начальное направление ее радиуса совпадает с положительным направлением оси абсцисс. Угол полярной системы отсчитывается против часовой стрелки.

Далее, новые декартовы координаты точек изображения символа преобразуются в полярные: радиус – расстояние от точки до центра, угол его наклона – \arctg отношения ординаты к абсциссе. Полярные координаты точек записываются в два массива (углов и радиусов), связь между ними осуществляется через дополнительный массив индексов (порядковые номера точек при считывании). Радиусы упорядочиваются с помощью сортировки по неубыванию угла и записываются в новый массив, затем упорядочиваются по значению и образуют еще один новый массив (все сортировки реализованы модифицированной сортировкой подсчетом с взаимно однозначным соответствием входных и выходных индексов [1–3]). В отдельный массив записывается и перестановка первоначального положения индексов радиусов. Связь с массивом углов реализуется за счет ссылок на индексы. Далее, рекурсивно в три этапа проводится поиск локально экстремальных (максимальных) элементов, с целью отфильтровать несущественные элементы (шумы) изображения. На первом этапе входными данными являются непосредственно элементы массива радиусов всех точек изображения, упорядоченных по неубыванию, а выходными – новый массив, состоящий из прореженных с помощью локализации на основе сортировки экстремальных элементов входного массива. На втором этапе входными данными являются данные экстремальные элементы, полученные на выходе первого этапа, а выходными – новый массив, строящийся по тому же

принципу, что и массив на выходе первого этапа. Аналогично, на третьем этапе входные данные – выборка на выходе второго этапа; выходные данные – новая последовательность, состоящая из прореженных с помощью локализации на основе сортировки экстремальных элементов входного массива. При этом на каждом из трех этапов фильтрации экстремумов дополнительно выполняются следующие действия. Первый этап включает поиск заранее заданного количества максимально удаленных от центра точек изображения в отсортированном по неубыванию массиве радиусов. Количество элементов выборок этого и последующих этапов вычислены в результате программного эксперимента: для первого этапа – тысяча, для второго – тридцать, для третьего – шесть-семь. За основу выбора локально экстремальных элементов берется алгоритм поиска с помощью сортировки по соотношениям индексов [3], но в качестве условия выбора локально максимального элемента сравниваются не индексы радиусов, а соответствующие им углы. Таким образом, вводится понятие «угол поиска локально максимального элемента». Для первого этапа начальное значение угла поиска равно единице. Далее производится поиск локально максимальных элементов и их количества. Количество найденных элементов сравнивается с заранее заданным, если оно его превышает, то угол поиска пересчитывается по формуле: $gpoisk = gpoisk + 0.0001(kol - sh1)^2$, где $gpoisk$ – угол поиска, kol – априори заданное количество элементов, $sh1$ – найденное количество элементов. Эти данные в сочетании с их аналогами на следующих двух этапах используются для формирования признаков распознавания и идентификаторов изображения. Полученная на первом этапе выборка радиусов записывается в новый массив, сортируются по неубыванию угла с перезаписью в этот же массив. Перезаписывается и массив индексов в виде соответственной перестановки, при этом положения радиусов и их индексов имеют одинаковый порядковый номер в своих массивах (для сохранения связи углов и радиусов через ссылки индексов). Далее, преобразованный массив локализованных радиусов сортируется по неубыванию значений с перезаписью в новый массив, аналогично, записывается в новый массив перестановка индексов. Преобразованная выборка из тысячи элементов радиусов является входной для второго этапа. Аналогичные действия производятся на втором и третьем этапах с различиями в отмеченных значениях углов поиска и количества элементов. На всех этапах связь угла и радиуса осуществляется через ссылки индексов на основе соответственных массивов индексов. Конечная выборка радиусов упорядочивается по неубыванию угла с перезаписью массивов радиусов и индексов. Далее, она упорядочивается по неубыванию радиусов с перезаписью в новый массив, перестановка индексов сохраняется в новый массив.

В окончательной перестановке индексов имеют место устойчивые сочетания их взаимного положения. Для рукописных символов одного класса – они идентичны, для символов разных классов – попарно различны. Эти устойчивые сочетания определяются как первичная характеристика рукописного символа. Классификация всех изображений символов производится на основе целочисленной первичной и, при необходимости, вторичной характеристики (положения центра относительно точек изображения).

Программный эксперимент. Все изображения символов обработаны с помощью программы, реализующей изложенный метод [4]. Ниже на примере класса рукописных символов русской буквы «в» (рис. 1) иллюстрируются компоненты идентификации.

