

В. В. С ю з е в, А. А. Х а н и н

**ГИБРИДНЫЙ МЕТОД ОПТИЧЕСКОГО  
РАСПОЗНАВАНИЯ ТЕКСТА С КОРРЕКЦИЕЙ  
РЕЗУЛЬТАТОВ РАСПОЗНАВАНИЯ**

*Рассмотрена задача перевода информации с бумажных носителей в электронный вид. Предложен гибридный метод оптического распознавания символов на основе комплексирования результатов анализа полутоновых и бинарных изображений. Предложена метрика нечеткого поиска для орфографической и синтаксической коррекций распознанного текста с использованием словаря. Разработан программный комплекс для распознавания текстовых изображений и коррекции текста. Проведено сравнение результатов распознавания полутоновых изображений, полученных различными методами, показавшее высокую точность разработанного метода.*

**E-mail: a.khanin@vniins.ru; alexandr.khanin@gmail.com**

**Ключевые слова:** обработка изображений, сегментация, распознавание символов и текста, синтаксический анализ, парсинг, OCR.

Оптическое распознавание символов (optical character recognition), далее — OCR, является технологией механического или электронного перевода отсканированных изображений рукописного, машинописного или печатного текстов в электронный вид. OCR позволяет сократить объем занимаемой информации, делает возможным редактирование текста, поиск по слову или фразе, позволяет просматривать или распечатывать документы без артефактов сканирования и применять к тексту такие действия, как машинный перевод, синтез речи и т.д.

Оптическое распознавание текста является исследуемой проблемой в областях распознавания образов, искусственного интеллекта и компьютерного зрения. Работы в данном направлении ведутся уже на протяжении нескольких десятилетий, и достигнутые результаты впечатляют. В настоящее время разработано достаточно большое число разнообразных методов. В отличие от ранних версий, существующие OCR подходы не требуют калибровки для работы с конкретным шрифтом и позволяют обеспечить достаточно высокую вероятность достоверного распознавания большинства шрифтов. Некоторые системы оптического распознавания текста способны восстанавливать исходное форматирование текста, включая изображения, колонки и другие нетекстовые компоненты.

Сегодня OCR-технологии являются неотъемлемой частью повседневной жизни и широко используются для конвертации книг и документов в электронный вид, для автоматизации систем учета в бизнесе или публикации текста на веб-странице. Помимо стандартных

применений возможно использование OCR в корпоративных системах фильтрации спама для случаев, когда спам рассылается в виде изображений, а также для предотвращения утечек секретной информации из организаций — подобные данные очень часто представлены в виде отсканированных копий документов.

Поскольку многие изображения и архивные документы обычно имеют достаточно низкое качество и содержат артефакты, даже современные модули распознавания генерируют много ошибок распознавания. Следует отметить, что существующего в некоторых методах набора простых синтаксических правил в большинстве случаев недостаточно. Очевидно, классические подходы к оптическому распознаванию текстов подходят к своему технологическому пределу. По мнению автора, общий контекст операций преобразования документов в электронный вид предоставляет возможность распознавания текста на уровне всего документа и знания коллекции уровня, что целесообразно использовать для решения задач конвертации документа в электронный вид с минимальным числом ошибок.

В настоящей статье рассматривается техника эффективного перевода изображений документов текста в последовательность кодов, использующихся для представления в текстовом редакторе. При этом в центре внимания оказывается симбиоз оптического распознавания символов и постпроцессора результатов. Предложен разработанный автором гибридный алгоритм оптического распознавания текста со встроенной коррекцией результатов распознавания. Основная идея OCR-алгоритма состоит в комплексировании результатов распознавания полученного бинарного изображения разработанным набором шаблонов и дескрипторов полутоновых изображений машиной опорных векторов (support vector machine), далее SVM [1]. Коррекция результатов распознавания осуществляется по словарю и набору знаний на базе структур верхнего уровня (слово, строка) и позволяет наделять систему искусственным интеллектом и существенно повысить вероятность распознавания отдельных символов — по аналогии с человеческим мозгом информация воспринимается не однобоко, а с учетом рядом стоящих символов и контекста. Ядро постпроцессора построено вместо наиболее часто применяемого расстояния Левенштейна [2] на нетривиальной метрике — лексические структуры представляются радиусом-вектором в цифробуквенной системе координат, что позволяет существенно повысить скорость алгоритмов нечеткого поиска по словарю для устранения ошибок модуля распознавания и орфографической коррекции документа.

Приведены результаты экспериментов по измерению качества распознавания OCR-модуля как в отдельности, так и при использовании знаний на уровне документа постпроцессором.

**Предварительная обработка.** Отсканированные документы и книги, как правило, для удобства восприятия человеком хранятся в виде цветных изображений или изображений в оттенках серого, в то время как подавляющее большинство OCR-систем для перевода изображения в последовательность символов используют бинарные изображения. Для разработанного OCR-метода данное утверждение тоже справедливо, но только с важным замечанием – на этапе распознавания используются и бинарное, и полутоновые изображения.

