Структурный анализ текстовых полей в системах потокового ввода оцифрованных документов*

В. В. Арлазаров, В. М. Кляцкин, О. А. Славин

Аннотация. Статья посвящена проблематике структурного анализа текста при распознавании документов-форм на примере системы потокового ввода документов Cognitive Forms 2.0. Авторы предложили универсальный подход понимания структуры текстовых строк, одинаково пригодный для неструктурированных машинописных листов и полей документов-форм, робастный к искажениям, характерным для планшетных сканеров и камер мобильных устройств. Методология данной работы основана на различных подходах анализа данных, таких как авто-классификация и кластер анализ, методах гистограммного анализа и геометрических моделях искажения изображения при сканировании документов различными устройствами. В работе описаны алгоритмы компенсации ошибок положения строк, нахождения слов и знаков препинания в тексте.

Ключевые слова: *изображение документа, поле документа, строка символов, сложноструктурированный документ, сегмент строки.*

Введение

Важнейшей функцией современных систем ввода документов в компьютер является распознавание структуры и содержимого оцифрованных документов [7].

Структурному анализу текста посвящено немало работ (см. для примера статьи [1–4], что неудивительно, поскольку анализ структуры текстовых строк и слов является неотъемлемой частью любой OCR системы (системы распознавания печатного текста). Актуальность данной тематики и непрекращающийся интерес к ней (несмотря на кажущуюся решенность описываемых задач) основаны на попытках анализа сложноструктурированных текстовых документов и форм со сложноорганизованными полями, а также новых вызовах, возникающих при обработке «классических» текстовых структур, но с использованием мобильных устройств съема информации (фотокамеры мобильных телефонов, смартфонов, планшетов и прочих гаджетов). Особенностями мобильных образов документов являются искажения специфических типов, не присутствующие при сканировании документов стационарными сканерами.

Контекстом данной задачи в рамках системы ввода документов (далее мы будем рассматривать контекст системы Cognitive Forms [7]) является стадия обработки текстовых полей после привязки документа к описанию формы. Предполагается, что в этот момент произошло выделение полей формы и формирование текстовых строк в выделенных полях. Следующей задачей анализа поля как раз и является структурный анализ текстовых строк, найденных в данном поле. В настоящей статье описаны оригинальные алгоритмы авторов, разработанные для решения следующих задач с учетом специфики системы Cognitive Forms:

- обнаружение неточностей алгоритма привязки форм в позициях текстовых строк с последующим уточнением позиций текстовых строк;
- анализ структуры текстовой информации строк поля с выделением следующих структурных элементов:
  - построение слов (частей слов в случае переносов);
  - обнаружение знаков препинания.

1. Обзор существующих методов структурного анализа текстовых полей

Приведем несколько использующихся на практике методов анализа структуры текстовых строк.

В работе [2] предложен метод выделения текстовых строк и слов, базирующийся на понятии текстового фрагмента (последовательности компонент связности, гарантированно принадлежащих одной

* Работа выполнена при поддержке РФФИ, проекты № 13–07–00935, № 13–07–12170, № 13–07–12171.

строке). Для поиска текстовых фрагментов предложен многопроходной метод кластер-анализа, последовательно выделяющий фрагменты текста, текстовые строки и слова. К достоинствам данного подхода относятся высокая скорость анализа и приемлемые результаты понимания структуры текста на простых страницах типа журнальной статьи или технического отчета. К недостаткам метода относится невозможность анализа более сложных текстовых структур, включающих таблицы, иллюстрации, математические формулы и проч.

Развитием вышеописанного подхода является работа [1], позволяющая анализировать более сложные текстовые структуры, включающие табличные формы (в том числе иерархические), включения в текст иллюстраций и картинок. В данной публикации разработана универсальная модель описания достаточно широкого класса документов и описан алгоритм анализа текстовой структуры с использованием данной модели. Авторы предложили метод выделения строк текста и нахождения слов в табличных ячейках и проиллюстрировали свой метод на исторических документах (норвежские налоговые формы XIX века). Ограничением описанного подхода является лимитированность допустимых текстовых структур, что не позволяет использовать его для анализа текста на современных формах и документах. Помимо того, описанный подход не может быть использован для изображений посредственного качества, например, полученных с камер мобильных устройств, а также не позволяет сочетать алгоритмы построения слов с предраспознаванием знаков препинания, не может компенсировать ошибки привязки текстовых строк и не позволяет учитывать контекст текстовых полей.