Первые три варианта написаны рукой одного, четвертый – рукой второго человека, пятый – шрифт. В табл. 1 приведены первичные характеристики для данных вариантов. Здесь и ниже «№» – входной номер индекса, «Re ind» – перестановка индексов, «Ex R» – экстремальный радиус.

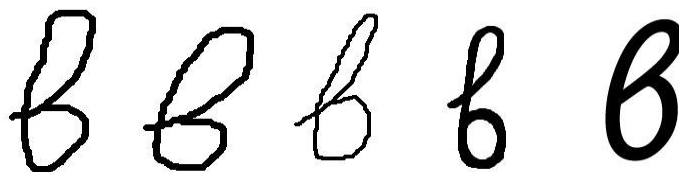


Рис. 1. Буква «в» в 5 вариантах (слева направо)

Таблица 1

№	Вариант I		Вариант II		Вариант III		Вариант IV		Вариант V	
	Re ind	Ex R	Re ind	Ex R	Re ind	Ex R	Re ind	Ex R	Re ind	Ex R
1	6	5,381	6	9,994	6	15,447	4	24,199	<u>1</u>	29,232
2	7	6,987	7	13,579	<u>3</u>	25,227	2	32,749	<u>3</u>	39,588
3	<u>3</u>	31,905	<u>3</u>	36,098	2	40,979	<u>3</u>	33,903	6	42,632
4	4	32,315	2	37,118	4	59,818	<u>5</u>	68,434	4	54,282
5	2	33,236	4	75,229	<u>5</u>	73,707	6	68,941	<u>5</u>	74,218
6	<u>5</u>	86,153	<u>5</u>	75,526	<u>1</u>	89,597	<u>1</u>	82,893	2	80,2
7	<u>1</u>	92,948	<u>1</u>	91,782	–	–	–	–	–	–

Индекс 3 всегда с точностью до единицы находится на второй позиции. Индекс 5 с точностью до единицы находится на пятой позиции. Индекс 1 находится или на последней, или на первой позиции. Индексы 2 и 4 находятся или рядом, или через одну позицию.

Для первого варианта эти особенности выражают подстановки:

$$I. \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 7 & \underline{3} & \bar{4} & \bar{2} & \underline{5} & \underline{1} \end{pmatrix} \quad (1) \quad II. \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 7 & \underline{3} & \bar{2} & \bar{4} & \underline{5} & \underline{1} \end{pmatrix} \quad (2)$$

$$III. \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & \underline{3} & \bar{2} & \bar{4} & \underline{5} & \underline{1} \end{pmatrix} \quad (3) \quad IV. \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \bar{4} & \bar{2} & \underline{3} & \underline{5} & 6 & \underline{1} \end{pmatrix} \quad (4)$$

$$V. \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \underline{1} & \underline{3} & 6 & \bar{4} & \underline{5} & \bar{2} \end{pmatrix} \quad (5)$$

Подстановки (1 – 5) характерны только для буквы «в», для других – они будут попарно различны между собой.

Первичные характеристики вариантов написания рукописных символов рукой одного человека устойчивее, чем другого. Как видно из табл. 1 для первых трех вариантов индекс 6 находится на первой позиции. Индексы 5 и 1 расположены рядом на предпоследней и последней позициях соответственно. Индексы 4 и 2 находятся в центре последовательности. Таким образом, согласно программному эксперименту первичная характеристика позволяет не только идентифицировать символ, но и классифицировать почерк человека.

Аналогично, в ходе программного эксперимента установлено, что подстановки находятся во взаимно однозначном соответствии со всеми рукописными символами алфавита русского языка, при этом соответствие сохраняет различие для рассматриваемого набора почерков [4].

Центр буквы «в», как ее вторичная характеристика, представлен на рис. 2.

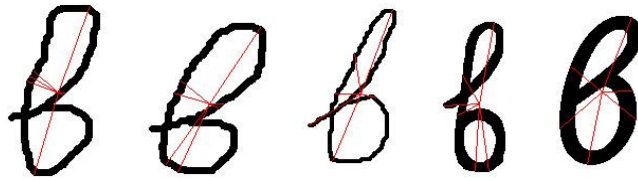


Рис. 2. Различные варианты написания символа «в». Положение ее центра и радиусов конечной выборки (центр находится в пересечении радиусов)

В качестве разновидностей вторичной характеристики можно рассматривать резкие изменения значений радиуса в преобразованной конечной выборке (см. табл. 1). Вторичные характеристики дают дополнительное упрощение классификации символа.

На основе первичной и вторичной характеристик изображения рукописный символ идентифицируется логическими условиями соответствия эталонным характеристикам базы, сформированной при помощи подстановок и учета длин полярных радиусов.