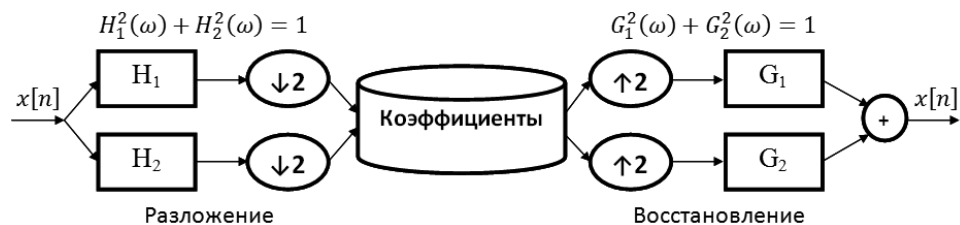
Рассмотрим шумоподавление и бинаризацию отсканированных печатных документов в качестве предварительного шага обработки для распознавания текста.

*Вейвлет-фильтрация.* Применительно к области шумоподавления вейвлет-преобразование позволяет удалять шум с изображения, слабо затрагивая границы и детали. Оно является доказуемо оптимальным для аддитивного гауссового белого шума с известной дисперсией [3], а также позволяет эффективно подавлять шумы со спектрами, отличными от белого. Схематично одномерное дискретное вейвлет-преобразование изображено на рис. 1.

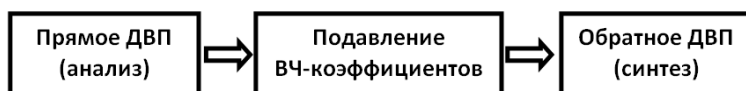
Вейвлет-фильтр применяется к изображению в оттенках серого до бинаризации. Автор полагает, что читатель знаком с теорией вейвлетов, поэтому далее приведена лишь общая схема фильтрации изображения (рис. 2).

Следует отметить, что присутствующие шум и артефакты в отсканированных документах не являются белым гауссовым шумом в чистом виде. Тем не менее эксперименты показали, что в большинстве случаев этот подход позволяет улучшить качество изображения и является полезным шагом предварительной обработки.

*Бинаризация диффузией ошибок.* Следующим после фильтрации изображения этапом является бинаризация методом диффузии оши-



**Рис. 1.** Схема одномерного вейвлет-преобразования



**Рис. 2.** Общая схема подавления шума на основе вейвлет-преобразования

бок. Диффузия ошибок (MarginED) — метод бинаризации, который часто используется для растривания изображений [4], т.е. преобразования изображений в оттенках серого в однобитные изображения (бит соответствует точке, печатаемой плоттером). Как показывает практика, MarginED позволяет получить очень качественные растровые изображения, сохраняя локальные особенности изображений, а шум, добавляемый данным типом бинаризации, преимущественно высокочастотный. Кроме того, MarginED автоматически усиливает резкость краев при бинаризации.

Выбор именно этого типа бинаризации для OCR обоснован тем, что после округления значения пикселя, результирующая ошибка (между округленным и оригинальным значениями пикселя изображения) передается дальше для корректировки результатов. Впоследствии движение накопленной ошибки вперед позволяет компенсировать ошибки бинаризации прошлых пикселей. С учетом описанного эффекта порог бинаризации выбирается совместно (причинным образом) по группе соседних пикселей. Обычно диффузия ошибки выполняется в процессе сканирования растрового изображения, поэтому предпочтительно представление изображения в виде вектора конкатенированных строк с индексацией слева направо. Выполняются следующие операции сравнения с пороговым значением  $l$  (обычно  $l = 0,5$ ):

$$b_i = 1, \text{ если } x_i + e_i \geq l;$$

$$b_i = 0, \text{ если } x_i + e_i < l.$$

Вектор  $e$  — вектор ошибок бинаризации. Это взвешенная кумулятивная сумма значений ошибок на предыдущих итерациях:

$$e_i = \sum_{h=1}^H (x_{i-h} - b_i) f_h.$$

Фильтр ошибок  $f$  должен взвешивать и передавать ошибку между оригинальным пикселем и его бинарной версией. Чаще всего в качестве подобного фильтра используется фильтр Флойда–Штейнберга [5], который применяет к передаваемой ошибке следующий пространственный шаблон:

$$\begin{array}{ccc} & & 7 \\ & x_i & \frac{7}{16} \\ \frac{3}{16} & \frac{5}{16} & \frac{1}{16} \end{array}.$$

Следует отметить, что сумма значений фильтра равна единице — это стандартное ограничение для фильтров диффузии ошибок, и оно

должно гарантированно соблюдаться. Однако оно не работает в таком виде применительно к бинаризации для OCR-алгоритмов, так как значения изображения сохраняются, что приводит к сильному зашумлению фона, если значение фона отлично от нуля. Вместо этого в используемой бинаризации MarginED диффундируют только ошибки или неопределенные пиксели. Пиксели, определенные точнее, бинаризуются полностью.

Алгоритм состоит из пяти шагов и использует вспомогательный массив  $z$ ,  $z_i \in [0, 1]$  (рис. 3).