Другой подход к анализу структуры текстовых строк предложен коллективом авторов в работе [3], где предпринята попытка выделения текстовых строк и слов на изображениях низкого качества (с недостаточным уровнем фокусировки и/или невысоким разрешением). Авторы предлагают сочетать результаты распознавания OCR (пакета Tesseract [5]) с кластеризацией распознанных символов (предварительно отфильтровав большие склейки) с последующим анализом графа смежности для выделения слов. (Под склейками будем подразумевать совокупность символов, объединенных в одну компоненту связности в смысле 8-связности.) Достоинством данного подхода является возможность выделения строк и слов на сложных изображениях (низкое разрешение, пространственные искажения, и т. п.).

Ограничительными особенностями метода являются то, что для выделения строк используется «полноразмерное» распознавание символов. Это, помимо замедления, может приводить к потере символов и слов, отбракованных на этапе фильтрации склеек. Кроме того, метрики оценивания качества поиска слов зависят от характеристик OCR системы Tesseract, а также при кластеризации символов для построения слов не используются сведения о расположении знаков препинания и отсутствует возможность учета контекста текстовых полей.

В работе [4] предложена обобщенная модель сегментации символов, позволяющая хранить альтернативы разрезания и использовать их для последующего анализа структуры строки. Данная схема позволяет уменьшить зависимость качества выделения слов от ошибок сегментации символов. На первом этапе триплеты рамок символов (альтернативы сегментации) кластеризуются в «квази-строки». После чего строится граф соседства альтернатив сегментации. В качестве предобработки используется распознавание отдельных символов с помощью некоторой OCR системы, обученной на синтетических фонтах. При построении графа альтернативы с низкими оценками распознавания игнорируются, чтобы по возможности удалить шумовые компоненты и ошибки сегментации. Далее методом динамического программирования выбирается оптимальный путь на ориентированном графе смежности альтернатив. После этого на полученной последовательности рамок предполагается построение слов данной строки. К достоинствам данного метода следует отнести «мягкую» схему сегментации символов, позволяющую в комбинации с методом динамического программирования достичь неплохих результатов выделения рамок отдельных символов и их группировки в строки. К определенным ограничениям метода относятся необходимость использования OCR системы как стадии предобработки для выделения строк, а также игнорирование знания о знаках препинания при построении строк и слов, а также ориентированность системы на обычные страницы текста (без возможности настройки на специфические особенности текстовых полей).

2. Постановка задачи структурного анализа текста

Целью настоящей работы является разработка универсального метода понимания структуры текстовых строк, позволяющего анализировать текстовую информацию, как в контексте простых текстовых страниц, так и в контексте полей структурированных форм. Метод должен быть робастным как по отношению к искажениям изображений, специфичных для сканеров, так и для пространственных искажений, специфичных для камер мобильных устройств.

Входной информацией метода является оригинальный образ документа, его бинаризованная версия, а также описание поля формы, для которого требуется проанализировать структуру текстовой информации. Описание поля включает априорную информацию о поле (такую как его «идеальная» позиция на форме, алфавит, возможные сведения о шрифтах, числе слов и проч.), а также информацию о поле, полученную в процессе его обработки (положение поля после привязки формы, наклоны текстовых строк поля и глобальный наклон отсканированного листа).

Рассмотрим постановку задачи: используя исходное цветное/серое изображение I_{orig} и набор бинарных изображений $I_1, \ldots, I_k\}$, полученных с использованием различных методов бинаризации, необходимо выделить и проанализировать структуру текстовой информации поля, задаваемого своей позицией на форме (R_{ideal}) и позицией на образе обрабатываемого документа (R_{real}) При этом предполагаются также заданными начальные оценки позиций текстовых строк $\{S_1, \ldots, S_N\}$ и их углы перекоса $\{\phi_1, \ldots, \phi_N\}$, где N — число строк, найденных в данном поле.