Набор логических условий на примере буквы «в» строится в программном коде следующим образом:

```

k1:=0;k2:=0;k3:=0;k4:=0;k5:=0;ku:=0;
k1:=pos('3',s);k2:=pos('5',s);k3:=pos('1',s);k4:=pos('4',s);k5:=pos('2',s);
if (abs(k1-2)<=1) then begin
  ku:=ku+1;
  if (abs(k2-5)<=1) then begin
    ku:=ku+1;
    if ((k3=1) or (k3=length(s))) then begin
      ku:=ku+1;
      if abs(k4-k5)<=1 then begin
        ku:=ku+1;
        if center_black=true then begin
          ku:=ku+1;
          if change_radius=true then begin
            ku:=ku+1;
          end;
        end;
      end;
    end;
  end;
end;
end;
end;
end;
if kol<ku then begin
  kol:=ku;
  letter:='в';
end;

```

В данном фрагменте кода s – перестановка индексов, представленная в виде строки, ku – счетчик выполненных условий, kol – максимальное количество выполненных условий из всех наборов, $letter$ – идентифицированный символ.

Аналогично составляются логические условия для идентификации других изображений букв русского алфавита. В идентифицирующей программе проверяются все наборы логических условий, взаимно однозначно определяющие изображение символа, полный программный эксперимент представлен в [4].

В дополнение к изложенному методу используется циклический поворот полярных координат по/против часовой стрелки до ближайшего угла элемента конечной выборки. Это приводит к изменению нумерации элементов конечной выборки и, впоследствии, к изменению порядка следования индексов. Согласно эксперименту [4] для данной операции необходимо и достаточно получить конечную выборку по изложенному методу ровно из шести элементов. Соответственно число циклических поворотов равно шести. В одной из последовательностей индексов всегда появляется устойчивая характеристика рукописного символа, отличная от всех других, как это иллюстрирует табл. 2 для изображения символа «а» (рис. 3).



Рис. 3. Буква «а» в 5 вариантах (слева направо)

Таблица 2

№ варианта	I	II	III	IV	V
Порядковые номера	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6
Циклическая перестановка	$\begin{pmatrix} 6 & 4 & 2 & 3 & 1 & 5 \\ 1 & 5 & 3 & 4 & 2 & 6 \\ 2 & \underline{6} & \underline{4} & \underline{5} & \underline{3} & 1 \\ 3 & 1 & 5 & 6 & 4 & 2 \\ 4 & 2 & 6 & 1 & 5 & 3 \\ 5 & 3 & 1 & 2 & 6 & 4 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 4 & 5 & 3 & 6 \\ 2 & 3 & 5 & 6 & 4 & 1 \\ 3 & \underline{4} & \underline{6} & 1 & \underline{5} & \underline{2} \\ 4 & 5 & 1 & 2 & 6 & 3 \\ 5 & 6 & 2 & 3 & 1 & 4 \\ 6 & 1 & 3 & 4 & 2 & 5 \end{pmatrix}$	$\begin{pmatrix} 5 & 1 & 4 & 2 & 3 & 6 \\ 6 & 2 & 5 & 3 & 4 & 1 \\ 1 & 3 & \underline{6} & \underline{4} & \underline{5} & \underline{2} \\ 2 & 4 & 1 & 5 & 6 & 3 \\ 3 & 5 & 2 & 6 & 1 & 4 \\ 4 & 6 & 3 & 1 & 2 & 5 \end{pmatrix}$	$\begin{pmatrix} 1 & \underline{6} & \underline{4} & 3 & \underline{5} & \underline{2} \\ 2 & 1 & 5 & 4 & 6 & 3 \\ 3 & 2 & 6 & 5 & 1 & 4 \\ 4 & 3 & 1 & 6 & 2 & 5 \\ 5 & 4 & 2 & 1 & 3 & 6 \\ 6 & 5 & 3 & 2 & 4 & 1 \end{pmatrix}$	$\begin{pmatrix} 5 & 3 & 1 & 4 & 2 & 6 \\ 6 & 4 & 2 & 5 & 3 & 1 \\ 1 & 5 & 3 & 6 & 4 & 2 \\ 2 & \underline{6} & \underline{4} & 1 & \underline{5} & \underline{3} \\ 3 & 1 & 5 & 2 & 6 & 4 \\ 4 & 2 & 6 & 3 & 1 & 5 \end{pmatrix}$

Циклическая обработка упрощает идентификацию на основе сравнения матриц перестановок с эталонными прототипами. Вместе с тем можно совместить данный способ идентификации с проверкой логических условий соответствия характеристикам эталона с целью повышения устойчивости предложенного метода распознавания изображения рукописного символа.