*Шаг 1.* Инициализируем  $z_i = x_i$  для всех  $i$ .  $\delta = std(x_i)/2$ ,  $t = OtsuThreshold(x)$ .

*Шаг 2.*  $z_i = 0$  для всех  $x_i < (t - \delta)$ .

*Шаг 3.*  $z_i = 1$  для всех  $x_i > (t + \delta)$ .

*Шаг 4.*  $z_i = (z_i - (t - \delta)) / 2\delta$  для всех  $(t - \delta) \leq x_i \leq (t + \delta)$ .

*Шаг 5.* Устанавливаем  $b_i$  равным  $i$ -му значению диффузирующей ошибки массива  $z$ , используя порог 0,5 и фильтр Флойда–Штейнберга.

На этапе предварительных испытаний проводилось сравнение различных фильтров диффузии ошибок, включая классические фильтры диффузии ошибок [4], green noise фильтры [6], и фильтры, специально созданные для обработки вертикальных, горизонтальных и диагональных особенностей символов. Наилучшие результаты бинаризации для последующей сегментации были получены именно при использовании фильтра Флойда–Штейнберга.

**Сегментация.** Следующим этапом OCR-алгоритма является сегментация полученного бинаризованного изображения. Как правило, печатные документы имеют прямоугольную форму. В идеальных случаях строки текста параллельны или перпендикулярны краям страницы. Однако из-за неточности или трудности в размещении оригиналов документов в процессе сканирования захваченные края документов не всегда могут совпадать с краями изображения, что приводит к наклону текста на изображении. Поэтому корректное определение угла наклона и сегментация текста являются необходимыми условиями качественного распознавания.

В данном разделе описан алгоритм сегментации изображения, способный сегментировать документы с несколькими областями текста разного наклона. В его основе лежит построение полигонов (рис. 4) лексических структур — минимально возможных выпуклых многоугольников, полностью содержащих символы, слова, параграфы.

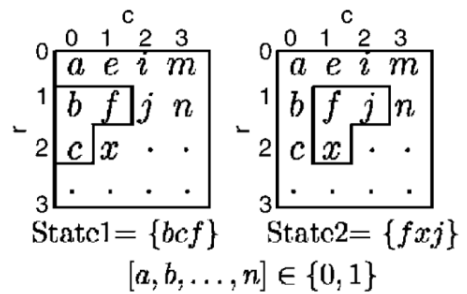


Рис. 3. Одна итерация

Whorator calculates, and reports in Fig. 7(a) of his recent article,<sup>6</sup> the fundamental of a fluorescent white daylight which he has given as a reference to his observers to

Рис. 4. Фрагмент текста, сегментированного методом выпуклых полигонов

Для того чтобы обнаружить блоки текста, используется функция группировки компонентов, разделяющая части документа на группы по критериям схожести дистанций и размеров. Далее вычисляются выпуклые полигоны обнаруженных групп, по углам наклона граней которых определяется доминантный угол наклона блока текста. Этот метод хорошо работает даже при наличии сильных шумов и на документах с большим количеством графической информации. Устойчивость к шумам объясняется следующими особенностями:

- 1) группирующая функция способна разделять компоненты различных размеров и форм;
- 2) грани выпуклых полигонов способны предопределять направление текста в блоке.

Предложенный метод является адаптивным и демонстрирует конкурентоспособную точность сегментации в сочетании с высокой скоростью работы.

*Группировка компонентов.* Пусть известен компонент  $c_1$  области  $s_1$  и компонент  $c_2$  области  $s_2$ . Если евклидово расстояние между центроидами компонентов  $c_1$  и  $c_2$  меньше либо равно значению группирующей функции  $f(s_1, s_2)$ , компоненты считаются непосредственно связанными. Связанная группа — это множество компонентов, для каждой пары которых всегда существует хотя бы одна непосредственная связь:

$$f(s_1, s_2) = \sqrt{k s_1 s_2 / (s_1 + s_2)}, \quad (1)$$

где  $k$  — скалярный параметр, который может изменяться в процессе построения непосредственных связей между компонентами.

Группирующая функция (1) имеет несколько полезных свойств: является мерой расстояния; симметрична относительно компонентов  $c_1$  и  $c_2$ ; инвариантна к повороту; инвариантна к масштабу; существует возможность выбора площадей компонентов в уравнении (1), таких как общее число пикселей, площадь выпуклого полигона, площадь ограничивающего прямоугольника; вероятность объединения символов с графическими компонентами крайне мала ввиду существенной несхожести их размеров, благодаря чему метод устойчив к шумам.

Далее приведен псевдокод описанного алгоритма.

