

АЛГОРИТМЫ РАСПОЗНАВАНИЯ РУКОПИСНЫХ СИМВОЛОВ НА ОСНОВЕ ПОСТРОЕНИЯ СТРУКТУРНЫХ МОДЕЛЕЙ

П.А. Хаустов¹

¹ Томский политехнический университет
(национальный исследовательский университет) (ТПУ), Томск, Россия

Аннотация

Статья посвящена разработке алгоритмов распознавания рукописных символов на основе построения структурных моделей, существенным преимуществом которых является возможность работы в условиях малого количества эталонных изображений. В том числе был предложен подход к скелетизации бинарного представления изображения символа, основанный на совместном применении алгоритмов Зонга–Суня и Ву–Цая. Эффективность данного подхода к утоньшению бинарного представления символа была подтверждена результатами проведённых экспериментов.

Работа содержит подробное описание всех стадий алгоритма построения структурных моделей и описание подходов к оценке степени их схожести. Приведены результаты апробации предложенных алгоритмов. Выполнено сравнение полученных результатов с результатами аналогов, способных функционировать в условиях малого количества эталонных изображений.

Ключевые слова: распознавание символов, структурные составляющие, структурные модели, компьютерное зрение, скелетизация, бинаризация.

Цитирование: Хаустов, П.А. Алгоритмы распознавания рукописных символов на основе построения структурных моделей / П.А. Хаустов // Компьютерная оптика. – 2017. – Т. 41, № 1. – С. 67-78. – DOI: 10.18287/2412-6179-2017-41-1-67-78.

Введение

Теория компьютерного зрения на сегодняшний день является одной из наиболее востребованных областей исследований. Решения задач этой области находят применение во многих сферах человеческой деятельности, включая образование и медицину [1].

Распознавание символов – одна из наиболее важных задач теории компьютерного зрения. Особенно востребованы решения такой задачи при переводе больших архивов из бумажного в электронный вид хранения информации [2]. Задача распознавания рукописных символов является трудно формализуемой, а точность существующих методов её решения несколько ниже, чем точность аналогичных методов, применяемых для печатных символов [3]. Методы и алгоритмы решения задачи распознавания рукописных символов используются при автоматической обработке анкет и проверке тестовых заданий. Даже если для заполнения документа предоставляется набор образцов правильного начертания каждого из символов, особенности почерка и различия в способах начертания различных людей приведут к тому, что форма графического представления одних и тех же символов этими людьми будет существенно различаться.

Большинство методов, которые применяются для решения задачи распознавания рукописных символов, схожи с методами, которые используются для других задач классификации изображений [4]. Как правило, такие методы предполагают использование искусственных нейронных сетей [5] или аналогичных признаков классификаторов, которые не позволяют отследить логику процесса классификации [6, 7].

Особый интерес представляет группа методов, основанных на извлечении структурных составляющих из исходных изображений символов. В таких методах сим-

волы представляются не в виде вектора некоторых абстрактных признаков, а в виде набора определённых элементов графического начертания символа, например, в виде ключевых точек и связывающих их геометрических примитивов. Полученные представления используются для построения структурных моделей, которые, в свою очередь, используются непосредственно в алгоритмах сравнения [8]. Структурная модель символа может быть получена путём дополнения топологической модели геометрическими характеристиками.

Преимуществами таких методов над группой методов с использованием признаков классификаторов является лучшая наглядность процесса распознавания для пользователя. При разработке таких методов существенно проще реализовать построение аргументированного отчёта о каждом из этапов построения структурной модели символа и последующего процесса сравнения.

В этой связи разработка алгоритмов, основанных на построении структурных моделей, является достаточно актуальной, а её решения найдут широкое применение в медицинских и образовательных учреждениях, где существует необходимость автоматической и полуавтоматической обработки большого количества бланков.

1. Бинаризация и скелетизация изображения

Для выполнения распознавания символа необходимо предварительно разделить все пиксели этого изображения на пиксели, которые принадлежат графическому представлению символа, и оставшиеся пиксели.

Можно считать, что на изображении в градациях серого пиксели, принадлежащие начертанию символа, существенно отличаются по среднему уровню яркости от пикселей фона. Для определённости будем полагать, что более светлые пиксели принадлежат

фону. В таком случае для разделения пикселей на два описанных класса обычно используется некоторое пороговое значение яркости T .

Для определения оптимального порога бинаризации T предложено огромное количество различных алгоритмов. Как правило, такие методы опираются на использование локальной или глобальной гистограммы изображения. Одним из наиболее широко используемых подходов является подход, предложенный Оцу; преимуществами данного алгоритма являются понятный статистический смысл и низкая вычислительная сложность [9].

Скелетизация бинарного представления символа – процесс, в ходе которого выполняется максимальное утоньшение всех линий на бинарном изображении данного символа, результатом чего является скелет его исходного изображения, в котором все линии имеют толщину в один пиксель. Под термином «скелет бинарного изображения символа» понимается подмножество его пикселей, равноудалённых от его границ, образующих объект единичной толщины, топологически эквивалентный данному символу.