Заключение. Изложенный метод позволяет на основе подстановки индексов при сортировке полярных координат множества точек изображения распознать и идентифицировать рукописный символ. При ограниченном уровне искажений изображения метод сохраняет значимые целочисленные идентификаторы символов, которые сравнительно просто соотносятся с эталонами. В [4] дан полный алгоритм и его программная реализация, позволяющая различать все рукописные символы русского алфавита с учетом почерка.

Предложенный метод отличается от известных [5–8] по построению, целочисленностью компонентов идентифицирующих изображение векторов, что влечет устойчивость идентификации, упрощение вычислений и структуры базы эталонов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ромм Я.Е. Параллельная сортировка слиянием по матрицам сравнений. Ч. I // Кибернетика и системный анализ. – 1994. – № 5. – С. 3-23.
2. Ромм Я.Е. Параллельная сортировка слиянием по матрицам сравнений. Ч. II // Кибернетика и системный анализ. – 1995. – № 4. – С. 13-37.
3. Ромм Я.Е. Локализация и устойчивое вычисление нулей многочленов на основе сортировки. Ч. I // Кибернетика и системный анализ. – 2007. – № 1. – С. 165-183.

4. Ромм Я.Е., Дзюба А.С. Метод распознавания рукописных символов на основе сортировки полярных координат. – Таганрог: Изд-во ТГПИ, 2012. – 42 с.
5. Tou J.T., Gonzalez R.C. Pattern recognition principles. – Addison-Wesley publishing company. – 1974.
6. Ян Д.Е. Исследование, развитие и реализация методов автоматического распознавания рукописных текстов в компьютерных системах: Дисс. ... канд. физ.-мат. наук. – М., 2003. – 179 с.
7. Горошкин А.Н. Применение векторного подхода к распознаванию рукописных символов: научная статья // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2006. – С. 15-17.
8. Рюмин О.Г. Разработка и исследование алгоритмов распознавания изображений на основе определения экстремальных признаков замкнутых контуров с помощью сортировки: Автореф. дисс. ... канд. техн. наук. – Таганрог: Изд-во ТТИ ЮФУ, 2008, – 16 с.

Статью рекомендовал к опубликованию д.т.н., профессор Д.П. Фельдман.

Ромм Яков Евсеевич – ФГБОУ ВПО "Таганрогский государственный педагогический институт имени А.П. Чехова"; e-mail: romm@list.ru; 347926, г. Таганрог, ул. Инициативная, 48; тел.: 88634601753, 88634601812, 88634601807; 89094081126; д.т.н.; профессор; член Европейской Академии Естествознания (EuANH); зав. кафедрой информатики.

Дзюба Андрей Сергеевич – e-mail: dzuba_ni@aaanet.ru; тел.: 89094080776; аспирант.

Romm Yakov Evseevich – Taganrog State Pedagogical Institute; e-mail: romm@List.ru; 48, Initiativnaya street, 347926, Russia; phones: +78634601753, +78634601812, +78634601807, +79094081126; dr. of eng. sc.; professor; the member of European Academy of Natural History (EuANH); head of the chair of information science.

Dzuba Andrew Sergeevich – e-mail: dzuba_ni@aaanet.ru; phone: +79094080776; postgraduate student.

УДК 519.6:681.3

Я.Е. Ромм, И.И. Стаховская

ВРЕМЕННАЯ СЛОЖНОСТЬ ПАРАЛЛЕЛЬНОГО КУСОЧНО-ИНТЕРПОЛЯЦИОННОГО ВЫЧИСЛЕНИЯ ФУНКЦИЙ

Представлены алгоритмы и оценки временной сложности параллельного кусочно-полиномиального вычисления функций одной действительной переменной на основе интерполяционного полинома Ньютона. Для рассматриваемой схемы учитывается вариация степени полинома и количества подынтервалов, различные границы погрешности, предложена максимально параллельная форма вычислительного алгоритма. На модели неветвящихся параллельных программ достигается оценка временной сложности $O(\log_2 N)$, где N – заданная граница вариации степени полинома, количество процессоров соответствует максимальному числу подынтервалов и проверочных точек. Для стандартных функций достигается временная сложность вычисления $O(1)$. Синхронно со всеми полиномиальными приближениями функций выполнимо параллельное вычисление их производных и определенных интегралов.

Кусочно-полиномиальное вычисление функций; параллельные вычисления; временная сложность; интерполяционный полином Ньютона.