Пусть  $CC$  — массив компонентов размера  $L$ ;

Пусть  $CC[p]$  — отдельный компонент,  $p \in [0, L)$ ;

Пусть  $area(CC [p])$  — функция площади  $p$ -го компонента;  
Пусть  $dist(CC [p], CC [p1])$  — евклидово расстояние между  $p$ -м и  $p1$ -м компонентами;  
Пусть  $f(area(CC [p]), area(CC [p1]))$  — группирующая функция  $p$ -го и  $p1$ -го компонентов;  
Пусть  $swar(area(CC [p]), area(CC [p1]))$  — функция перестановки  $p$ -го и  $p1$ -го компонентов;  
 $p = p1 = 0$ ;  
while ( $p1$  меньше  $L$ )  
{  
     $p2 = p1 + 1$ ;  
    while ( $p1$  меньше  $p2$ )  
    {  
         $p = p2$ ;  
        while( $p$  меньше  $L$ )  
        {  
            if ( $dist(CC[p], CC[p1])$  меньше  $f(area(CC[p]), area(CC[p1]))$ )  
            {  
                if ( $p$  не равно  $p2$ )  
                 $swar (area(CC [p]), area(CC [p1]))$  ;  
                инкрементировать  $p2$ ;  
            }  
            инкрементировать  $p$ ;  
        }  
        инкрементировать  $p1$ ;  
    }  
     $CC[p0, p1]$  : обнаруженная группа;  
     $CC[p1, p2]$  : принадлежит другой группе;  
     $p0 = p1$ ;  
}

Вычислительная сложность алгоритма составляет  $O(n^2)$  в худшем случае, когда каждый компонент изолирован на изображении с общим числом компонентов  $n$ . Очевидно, что для реальных данных затрачиваемое время будет существенно меньше. Описанная реализация гарантирует, что для каждой пары компонентов будет определена только одна прямая связь в процессе группировки. Кроме того, гарантируется независимость результата группировки от выбранного стартового компонента. В итоге результаты начальной группировки объединяются по локализации и критерию пересечения в выпуклые полигоны блоков текста разного уровня.

Далее описаны роль параметра  $k$  и выбор его значения. Для группы из  $n_g$  компонентов плотность  $p_g$  определяется как отношение суммы

площадей выпуклых полигонов компонентов  $s_i$  в группе  $k$  площади выпуклого полигона группы  $s_g$ :

$$p_g = \sum_{i=0}^{n_g-1} s_i / s_g. \quad (2)$$

Для изображения с числом групп, равным  $N$ , взвешенная плотность  $p_{wtd}$  определяется как отношение суммы плотностей групп  $p_g$  к сумме площадей групп  $s_g$ .

$$P_g = \sum_{g=0}^{N-1} p_g s_g / \sum_{g=0}^{N-1} s_g = \sum_{g=0}^{N-1} \sum_{i=0}^{n_g-1} s_i / \sum_{g=0}^{N-1} s_g. \quad (3)$$

На рис. 5 показаны уровни группировки в процессе изменения параметра  $k$ . При  $k = 0$  все компоненты изолированы, так что взвешенная плотность равна единице. При росте значения  $k$  компоненты постепенно объединяются во все более крупные группы в следующей последовательности: символы  $\rightarrow$  слова  $\rightarrow$  строки  $\rightarrow$  параграфы  $\rightarrow$  блоки  $\rightarrow$  страницы. Задача заключается лишь в поиске наименьшего значения  $k$ , при котором формирование параграфов стабилизируется. Цикл начинается со значения  $k = 35$ . При постепенном уменьшении вычисляется разность взвешенных плотностей, которая не должна составлять более 0,05.

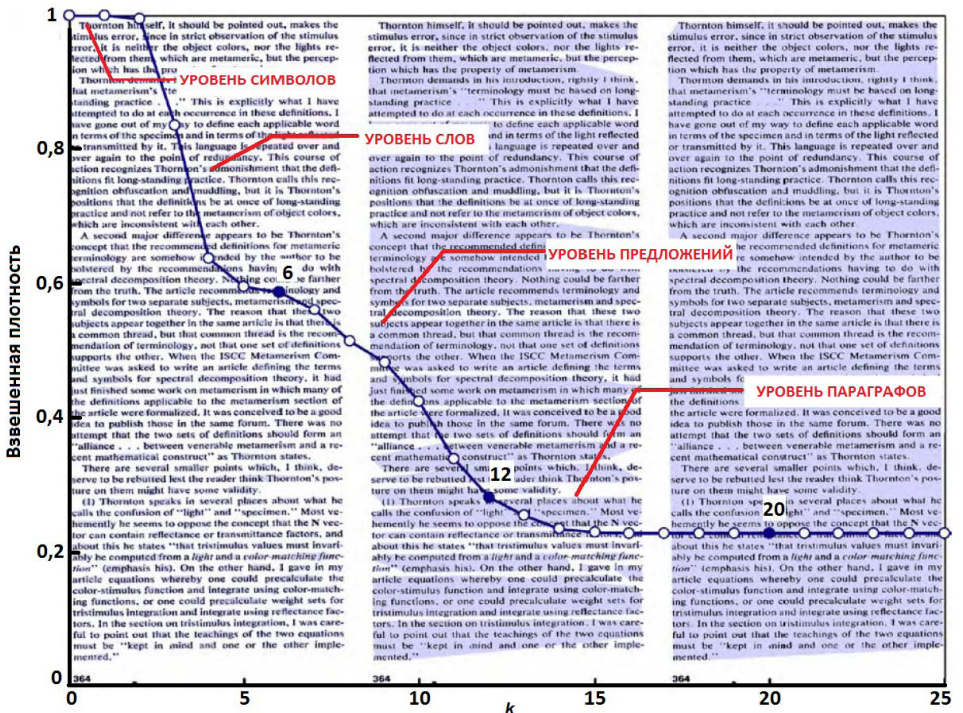


Рис. 5. Стадии группировки компонентов (для  $k = 6, 12, 20$ )



Следует отметить, что описанный механизм группировки инвариантен относительно разрешения отсканированного изображения (конечно, если оно не слишком мало).