Требуется уточнить позиции текста, сформировав набор оценок рамок строк поля $\{\tilde{S}_1, \ldots, \tilde{S}_N\}$ и произвести анализ текстовых строк, выделив компоненты связности, принадлежащие полю $CC_1, \ldots, CC_n$, сгруппировав их по строкам $s_i = CC^i_1, \ldots, CC^i_{k_i}$, где CC^i_j — j-ая компонента i-ой строки.

При этом необходимо классифицировать компоненты строки на символьные, шумовые и компоненты, принадлежащие знакам препинания.

В каждой строке необходимо выделить текстовые слова Алгоритм поиска слов должен учитывать априорную информацию поля при ее наличии (опираться на алфавит распознавания поля, учитывать размерностные характеристики шрифтов поля). Алгоритм должен сформировать слова как последовательности символов строки

$$W = \{w_j,\ j = \overline{1, L}\},$$

где L – общее число слов, найденных в строке s_i, а каждое слово w_j состоит из принадлежащих ему компонент связности $w_j = \{CC \in w_j\}$ и оценки качества его выделения.

Помимо того, должны быть сформированы интегральные оценки сегментации строк на слова и вероятности выделения строк (т. е. вероятности того, что сформированные группы компонент связности действительно являются текстовыми строками).

После решения поставленной задачи у OCR появится возможность обрабатывать сгруппированные в строки и слова, и частично классифицированные символы. Для таких групп символов принимается гипотеза об однородности символов, входящих в группу. Например, в печатных документах невозможно (или маловероятно) печать в одном слове символов различных гарнитур, кегля, модификации (жирный, серифный шрифты). Это позволяет редуцировать алфавит распознавания и существенно улучшить качество сегментации границ и качество классификации отдельных символов

3. Обнаружение дефектов привязки и уточнение позиций текстовых строк

Обнаружение ошибок местоположения текстовой строки основано на поиске символов, «почти» принадлежащих данной строке, но выступающих за ее границу.

Для понижения уровня ложных тревог детектора анализируются лишь компоненты связности определенного диапазона размеров с достаточной степенью попадания в строку. Также игнорируются символы, выступающие более чем за 2 границы строки, поскольку для таких компонент вероятность их принадлежности к строке невелика.

Если найдена хоть одна выступающая компонента, удовлетворяющая всем вышеперечисленным условиям, производится независимое уточнение каждой границы строки.

Ниже приведена иллюстрация работы алгоритма. Детектор обнаружил дефект позиции нижней строки — подсвеченные серым символы «у» выступают за нижнюю границу своей строки — см. на рис. 1.

Рис. 1. Пример с выступающими компонентами.

После корректировки нижней границы строки все символы полностью попадают в ее зону — см. на рис. 2.

Рис. 2. Корректировка границы для примера с рис. 1.

4. Алгоритм анализа структуры текстовых строк полей

Данный алгоритм предназначен для исследования структуры текстовых строк в составе привязан-

ных полей на образах документов. Приведем основные стадии алгоритма структурного анализа строки:

1. *Выбор компонент связности строки и классификация компонент строки* на основные и дополнительные (включает в себя следующие элементы — шумы, отколовшиеся кусочки букв, знаки препинания).
2. *Переход в идеальные координаты и формирование размерностных фонтов*, *статистик оценка базовых линеек*. Для оценивания верхней и нижней границ строки используются медианные оценки по верхним и нижним границам символов.
3. *Разбиение на слова и выделение знаков препинания* на нижней линейке — см. примеры на рис. 3. Каждое слово строки выделено рамкой, зарегистрированные знаки препинания — залиты серым цветом:

Рис. 3. Пример разбиения на слова и знаки препинания

5. Описание алгоритма построения слов

Данный алгоритм предназначен для выделения слов в текстовой строке, а также в более общем случае для выделения слов в произвольном текстовом фрагменте (многострочное поле, параграф, колонка, текстовая страница).