В задаче распознавания символов очень важно, чтобы в процессе скелетизации не происходило потери информации о ключевых элементах графического начертания исходного символа. С другой стороны, для методов, опирающихся на использование скелетизированного изображения, важно, чтобы ни один из пикселей, который можно перекрасить, не нарушая топологии графического представления, не был пропущен в ходе работы алгоритма. Описанные случаи зачастую приводят к появлению ошибок в ходе последующей классификации, особенно при использовании алгоритмов, выделяющих структурные составляющие символов.

В исследованиях A.R. Widiarti приведён сравнительный анализ алгоритмов построения скелетов бинаризованных растровых изображений [10]. В приведённом исследовании отмечается, что алгоритм Зонга–Суня [11] показал наиболее высокое быстродействие и низкую вероятность удаления важных элементов графического начертания в ходе процесса скелетизации. Однако на результирующем изображении достаточно часто остаются пиксели, которые можно перекрасить без нарушения топологии исходного графического начертания. На рис. 1 показан пример такого результата применения алгоритма.

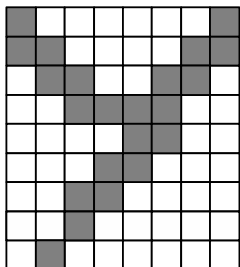


Рис. 1. Пример не до конца скелетизированного участка изображения символа

Такой подход может оказать существенное влияние на поведение алгоритмов обхода бинарного изобра-

жения, направленных на выделение структурных составляющих. Как показывают результаты исследования, все рассмотренные алгоритмы на определённых изображениях допускают наличие не до конца скелетизированных участков.

Для устранения областей изображения, где скелетизация была выполнена не полностью, предложено дополнительно использовать алгоритм скелетизации Ву–Цая [12]. После применения такого подхода нежелательные участки на результирующем изображении устраняются.

Следует отметить, что самостоятельное применение алгоритма Ву–Цая без предварительной обработки алгоритмом Зонга–Суня оказывается нецелесообразным. Алгоритм Ву–Цая достаточно часто нарушает исходную топологию объекта, удаляя небольшие элементы графического представления символа. Можно сделать вывод, что использование лишь одного из двух рассмотренных алгоритмов скелетизации приведёт к потере информации о топологии символа или к неоднозначности при попытке извлечения структурных составляющих. В свою очередь, использование сначала алгоритма Зонга–Суня, а затем алгоритма Ву–Цая приводит к устранению обоих типов недостатков.

Однако существуют изображения, при обработке которых даже при таком подходе удаляются важные элементы графического представления символа. Пример такого случая изображен на рис. 2.

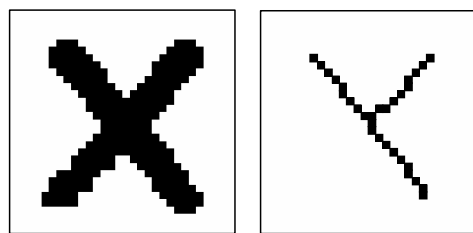


Рис. 2. Пример некорректной скелетизации последовательным применением алгоритмов Зонга–Суня и Ву–Цая

Причиной этого является алгоритм Зонга–Суня, который тоже иногда допускает удаление элементов, важных с точки зрения топологии символа. Так происходит с приведенным графическим начертанием символа «X», которое содержит плоское завершение нижнего левого элемента, представленное последовательностью горизонтально расположенных пикселей. Алгоритмом оно воспринимается как локальное утолщение и постепенно удаляется. Одним из способов равномерного утолщения бинарного изображения является дилатация – одна из операций морфологической обработки изображений. Однако применение такой операции может привести к соединению двух близко расположенных друг к другу элементов, и тем самым нарушить исходную топологию объекта.

Была предложена собственная операция, имеющая сходство с морфологической операцией дилатации. Для каждого белого пиксела рассмотрим пиксели, граничащие с ним по стороне. Если цвета четырёх соседних

пикселей не образуют одну из четырех комбинаций, приведённых на рис. 3, то пиксел становится черным.

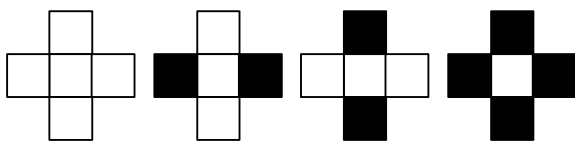


Рис. 3. Комбинации соседних пикселей, не требующие перекраски центрального пиксела в чёрный цвет

Чтобы избежать нарушений исходной топологии представления символа, после применения данной операции осуществляется проверка на то, оказались ли два чёрных пикселя в одной компоненте связности, если ранее они принадлежали разным компонентам. Такая проверка может быть выполнена с помощью одного дополнительного запуска алгоритма обхода в ширину, время работы которого линейно зависит от количества пикселей на изображении. Если такие пиксели обнаружены, то результат применения операции не используется.