*Определение угла наклона* для сгруппированных структур текста является тривиальной задачей, поэтому останавливаться на ней не будем. Методы определения угла наклона изложены во многих источниках [7–9], и при необходимости читатель может ознакомиться с ними.

**Распознавание символов.** После сегментации и определения угла наклона, когда вышеописанным алгоритмом определена структура документа, извлеченные сегменты передаются в модуль распознавания для последовательной обработки. Следует отметить, что в данном разделе будет описан алгоритм распознавания отдельных символов, хотя в разработанном модуле эффективная работа достигается коррекцией результатов на завершающих стадиях распознавания отдельных групп символов. Предлагаемый алгоритм оптического распознавания символов основан на комплексировании результатов распознавания бинарного и полутонового изображений символа.

*Корреляционное сравнение с шаблонами.* Первым этапом распознавания сегментированного символа является вычисление его коэффициента корреляции  $r$  со специально разработанными шаблонами (рис. 6):

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right) \left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

Классические подходы шаблонного распознавания бинарных изображений подразумевают вычисление корреляции или меры близости между извлеченным бинарным сегментом и коллекцией бинарных шаблонов. Новизна предлагаемого подхода заключается в корреляционном сравнении поданного на вход бинарного символа с набором четырех- или шестибитных шаблонов с допустимым смещением  $\pm 1$  пиксель по горизонтали и вертикали. Шаблоны отличаются от классических наличием запретных областей — при попадании части распознаваемого сегмента в такую область возвращается штраф, пропорциональный удаленности от правильной формы символа.

В результате многочисленных экспериментов разработанный метод подтвердил свою адекватность. Кроме того, он позволяет эффективно различать такие затруднительные для обычных алгоритмов пары символов, как К–Х, М–Н и т.п.