В основе данного алгоритма лежит анализ интегрального распределения вероятностей внутрисловных и межсловных интервалов символов строки, сходный с алгоритмом Отсу [6] поиска пиков на бимодальном распределении, но имеющий свои особенности, учитывающие особенности распределений, возникающих в данной задаче (возможность отсутствия двух пиков, возможность сильного перекрытия распределений внутрисловных и межсловных интервалов, значительная неравнозначность пиков и многое другое).

На первой стадии алгоритма используются лишь компоненты достаточного размера, чтобы избежать искажения статистик от шумовых пятен, знаков препинания и развалленных букв. Все компоненты связности каждой строки упорядочиваются слева направо в порядке их следования в строке. Затем вычисляется глобальный (уровня всего текстового фрагмента) порог межсимвольного интервала для деления на слова, для чего формируется массив межсимвольных интервалов, объединенный по всем строкам фрагмента. Интервал между соседними компонентами включаем в массив, если он положительный (компоненты не перекрываются по вертикали) и не более двух третей стандартной оценки ширины символа фрагмента (эвристическое привило для исключения межсловных интервалов). По сформированному массиву интервалов формируется усредненная глобальная оценка внутрисловного интервала фрагмента Δ_{col}. Далее оцениваем глобальный межсловный интервал, для чего вычисляем гистограмму Hist межсимвольных интервалов, игнорируя перекрывающиеся символы и интервалы, более чем в три раза превышающие стандартную ширину символа фрагмента (чтобы исключить влияние выбросов типа интервалы между словами из разных строк при ошибочном слиянии строк и просто встречающихся экстра-широких межсловных интервалов). Сдвигаем правую границу гистограммы к самой правой непустой ячейке. Пытаемся найти правую границу пика гистограммы, соответствующего внутрисловным интервалам, двигаясь вправо от усредненной глобальной оценки внутрисловного интервала Δ_{col} до тех пор, пока значение гистограммы не станет ниже некоторой пороговой доли T_R от значения в ячейке Δ_{col} : $\text{Hist}(\text{pos}) < T_R\text{Hist}(\Delta_{\text{col}})$.

После этого сдвигаем оценку Δ_{col} к текущей позиции pos гистограммы (правому склону пика).

Теперь попытаемся найти раздел между пиками, соответствующими внутрисловному и межсловному интервалам. Для этого находим минимальное значение гистограммы правее скорректированной оценки Δ_{col}.

1. Если найденный минимум меньше некоторого порога T_1 и позиция минимума отстоит не менее чем на наперед заданное число ячеек $N_{\min}$ от правой границы, проводим следующий анализ: если хотя бы в одной из соседних ячейках значение превосходит T_1, бракуем найденную минимальную позицию, так как есть опасность, что это локальная впадина на правом склоне зоны внутрисловных интервалов. Отступаем на ячейку вправо от найденного минимума и находим следующий минимум. Если значение найденного минимума или в одной из ближайших соседей больше T_1 или его позиция отстоит от правой границы ближе чем на $N_{\min}$ ячеек, фиксируем ситуацию отсутствия впадины между пиками и переходим на п. 2. Иначе доходим до правой границы впадины и сравниваем интегральное значение распределения до нее и после нее и, если это отношение превосходит некоторое пороговое значение $R_{\min}$

это означает, что число межсловных интервалов намного меньше, нежели внутрисловных, что, в свою очередь, может означать ошибочное нахождение зоны раздела. (Отметим, что $R_{\min}$, представляет собой ограничение на среднюю длину слова в текстовом сегменте.) В этом случае бракуем найденную границу и переходим на п. 2. Иначе пытаемся вычислить точное значение центра пика межсловных интервалов, используя следующую процедуру:

a. Огрубляем гистограмму (коэффициент кратности зависит от оценки ширины символа фрагмента). Двигаясь слева направо по огрубленной гистограмме, пытаемся найти первую ячейку со значением, превосходящим некоторую фиксированную долю от общего числа интервалов в межинтервальной зоне. Как только такая ячейка найдена, вычисляем точное значение центра пика Δ^{w}_{col} как середину сканирующего окна. Иначе переходим на п. 1б;

b. Повторяем шаг 1a, но со сканирующим окном = 2 и другим значением порога числа интервалов в нем. При этом после нахождения первой ячейки, удовлетворяющей условию, поиск не прерываем, а выставляем порог значению этой ячейки и движемся правее до нахождения самой многочисленной ячейки.