Предобработка с использованием такой операции позволяет избежать плоских окончаний элементов бинарных изображений, не требующих удалений. Для ранее приведённого примера символа применение такой операции приводит к существенному улучшению результата.

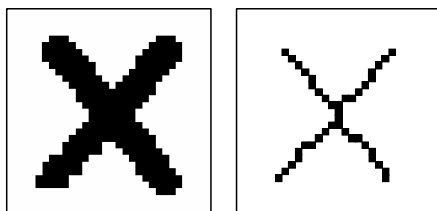


Рис. 4. Результат выполнения скелетизации после применения предложенной операции

Таким образом, предложенный подход к выполнению скелетизации состоит из трёх последовательных этапов:

1. Предварительная обработка бинарного изображения с целью устранения плоских окончаний элементов графического представления.
2. Применение алгоритма скелетизации Зонга–Суна.
3. Применение алгоритма Ву–Цая для устранения участков с незаконченной скелетизацией на полученном изображении.

2. Предложенный метод выделения структурных составляющих

Выполнить выделение структурных составляющих на скелетизированном бинарном изображении символа существенно проще, чем на произвольном бинарном изображении. Для такого рода изображений гораздо проще сформулировать алгоритмы, которые позволят извлечь такого рода составляющие.

Выделение таких важных точек даже для скелетизированных бинарных изображений остаётся нетривиальной задачей. Еще более нетривиальной задачей является характеристика соединяющих эти точки элементов.

Обозначим термины, которые будут использоваться для описания извлекаемых структурных составля-

ющих представления символа, а также для описания алгоритма их получения.

Ключевые пиксели – чёрные пиксели скелетизированного бинарного изображения символа, соответствующие ключевым точкам или изгибам его структурной модели.

Соединяющее ребро – структурная составляющая представления символа, описывающая последовательность чёрных пикселей на скелетизированном бинарном изображении символа, соединяющую два ключевых пикселя.

Ключевая точка – структурная составляющая представления символа, которая является местом стыка одного, трёх или более соединяющих ребер. Также ключевой точкой можно назвать место стыка двух соединяющих ребер, направляющие вектора которых при выборе в качестве начального положения данной ключевой точки расположены под углом менее 120 градусов.

Направляющий вектор некоторого соединяющего ребра может быть вычислен с использованием следующего принципа. Рассмотрим соединяющее ребро между двумя ключевыми пикселями как упорядоченную в порядке удаления от начального положения последовательность пикселей. В качестве начального положения следует выбрать один из инцидентных ключевых пикселей. Поочередно проведём вектора из начального ключевого пикселя в каждый из пикселей пути. Последовательно просуммируем в порядке удаления от начального ключевого пикселя все вектора с угасающими коэффициентами: например, первый – с множителем 1; второй – с множителем 0,5; третий – с множителем 0,25 и так далее.

Изгиб (точка изгиба) – структурная составляющая представления символа, которая не может быть отнесена к ключевым точкам, но линия, соединяющая два ключевых пикселя и проходящая через этот пиксел, существенно изменяет направление на участке графического представления символа, соответствующем этому пикселу. Фактически изгиб является стыком двух соединяющих ребер, направляющие вектора которых при выборе в качестве начального положения данного изгиба расположены под углом не менее 120 градусов.

Композитное ребро – последовательность соединяющих ребер, начинающаяся и заканчивающаяся в ключевых точках, которая не содержит ни одной ключевой точки, кроме начальной и конечной.

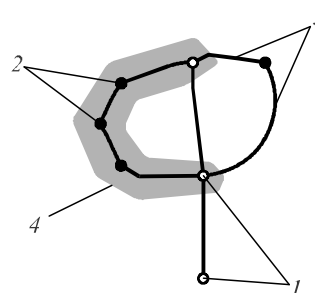


Рис. 5. Основные структурные составляющие: ключевые точки (1), изгибы (2), соединяющие ребра (3) и композитные ребра (4)

Выделение структурных составляющих рационально начать с выделения ключевых пикселей. В первую очередь к таким пикселям можно отнести все пиксели, количество соседей у которых не равно двум – изолированные пиксели, конечные пиксели и стыки трёх и более элементов.

Сложнее анализировать пиксел, который имеет два соседа. В таком случае он может являться частью соединяющего ребра или точкой сочленения двух элементов графического начертания. Можно выделить первую группу шаблонов соседних пикселей, которая всегда соответствует ключевому пикселу или изгибу (рис. 6), назовём эту группу шаблонов α -группа.