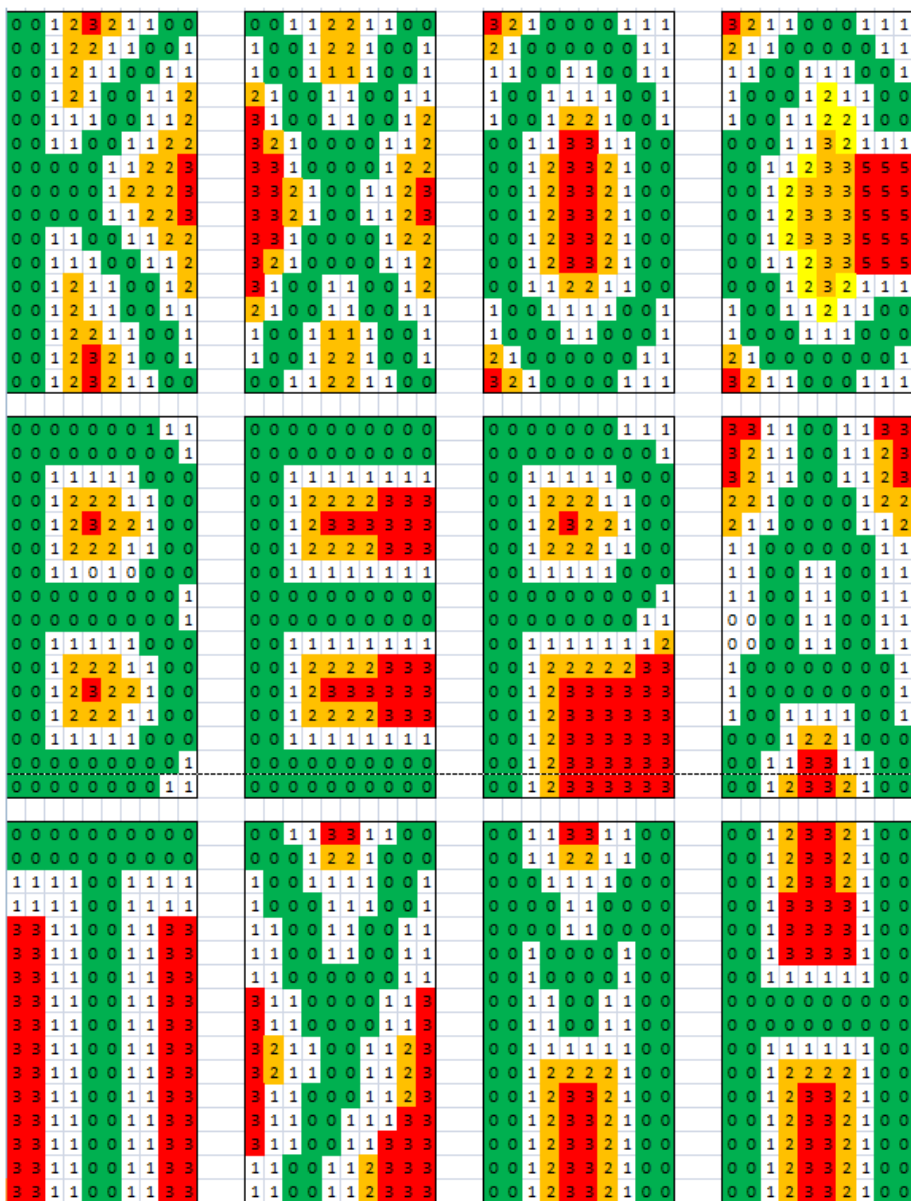


Рис. 6. Примеры разработанных шаблонов символов

*Распознавание с использованием SVM.* В результате бинаризации изображения и последующей сегментации становятся известными границы символов. Однако, как известно, в результате бинаризации могут возникать ложные контуры и сегменты, которые могут оказать негативное влияние на качество распознавания. Для нивелирования этого эффекта автором было предложено дополнительно распознавать полутоновые изображения в пределах известных границ, полученных на предыдущих этапах. Использование полутоновых изображений по-

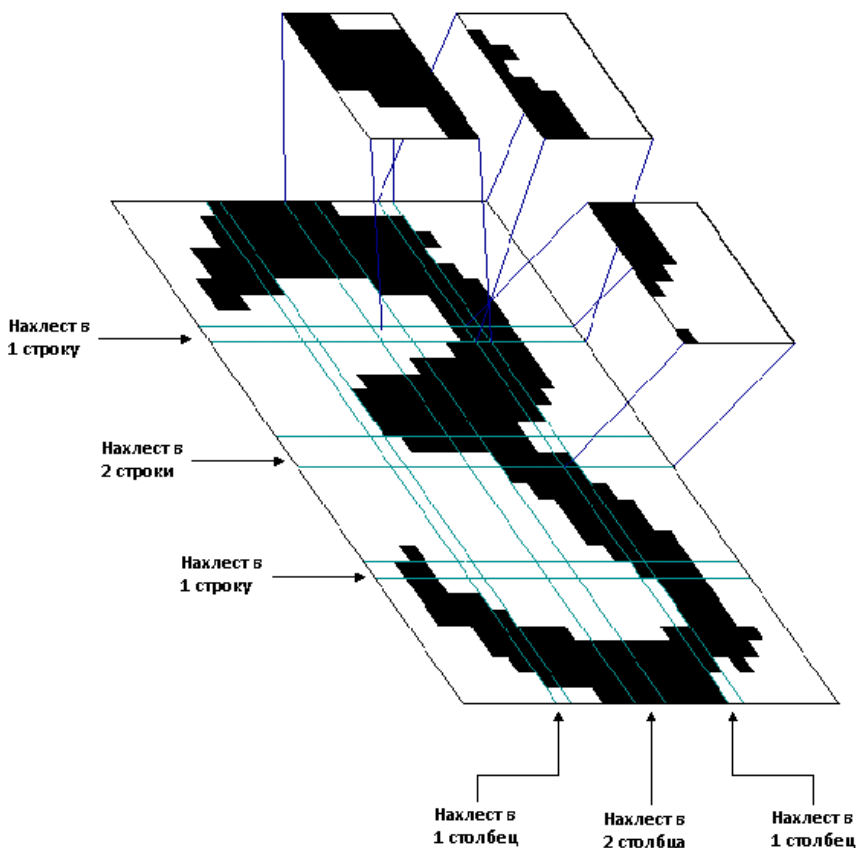


Рис. 7. Схема разбиения сегмента для построения дескриптора

зволило исключить возможные “выбросы” бинаризации и повысить робастность алгоритма распознавания.

В результате экспериментов было получено наилучшее разбиение символов на блоки (рис. 7), позволяющее учесть структурные особенности символа.

Для каждого полученного сегмента с учетом нахлестов строится гистограмма ориентированных градиентов [10], значения которой последовательно записываются в вектор дескриптора распознаваемого символа.