2. Если не удалось найти межсловный интервал и/или впадину после внутрисловного интервала, пытаемся найти «стандартное» значение межсловного интервала, двигаясь в обратном направлении по гистограмме (справа налево, начиная от текущей позиции с небольшим сдвигом вправо) узким скользящим окном в поисках достаточно большого значения в окне, на достаточном удалении от правой границы внутрисловного интервала. Значение порога и сама процедура полностью аналогичны используемым на шаге 1b.

3. Далее переходим к уточнению оценок межсловных интервалов на уровне отдельных строк (в случае текстовых фрагментов и многострочных полей):

a. Формируем массив межсимольных интервалов строки. При этом правило его заполнения немного отличается от случая целого текстового фрагмента — а именно для пересекающихся рамок символов в случае, если оба символа достаточно высоки и их вертикальное перекрытие тоже достаточно высоко, помещаем их интервал, но полагаем его значение равным нулю.

b. Если накопленное число интервалов невелико, полагаем оценку $\Delta_{\text{str}} = \Delta_{\text{col}}$, иначе полагаем эту оценку равной медиане накопленной выборки интервалов.

c. Далее оцениваем локальный (уровня строки) порог межсимвольных интервалов для построения слов, используя сформированную выше выборку интервалов. Вначале вычисляем число интервалов N_{Less} вблизи оценки Δ_{str} Далее выполняем следующие шаги:

i. Если разница между значениями Δ^{w}_{col} и Δ_{str} не превосходит заранее заданного порога T_{col}, переходим на п. 3.c.ii. Иначе вычисляем следующую величину:

$$g_{\text{add}} = \min\left(\max\left(4,\ \sigma\right),\ \Delta^{w}_{\text{col}} - \Delta_{\text{str}}\right)$$

Затем, если число внутрисловных интервалов N_{Less} составляет не менее заданной доли $r_{\min}$ от всех интервалов, вычисляем порог T_W по следующей формуле:

$$T_W = \Delta_{\text{str}} + g_{\text{add}},$$

иначе применяем формулу

$$T_W = \Delta_{\text{str}} + \min\left(0{,}3(\Delta^{w}_{\text{col}} - \Delta_{\text{str}}),\ 8\right).$$

ii. Если число внутрисловных интервалов N_{Less} составляет не менее доли $r_{\min}$ от всех интервалов и с. к. о. σ выборки интервалов не более порогового значения $\sigma_{\min}$, вычисляем порог T_W как следующую сумму:

$$T_W = \Delta_{\text{str}} + \max(\sigma, 2\sigma_{\min}).$$

iii. Иначе алгоритм локального оценивания возвращает отказ от обработки.

4. В случае, если алгоритм локального оценивания отказался от обработки, порог T_W формируется по одному из следующих правил:

a. Если алгоритм оценивания межсловного интервала Δ^{w}_{col} также отказался от обработки, используем следующую формулу:

$$T_W = \Delta_{\text{str}} + \text{coeff} \cdot \sigma.$$

b. Иначе вычисляем следующую оценку:

$$T_W = \Delta_{\text{str}} + \text{coeff}_1 \cdot \left(\Delta^{w}_{\text{col}} - \Delta_{\text{str}}\right).$$

5. Формируем разбиение на слова, используя сформированный порог T_W. Для повышения устойчивости алгоритма к случаю распавшихся букв

на данном этапе каждый межбуквенный интервал оцениваем как минимум интервалов от текущей компоненты связности до двух-трех ее соседей слева, перекрывающихся по вертикали.