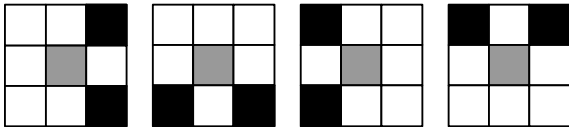


Рис. 6. α -группа шаблонов соседних пикселей

Для некоторых шаблонов соседних пикселей недостаточно информации лишь об их взаимном расположении. Необходимо также учесть, как расположены соседи непосредственных соседей центрального пикселя. Выделим некоторую группу таких шаблонов соседних пикселей, обозначив её как β -группу (рис. 7).

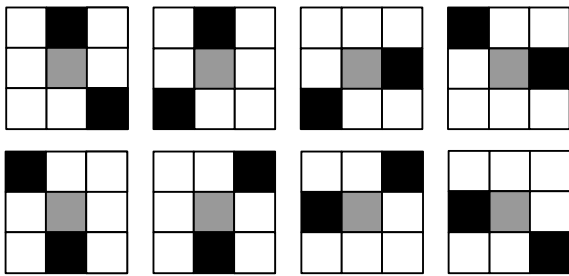


Рис. 7. β -группа шаблонов соседних пикселей

Как можно заметить, такие участки бинарного изображения могут соответствовать как участку соединяющего ребра, так и точке стыка нескольких соединительных ребер. Примеры случаев, когда шаблон β -группы соответствует ключевой точке или точке изгиба, показаны на рис. 8.

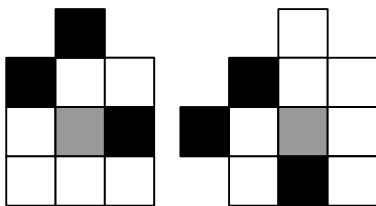


Рис. 8. Примеры случаев, когда шаблон β -группы не является частью соединяющего ребра

Один соседний пиксел одного из соседей центрального пикселя является ключевым для определения типа участка бинарного изображения. Однако можно выделить ещё одну группу шаблонов соседних пикселей, которая, как и β -группа, характеризуется зависимостью не только от непосредственных соседей центрального пикселя. Назовём такую группу шаблонов γ -группой. Формально такая группа состоит лишь из двух шаблонов (рис. 9).

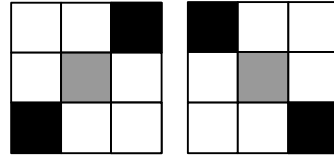


Рис. 9. γ -группа шаблонов соседних пикселей

Как показали эксперименты, достаточно часто возникает необходимость в обнаружении случаев, когда подобные шаблоны соответствуют изгибам и даже ключевым пикселям. Определить такие случаи можно, если рассмотреть сразу два пикселя, соседних с центральным. На рис. 10 показаны примеры подобных случаев.

В таких случаях шаблон соответствует достаточно резкой двукратной смене направления соединяющего ребра. В таком случае, как правило, возникает необходимость разбиения представления такого ребра в данной точке на два геометрических примитива.

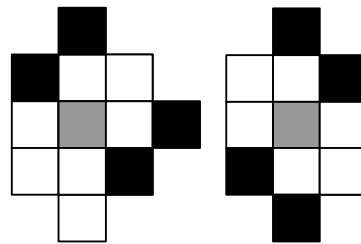


Рис. 10. Примеры случаев, когда шаблон γ -группы не является частью соединяющего ребра

Таким образом, были выделены три группы шаблонов расположения соседних пикселей: α -группа, β -группа и γ -группа. Пиксели, принадлежащие им, могут соответствовать ключевым пикселям. Стоит отметить высокое быстродействие процедуры сравнения окрестности пикселя с каждым из шаблонов. При эффективной реализации отдельное сравнение может быть выполнено с использованием единственной операции побитового «И». Такой подход может быть основан на хранении информации о цветах соседних пикселей в битах специально отведённого целочисленного значения. Операция проверки цветов пикселей, граничащих с непосредственными соседями, заключается в восьмикратном использовании процедуры сравнения окрестности пикселя. Стоит отметить, что общая сложность алгоритма обнаружения точек, соответствующих представленным шаблонам, составляет $O(P)$, где P – общее количество пикселей на изображении.

3. Алгоритм разделения ключевых пикселей на ключевые точки и изгибы

Для решения задачи разделения ключевых точек и изгибов недостаточно рассматривать ограниченные окрестности соответствующих пикселей. Требуется анализ всего бинарного изображения. В частности, необходимо анализировать маршруты между найденными кандидатами в ключевые пиксели по чёрным пикселям скелетизированного бинарного изображения.

Скелетизированное бинарное изображение можно представить в виде графа. Чёрные пиксели этого изображения в таком представлении являются вершинами графа, а рёбрами они соединены со всеми

чёрными пикселями, граничащими с ними по стороне или по углу.

Запуск обхода такого графа в ширину одновременно из всех вершин, соответствующих пикселям α -, β - и γ -групп, можно считать частным случаем использования алгоритма Ли для бинарного изображения [13]. В ходе работы алгоритма от каждой из начальных вершин будет расходитьсся так называемая волна. Две встречные волны могут встретиться на середине соединяющего ребра, в таком случае каждая из волн может нести информацию о вершине, из которой она начата, а также о количестве пикселей на пути от этой вершины до места встречи этих волн.