В качестве классификатора была выбрана машина опорных векторов (SVM) [11], не раз доказавшая свою эффективность в задачах многоклассовой классификации, в том числе и в OCR. Рассматривались модификации SVM с линейным, гауссовым и кубическим полиномиальным ядрами:

$$K_{linear}(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y};$$

$$K_{Gaussian}(\vec{x}, \vec{y}) = e^{-\gamma \|\vec{x} - \vec{y}\|^2};$$

$$K_{cubic}(\vec{x}, \vec{y}) = (\gamma \vec{x} \cdot \vec{y} - 1)^3.$$

Обучение SVM проводилось путем решения следующей проблемы оптимизации:

$$\min_{\vec{\omega}, b, \varepsilon} \frac{1}{2} K(\vec{\omega}, \vec{\omega}) + C \sum_{i=1}^l \varepsilon_i$$

при условии, что

$$y_i (K(\vec{\omega}, \vec{x}_i) + b) \geq 1 - \varepsilon_i$$

Здесь  $\vec{\varepsilon}$  – вектор размерности  $l$ , а  $\vec{\omega}$  – вектор в том же пространстве признаков, что и  $\vec{x}_i$ . Решение определяет следующий классификатор:

$$f(\vec{x}) = \text{sign}(K(\vec{\omega}, \vec{x}) + b).$$

Для обучения и тестирования вероятности распознавания на отдельных символах использовалась база данных символов CEDAR. Обучение осуществлялось с помощью открытой библиотеки LibSVM. Вероятность распознавания символа классификатором измерялась как

$$\text{MinRecall} = \min_{c=-1,1} P(f(\vec{x}) = y | y = c).$$

В сводной табл. 1 приведены вероятности, достигнутые на 10000 тестовых символов с использованием описанных модификаций SVM.

Таблица 1

MinRecall	Вероятности, %, для модификации SVM		
	linear	Gaussian	cubic
>0	100,00	100,00	100,00
>,5	98,67	99,05	99,05
>,6	98,48	98,67	98,48
>,7	98,30	98,30	98,30
>,8	97,35	97,54	97,35
>,9	95,83	95,64	95,64
>,95	92,99	93,18	92,61
>,97	90,34	89,77	89,96
>,99	84,28	82,95	84,28
>,995	78,22	74,62	77,84
>,999	69,13	64,39	69,89

В итоге по соотношению качество–скорость была выбрана именно линейная модификация ядра SVM.

*Комплексирование результатов.* Описанные два этапа распознавания выполняются независимо друг от друга. Решение о выборе символа, соответствующего поданному на вход сегменту, принимается на

базе первого метода при отличии доминантного значения корреляции более чем на  $\beta$  (по умолчанию  $\beta = 0,2$ ). В противном случае предпочтение будет отдано результату, полученному с помощью SVM.

В результате тестов комплекса *SpecTemplateCorr + SVM linear* на описанной тестовой выборке была достигнута вероятность распознавания 99,17 %.

**Коррекция результатов.** Как было сказано ранее, коррекция результатов OCR-модуля с учетом контекста документа может быть мощным инструментом повышения общего качества распознавания. Кроме того, даже если документ распознан абсолютно верно, нельзя исключать наличие допущенных в нем орфографических ошибок, коррекция которых тоже является необходимым этапом. Автором был разработан алгоритм подобной коррекции. Идея метода заключается в использовании общих словарей для проверки существования распознанных слов. Чаще всего имеется априорная информация об области принадлежности документа, что позволяет использовать тематические словари.

На завершающей стадии распознавания группы символов, образующих слово, выполняется нечеткий поиск близких ему слов. В качестве примера приведем известную демонстрацию близких слов с учетом возможных орфографических ошибок, удалений, вставок и перестановок букв (рис. 8.).

Весь спектр слов справа, возможно полученных в результате ошибочного распознавания или опечатки, должен быть приведен к единственному верному слову с учетом рода и падежа. Нечеткий поиск подразумевает использование критерия близости слов — метрики. Наиболее часто применяемой метрикой является расстояние Левенштейна [1] или расстояние редактирования. Автором был разработан метод коррекции, в основе которого лежит векторное представление слов. Суть подхода поясняет рис. 9.

Слово представляется в виде радиуса-вектора в многомерном евклидовом пространстве, число измерений которого определяется числом допустимых символов в системе. На рисунке отчетливо видно, что два трудно сравнимых, но близких, слова имеют очень похожую характерную подпись. Мерой близости в данном случае является скалярное произведение полученных векторов.

Для тестирования применялся отсканированный текст тематики Computer Science общим числом слов 1824, число символов равнялось 13 680. Тестирование проводилось с использованием вручную составленного словаря из 250 слов аналогичной тематики. Результаты эксперимента проводятся в табл. 2.