Отметим, что введенные выше пороговые значения не ограничивают общности алгоритма, а напротив способствуют гибкости его возможной настройки для обработки текстов разной природы. Проведенные авторами исследования показали, что некоторый выбранный набор численных значений параметров алгоритма покрывает широкий спектр типов документов и условий их сканирования, что свидетельствует о робастности предложенного алгоритма к внешним условиям.

После построения «костяков» слов распределяем по ним отколовшихся «отщепенцев» и знаки препинания. Таким образом, получаем окончательные слова как списки попавших в них компонент связности. Окончательно формируем описание структуры проанализированной строки, ее слов, знаков препинания, найденных шумов.

6. Описание алгоритма поиска знаков препинания

Данный алгоритм предназначен для обнаружения знаков препинания в текстовых строках привязанных полей формы. Алгоритм использует информацию о размерностных характеристиках сформированных компонент связности строки, а также их позициях на линейках.

Для обнаружения точек и запятых строятся медианные оценки верхней и нижней линеек строки, а затем кандидаты на знаки препинания (подходящие по размерам и позиции), соотносятся с соседними символами. Для обнаружения многокомпонентных знаков препинания и знаков препинания, лежащих на верхней линейке строки, применяются специализированные процедуры, основанные на предраспознавании и анализе контекста поля.

7. Описание методологии тестирования системы и численные результаты экспериментов

Для тестирования разработанной совокупности алгоритмов была разработана и использовалась программа пакетной обработки, позволяющая обрабатывать большие тестовые наборы в многопоточном режиме, а также накапливать и обрабатывать статистические оценки работы предложенных алгоритмов.

Для анализа качественных характеристик алгоритмов использовались следующие тестовые наборы:

- товарные накладные;
- заявления на выдачу паспортов;
- внутренние и международные российские паспорта.

При тестировании алгоритма уточнения позиций текста оценивались следующие характеристики: уровень ложной тревоги (ложное обнаружение события типа дефекта рамки строки), точность повторной классификации символов по строкам после уточнения их позиций. По результатам тестирования на полноразмерных документах (стенды товарных накладных и заявлений на выдачу паспортов) выявлен низкий уровень ложной тревоги (менее 0.1 %) и, соответственно, низкий уровень ошибок повторной классификации символов по строкам, оцененный в 0.15 %. На стенде паспортов вообще не было зарегистрировано ложных обнаружений событий типа дефекта рамки строки.

Тестирование анализа текста проводилось в двух режимах – «вслепую», когда вся априорная информация о контексте поля не учитывалась, и с полным учетом априорной и накопленной информации о структуре поля. Для оценивания точности алгоритма выделения слов использовалась база «идеальных» значений полей, по которой вычислялись числа слов в строках по каждому полю, и данные числа сравнивались с реальным числом слов, выделенных в каждой строке.

По результатам тестирования было выявлено, что учет априорной информации существенно увеличивает точность разбиения на слова (оценки разнятся от набора к набору, и например, на наборе товарных накладных ошибка пересегментации была уменьшена на 30 %). Усредненная по всем наборам ошибка «пересегментации» (ложного разбиения слова на две составляющих) оценивается в 5 %, ошибка «недосегментации» оценивается в 6–7 %. Отметим, что наибольший вклад в ошибку «недосегментации» составляют чисто числовые поля и поля, преимущественно состоящие из чисел (за счет трудности учета знаков препинания внутри слов типа «сумма»). Также отметим, что приведенные 95-процентные оценки точности алгоритма обнаружения слов уже являются приемлемыми, однако они могут быть существенно улучшены применением других механизмов распознавания и контекстного анализа.

8. Выводы

Таким образом, в настоящей работе предложены алгоритмы анализа структур текстовых строк,

которые могут быть использованы как для универсальных OCR при распознавании страниц машинописного текста, так и в системах потокового ввода при обработке структурированных форм. Были решены задачи уточнения позиций строк, разбиения строк на слова и поиска знаков препинания. Также были сформированы надежностные оценки сегментации строки на слова, выделения знаков препинания и шумовых компонент и строк. На основе вышеописанных алгоритмов была разработана на языке C++ (ANSI) и внедрена в систему потокового ввода документов Cognitive Forms 2.0 библиотека Layout.