Можно заранее определить два правила, по которым однозначно определяются пиксели, являющиеся ключевыми точками, а не изгибами. Вершина соответствует ключевому пикселю, если:

- имеет инцидентную ей петлю;
- имеет более двух исходящих соединительных рёбер.

В оставшихся случаях классификация вершин требует более тщательного анализа. К оставшимся случаям можно отнести лишь вершины, имеющие ровно два исходящих соединительных ребра. Если угол между двумя направляющими векторами исходящих из некоторой вершины соединительных рёбер не менее 120 градусов, то можно считать, что эта вершина соответствует изгибу.

Опираясь на рассмотренные утверждения, можно сформулировать алгоритм разделения ключевых точек и изгибов.

Для поиска множества пикселей, соответствующих ключевым точкам, и аналогичного множества пикселей, соответствующих изгибам, необходимо выполнить следующие шаги:

1. Выделим на бинарном скелетизированном изображении все пиксели, соответствующие α -, β - и γ -группам, обозначим все эти пиксели как соответствующие ключевым точкам.
2. Запустим алгоритм Ли одновременно из всех пикселей, соответствующих ключевым точкам.
3. Если в результате работы алгоритма Ли обнаружено, что некоторые из пикселей, на данный момент считавшихся соответствующими ключевым точкам, на самом деле являются местом стыка двух соединяющих рёбер, направляющие вектора которых при выборе в качестве начального положения рассматриваемого ключевого пикселя расположены под углом более 120 градусов, то обозначим их как соответствующие изгибам и вернёмся к шагу 2.
4. Если в результате работы алгоритма Ли не обнаружено новых изгибов, алгоритм завершён.

Важно заметить, что для каждого последующего запуска алгоритма Ли множество ключевых пикселей, из которых будут одновременно запускаться волны, будет уменьшаться. Это обусловлено тем, что по результатам предыдущего запуска алгоритма некоторые ключевые пиксели могут стать соответствующими изгибам, однако обратный процесс невозмо-

жен. Следовательно, в ходе работы алгоритма будет выполнено ограниченное количество итераций.

Оценить вычислительную сложность такого алгоритма можно как $O(P \cdot K)$, где K – количество пикселей бинарного скелетизированного изображения, соответствующих α -, β - и γ -группам, а P – количество чёрных пикселей бинарного скелетизированного изображения. Для реальных скелетизированных бинарных изображений величина P намного меньше, чем $H \cdot W$, где W и H – ширина и высота изображения соответственно. Величина K же для реальных изображений намного меньше P и редко превышает 15.

4. Алгоритм выделения композитных ребер

Ранее были разработаны алгоритмы для выделения ключевых точек и изгибов на скелетизированном бинарном изображении. Для построения полноценной структурной модели также необходимо выделить соединяющие рёбра, после чего объединить их в композитные.

Для того чтобы сформулировать алгоритм выделения композитных рёбер на исходном изображении, необходимо предварительно заметить, что никакой изгиб не принадлежит более чем одному композитному ребру или петле. Это утверждение явно следует из того факта, что на пути между двумя вершинами, соответствующими соседним ключевым пикселям, все вершины соответствуют изгибам и могут иметь лишь два инцидентных соединяющих ребра.

Так как любой пиксел, соответствующий изгибу, принадлежит только одному композитному ребру, посещение в порядке обхода алгоритмом Ли этого пикселя не вызывает неоднозначности при выборе дальнейшего направления обхода. Таким образом, при запуске алгоритма Ли одновременно из всех ключевых пикселей, пиксели, являющиеся изгибами, не могут повлиять на детерминированность обхода. В таком случае две волны, запущенные из двух ключевых пикселей, связанных композитным ребром, обязательно встретятся где-то между двумя этими пикселями. Если использовать ранее рассмотренный способ определения направляющего вектора волны и подсчёта количества посещённых этой волной пикселей, можно сформулировать алгоритм выделения композитных ребер.

Для того чтобы выделить все композитные рёбра на скелетизированном бинарном изображении с известным положением ключевых точек и изгибов, необходимо выполнить следующие шаги:

1. Запустим алгоритм Ли одновременно из всех ключевых пикселей.
2. Для каждой комбинации стартовой вершины и начального направления движения заведём стек посещённых изгибов и количества шагов на момент их посещения.
3. Если на очередной итерации алгоритма волна посещает вершину, соответствующую изгибу, положим в стек метку этого изгиба.
4. Если на очередной итерации алгоритма встретились две волны, объединим два стека этих волн,

получив итоговую последовательность изгибов в порядке их посещения и расстояния между двумя соседними изгибами. Для такой последовательности по начальным позициям волн однозначно восстанавливается информация о том, какие два ключевых пиксела соединяются через эту последовательность изгибов.