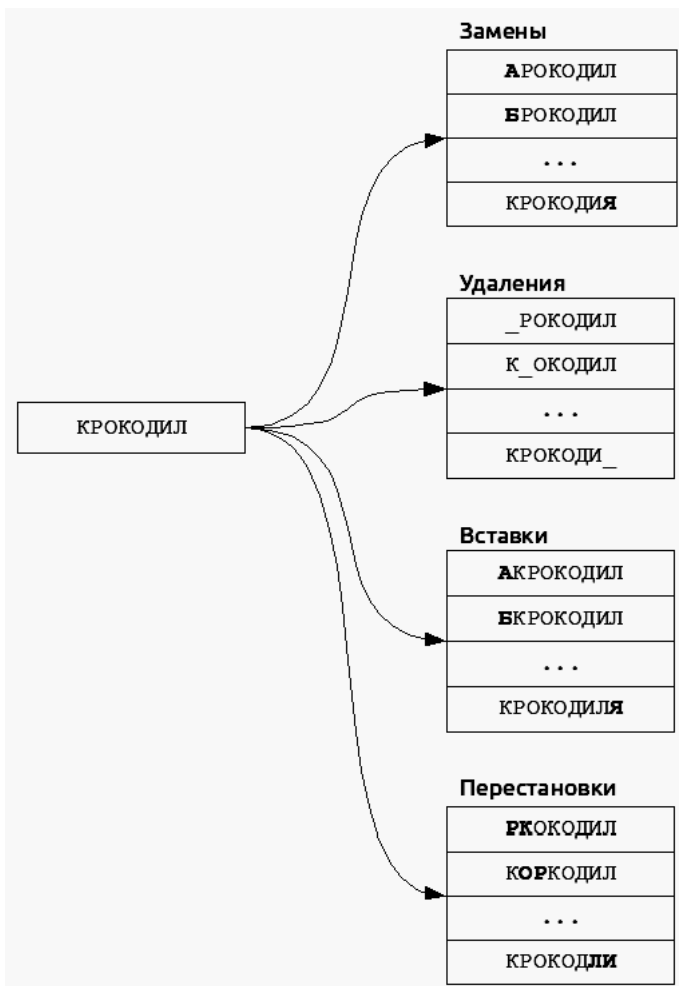


Рис. 8. Пример близких слов для нечеткого поиска

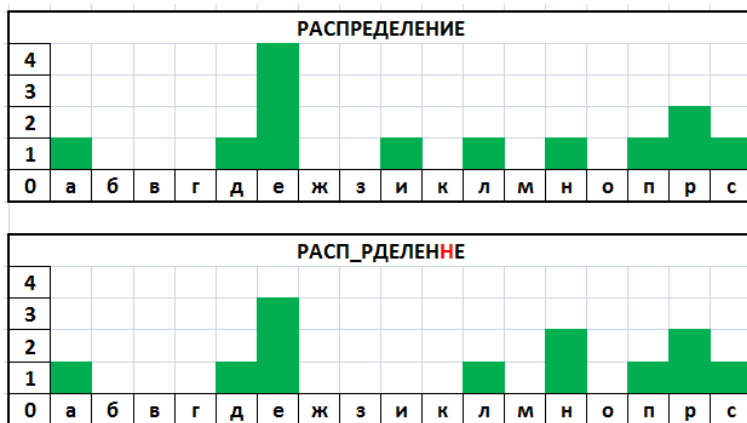


Рис. 9. Геометрический способ представления слов

Символы	Предложенный метод OCR	Предложенный метод OCR + коррекция
Общее число	13 680	13 680
Распознанных (число)	13 541	13 629
Распознанных (%)	98,98	99,63

Все замены слов были выполнены корректно, что доказывает работоспособность предложенного подхода.

**Заключение.** Предложен эффективный комплекс алгоритмов оптического распознавания символов для решения задач распознавания текстов и коррекции орфографии. Особенностью алгоритма является использование для распознавания и бинарных, и полутоновых изображений текста, с последующей коррекцией результатов, что позволило достигнуть конкурентоспособной вероятности достоверного распознавания. Результаты компьютерных экспериментов подтвердили работоспособность и перспективность идей, заложенных в основу подхода. В заключение следует отметить, что предложенные методы распознавания и коррекции можно применять не только к тестовым документам, но и в других областях использования компьютерного зрения.

## СПИСОК ЛИТЕРАТУРЫ

1. Schölkopf B., Burges C. J. C., and Smola A. J. *Advances in kernel methods: Support vector learning.* – MIT Press, Cambridge, MA, 1999.
2. Левенштейн В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. Акад. наук СССР.
3. Donoho D. L., Johnstone I. M. Adapting to unknown smoothness via wavelet shrinkage // *J. Am. Stat. Assoc.* 90 (1995). – P. 1200–1224.
4. Kang H. R. *Digital color halftoning.* – SPIE Press, Bellingham, 1999.
5. Floyd R. W., Steinberg L. An adaptive algorithm for spatial grey scale // *Proc. of the Society of Information Display* 17, 75–77 (1976).
6. Lau D. L., Arce G. R., Gallagher N. C. Green-noise digital halftoning // *Proc. of the IEEE* 86 (1998) 2424–2444.
7. Форсайт Дэвид А., Жан Понс. Компьютерное зрение. Современный подход. – М.: Вильямс, 2004. – 928 с.
8. Parker J. R. *Algorithms for image processing and computer vision* (second edition). – Wiley, 2010.
9. Cipolla R., Battiato S., Giovanni M. Farinella. *Computer vision: detection, recognition and reconstruction* (Studies in computational intelligence). – Springer, 2010.
10. Dalal N., Triggs B. Histograms of oriented gradients for human detection // *CVPR proc.*, 2005.
11. Joachims T. *Learning to classify text using support vector machines: Methods, theory, and algorithms.* – Kluwer, 2002.

Статья поступила в редакцию 15.12.2011