В результате тестирования библиотеки Layout были подтверждены высокая робастность методов структурного анализа по отношению к методам получения изображений (библиотека хорошо работает как на изображениях с планшетных сканеров, так и на изображениях, полученных с мобильных устройств), высокая точность алгоритмов анализа. Например, точность алгоритма поиска слов была поднята на 30–40 % по сравнению с методами, использованными в работах [1] и [2]. Кроме того, библиотека Layout не использует OCR для построения строк и слов, что выгодно отличает ее от подходов, развитых в работах [3] и [4]. Отметим еще несколько преимуществ данной библиотеки — она гибко сочетает алгоритмы поиска слов и знаков препинания (т. е. при наличии априорной информаци о содержимом поля позволяет использовать либо игнорировать информацию о знаках препинания при поиске слов, а также использовать разные правила комбинирования букв, цифр и знаков препинания в слова) для более глубокого понимания структуры текста без его распознавания, а также органично учитывает свойства полей (например, в случае заданного алфавита поля переключаться на разные режимы поиска слов или вообще трактовать все поле как одно слово).

Литература

1. *Kliatskine V.*, *Shchepin E.*, *Thorvaldsen G*, *Zingerman K.*, *Lazarev V.* A Structural Method for the Recognition of Complex Historical Tables // History & Computing, Edinburg University Press 1997, vol. 9, No. 3. PP 58–77.
2. *Klyahzkin V.*, *Shchepin E.*, *Zingerman K.* Application of hierarchical methods of cluster analysis to the printed text structure recognition // Shape, Structure, and Pattern Recognition Nahariya, Israel, October 1994, Dov Dori and Alfred Bruckstein, Eds., World Scientific, 1995, Singapore, New Jersey, London, Hong Kong (SSPR'94), PP 333–342
3. *Wang H.*, *Landa Y.*, *Fallon M.*, *and Teller S.* Computer Science and Artificial Intelligence Laboratory Massachusetts Institute of Technology USA - Spatially Prioritized and Persistent Text Detection and Decoding // ICDAR 2013 12^{th} International Conference on document analysis and recognition, Washington, DC, August, 2013 pp. 7–12.
4. *Neumann L.*, *Matas J.* Center for Machine Perception, Department of Cybernetics Czech Technical University, Prague, Czech Republic - On Combining Multiple Segmentations in Scene Text Recognition // ICDAR 2013 12^{th} International Conference on document analysis and recognition, Washington, DC, August, 2013 pp. 523–527
5. *Smith R.*. An overview of the Tesseract OCR engine. In Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR), pp. 629633, 2007
6. *Otsu N.* A threshold selection method from gray-level histograms. IEEE Trans. Sys., Man., Cyber. 1979. 9 (1): 62–66
7. *Арлазаров В. В.*, *Постников В. В.*, *Шоломов Д. Л.* Cognitive Forms - система массового ввода структурированных документов // Сборник трудов Института системного анализа РАН «Управление информационными потоками». М.: URSS, 2002. С. 35–46.

Арлазаров Владимир Викторович. Зав. лабораторией ИСА РАН. К. т. н. Окончил в 1999 г. МИСиС. Количество печатных работ: 13. Область научных интересов: распознавание образов, обработка изображений, системы массового обслуживания. E-mail: vva777@gmail.com

Кляцкин Виталий Менделевич. С. н. с. ЗАО «Интеллектуальные технологии». К. т. н. Окончил в 1985 г. Тульский ГУ. Количество печатных работ: 20. Область научных интересов: распознавание образов, обработка изображений. E-mail: Kliatskine@gmail.com

Славин Олег Анатольевич. Зав. лабораторией ИСА РАН. Д. т. н. Окончил в 1988 г. МИРЭиА. Количество печатных работ: 69 (в т. ч. 1 монография). Область научных интересов: распознавание образов, информационные системы. E-mail: oslavin@cs.isa.ru