5. По окончании обхода алгоритмом Ли у нас имеется множество последовательностей изгибов, характеризующих пути между соединёнными напрямую ключевыми пикселями. Рассмотрим кратчайшие расстояния $d_{u,v}$ между всеми парами соседних изгибов в каждой из последовательностей, а также количество пикселей $p_{u,v}$, посещённых между этими двумя изгибами. Обозначим за коэффициент искривления величину отношения $p_{u,v}$ к $d_{u,v}$. Эта величина в достаточной степени позволяет определить, насколько сильно выгнута линия, соединяющая два изгиба. Посчитаем эту величину для каждого из отрезков между двумя соединёнными изгибами.

6. Преобразуем каждую из найденных последовательностей в массив соединяющих рёбер. Каждое из таких рёбер будет однозначно задаваться метками изгибов, которые оно соединяет, направляющим вектором начала и конца этого ребра, а также коэффициентом искривления. Полученный массив и является композитным ребром.

Вычислительная сложность такого алгоритма составляет $O(P)$, где P – количество чёрных пикселей на скелетизированном бинарном изображении.

Композитные рёбра и ключевые точки, которые ими соединяются, являются важными структурными составляющими. Именно на их основе можно создать структурную модель символа, которая в дальнейшем будет использоваться в алгоритмах распознавания.

5. Построение структурной модели на основе выделенных структурных составляющих

Если считать все ключевые точки и изгибы вершинами некоторого графа G , а выделенные соединяющие рёбра – его рёбрами, то полученный граф G будет являться планарным графом, задающим топологическую модель символа. Такая модель не содержит информации о геометрических характеристиках графического представления.

Если дополнить полученную топологическую модель информацией о том, какая вершина соответствует ключевому пикселю, а какая – изгибу, можно существенно увеличить информативность этой модели.

Чтобы добавить в модель информацию о геометрических характеристиках графического представления, необходимо добавить положение каждой из вершин такого графа, а для каждого соединяющего ребра – информацию о направляющих векторах с обеих его сторон и коэффициенте искривления, который показывает, во сколько раз длина кривой, проведённой между двумя ключевыми пикселями, больше, чем евклидово расстояние между ними.

В совокупности с направляющим вектором этот коэффициент можно использовать, чтобы аппроксими-

ровать соединяющее ребро некоторым геометрическим примитивом.

В случае, если коэффициент искривления близок к единице, соединяющее ребро можно представить парой смежных отрезков, направление первого из которых совпадает с направляющим вектором, второй же отрезок соединяет первый отрезок со вторым концом соединяющего ребра. Длина первого отрезка подбирается таким образом, чтобы суммарная длина двух отрезков была равна длине соответствующего соединяющего ребра структурной модели.

Соединяющие рёбра с большей степенью искривления удобно представлять в виде дуги окружности или, при необходимости, дуги эллипса. Такие представления могут быть определены аналитически по известным положениям концов дуги и её длине.

6. Алгоритмы сравнения структурных моделей

Тривиальным подходом к решению задачи распознавания символов является поочерёдное сравнение тестируемого графического образа с каждым из эталонных образов, классификация которых известна, и последующий выбор класса наиболее похожего эталонного образа. В общем случае для оценки степени схожести графических представлений двух символов на основе структурных моделей необходимо построить структурную модель для каждого из этих символов в отдельности, после чего две полученные модели необходимо сравнить для оценки их степени схожести. При таком подходе особенно важен выбор алгоритма оценки степени схожести двух структурных моделей символов. Для того, чтобы оценить, какой подход к использованию построенных структурных моделей является наиболее успешным, было разработано три алгоритма оценки степени схожести двух моделей:

- алгоритм на основе метода пересечений;
- алгоритм на основе поиска максимального паросочетания минимального веса;
- вероятностный алгоритм.

Алгоритм на основе метода пересечений опирается на анализ пересечений некоторого набора прямых с каждым из сравниваемых символов. Он нашёл широкое применение в задачах распознавания символов, однако его применение существенно осложняется тем, что исходное графическое начертание символа задано в растровом представлении. Оценка количества пересечений геометрической прямой на плоскости с растровым изображением затрудняется неоднозначностью интерпретации понятия «пересечение прямой и растрового изображения», а также необходимостью анализа многократного пересечения привычного геометрического и растрованного отрезков. Однако в случае со структурной моделью достаточно выполнить пересечение прямой с её геометрическими примитивами.

Наиболее простой алгоритм получения вектора признаков для оценки степени схожести структурных моделей на основе метода пересечений может быть

сформулирован в виде следующей последовательности шагов:

1. Предварительно сгенерируем набор прямых для оценки количества пересечений с графическими представлениями символов. Для генерации можно использовать эвристические правила или эволюционные алгоритмы.
2. Будем рассматривать все прямые набора в определённом порядке. Выберем первую в порядке рассмотрения прямую l_i .
3. Изначально полагаем счётчик количества пересечений для прямой l_i равным нулю.
4. Поочередно выполним пересечение прямой l_i с каждым из геометрических примитивов, представляющих соединяющие рёбра структурной модели. Для каждого геометрического примитива, для которого существует единственная точка пересечения прямой l_i , необходимо выполнить увеличение счётчика количества пересечений для этой прямой на единицу.
5. Выберем следующую в порядке рассмотрения прямую l_i и перейдём к шагу 3. Если все прямые рассмотрены, то перейдём к шагу 6.
6. Объединим счётчики для всех прямых в вектор, который по своей сути является вектором признаков для дальнейшей классификации структурных моделей. Алгоритм закончен.

Вычислительная сложность метода пересечений на основе структурных моделей при таком подходе оценивается как $O(E \cdot L)$, где E – количество соединяющих рёбер в структурной модели, а L – количество прямых, используемых для подсчёта количества пересечений.

Алгоритм на основе поиска максимального паросочетания минимального веса использует площади фигур, заключённых между всеми возможными парами композитных рёбер из разных моделей. Эту площадь с противоположным знаком можно использовать как меру схожести между рёбрами: чем больше площадь, тем больше отличаются данные рёбра.

Помимо площади, на оценку степени схожести должно влиять расстояние между соответствующими точками композитных рёбер – чем дальше находятся эти точки, тем меньше похожи соответствующие композитные рёбра.

Для оценки степени схожести двух отрезков композитных рёбер можно воспользоваться следующей моделью. Пусть первый отрезок задан двумя своими концами (x_a, y_a) и (x_b, y_b) , а второй – концами (x_c, y_c) и (x_d, y_d) . Представим, что вдоль каждого из отрезков движется некоторый объект. В некоторый момент времени t объект на первом отрезке будет находиться в точке (x_1, y_1) , на втором – в точке (x_2, y_2) . Положения объектов описываются системами (1):

$$\begin{cases} x_1 = x_a + (x_b - x_a) \cdot t, & x_2 = x_c + (x_d - x_c) \cdot t, \\ y_1 = y_a + (y_b - y_a) \cdot t, & y_2 = y_c + (y_d - y_c) \cdot t. \end{cases} \quad (1)$$

Квадрат расстояния между ними может быть вычислен с помощью формулы (2):

$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2. \quad (2)$$

Первое слагаемое можно преобразовать следующим образом:

$$\begin{aligned} (x_1 - x_2)^2 &= ((x_a - x_c) + t(x_b + x_c - x_a - x_d))^2 = \\ &= (A_x + B_x t)^2 = B_x^2 t^2 + 2A_x B_x t + A_x^2. \end{aligned} \quad (3)$$

Преобразовав схожим с выражением (3) образом и второе слагаемое формулы (2), можно представить квадрат расстояния в виде выражения (4):

$$\begin{aligned} d^2 &= (B_x^2 + B_y^2) t^2 + 2(A_x B_x + A_y B_y) t + \\ &+ (A_x^2 + B_y^2) = at^2 + bt + c. \end{aligned} \quad (4)$$

Следовательно, степень схожести двух композитных рёбер в интервале времени от t_1 до t_2 может быть определена с помощью выражения (5):

$$S = \int_{t_1}^{t_2} \sqrt{at^2 + bt + c} dt. \quad (5)$$

С помощью справочников [14] или математических пакетов значение такого интеграла может быть представлено в аналитическом виде, что позволяет избежать необходимости в использовании численных методов, которые могут существенно понизить быстродействие алгоритма.

Пусть для каждой пары композитных рёбер из разных структурных моделей известна их степень схожести. Тогда можно построить двудольный граф G , в котором вершинами будут являться композитные рёбра исходных структурных моделей. К первой доле графа G будут относиться вершины, которые соответствуют композитным рёбрам из первой структурной модели, ко второй доле – вершины, соответствующие второй модели.

Для того, чтобы оценить степень схожести двух структурных моделей, необходимо вершинам первой доли сопоставить вершины второй доли так, чтобы было образовано наибольшее возможное количество пар. Из всех таких распределений по парам необходимо выбрать разбиение, при котором сумма степеней схожести будет максимальной. Такая задача будет равнозначна задаче поиска максимального паросочетания минимального веса в двудольном графе [14] при условии, что ребро такого графа будет иметь вес, равный площади между соответствующими композитными рёбрами.

Такая задача может быть сведена к задаче поиска максимального потока наименьшей стоимости в сети, полученной из исходного двудольного графа путём добавления истока, соединённого со всеми вершинами первой доли, и стока, соединённого со всеми вершинами второй доли. Задача поиска такого потока может быть решена с использованием алгоритма увеличивающих путей [15].

Для нахождения максимального потока наименьшей стоимости в сети G необходимо выполнить следующие шаги:

1. Предварительно подготовим граф G , добавив для каждого ребра обратное ребро с нулевой пропускной способностью и стоимостью, равной стоимости исходного ребра с противоположным знаком.
2. Используя в качестве весов рёбер их стоимости, найдём кратчайший путь от истока к стоку.
3. Если не существует ни одного пути от истока к стоку, алгоритм завершён, найденный к данной итерации поток является максимальным потоком минимальной стоимости.
4. Пустим поток по найденному маршруту. Для этого рассмотрим все его рёбра. Определим то, пропускная способность которого минимальна, обозначим её за f . Уменьшим пропускную способность всех рёбер найденного пути на f , одновременно увеличивая пропускную способность обратных им рёбер на эту же величину.
5. Для обновлённой сети снова выполним шаг 2.

Вычислительная сложность такого алгоритма $O(V^3 \cdot E)$, где V – количество вершин в сети, E – количество рёбер в ней, включая добавленные на первом шаге алгоритма обратные рёбра. Такая оценка возникает при использовании алгоритма Дейкстры для поиска кратчайшего пути. Быстродействие алгоритма можно улучшить, используя для реализации алгоритма Дейкстры бинарную кучу.

Для описания метода оценки степени схожести двух структурных моделей на основе вероятностного теста представим, что в некоторой точке структурной модели находится некоторый объект, который способен выполнять следующие операции:

- переместиться вдоль некоторого композитного ребра на расстояние d в направлении единичного вектора \mathbf{v} ;
- переместиться к ближайшей ключевой точке.

Первая операция должна выполняться только в том случае, если существует композитное ребро, направление которого отличается от заданного не более чем на определённую величину δ . Если таких рёбер несколько, то выбирается композитное ребро, направление которого ближе всего к направлению, заданному вектором \mathbf{v} . Расстояние d , на которое необходимо переместиться объекту, выбирается таким образом, чтобы оно было меньше самого короткого соединяющего ребра. На практике эта величина составляет порядка 8×10^{-2} при условии, что описывающий квадрат для этого символа имеет единичную сторону.

Один запуск теста заключается в размещении объекта внутри каждой из структурных моделей в ключевой точке, наиболее близкой к заданному начальному положению (x_s, y_s) , и последующем выполнении набора операций двух описанных ранее типов. После того, как выполнение операций заканчивается, расстояние между объектами на первой и на второй структурных моделях и является мерой различия: чем дальше расположены объекты, тем меньше соответствующие структурные модели похожи между собой.

Тем не менее, обнаружение различий структурных моделей существенно зависит от выбора набора команд, которые будут выполнены в ходе теста на схожесть. Зачастую такой набор формируется случайным образом. В таком случае существует вероятность выбрать нерепрезентативный набор операций, который для отдельных пар структурных моделей символов разных классов приведёт к схожей позиции итоговой позиции объекта. Аналогично, неудачно выбранный набор операций может для двух моделей символов одного класса привести к удалённым итоговым положениям объектов. Таким образом, с некоторой вероятностью будет выбран набор операций, который приведёт к неверной классификации структурной модели.

Стоит отметить, что начальная позиция объекта (x_s, y_s) тоже выбирается случайным образом, что усиливает вероятностный характер такого рода теста.

Сам алгоритм классификации структурной модели на основе вероятностного теста может быть сформулирован в виде следующей последовательности шагов:

1. Случайным образом зададим некоторое начальное положение (x_s, y_s) .
2. Случайным образом зададим набор команд двух описанных типов.
3. Для структурной модели каждого из эталонных изображений выполним шаги с 4-го по 6-й.
4. Разместим объект в ключевой точке рассматриваемой модели, которая наиболее близка к заданному начальному положению (x_s, y_s) .
5. Поочерёдно выполним все команды заданного набора для рассматриваемой структурной модели.
6. Запишем итоговое положение объекта для рассматриваемой структурной модели.
7. Рассмотрим одну из ранее не рассмотренных структурных моделей, соответствующую изображению, которое необходимо классифицировать.
8. Если нерассмотренных структурных моделей не осталось, то алгоритм завершён.
9. Иначе для рассматриваемой модели выполним шаги с 4-го по 5-й и пометим её как рассмотренную.
10. Определим структурную модель, соответствующую эталонному изображению, такую, что расстояние между сохранённым для неё итоговым положением и итоговым положением для рассматриваемой модели минимально.
11. Отнесём классифицируемое изображение к классу, соответствующему найденному эталонному изображению.
12. Перейдём к шагу 7.

7. Анализ результатов численных экспериментов по распознаванию рукописных символов

Для сравнения разработанных алгоритмов с существующими аналогами было реализовано приложение, осуществляющее выделение структурных составляющих и последующее построение структурных моделей на их основе. В разработанном приложении были реализованы три предложенных метода сравнения структурных моделей.

Исследование качества распознавания алгоритмов выполнялось на двух наборах данных: наборе символов из экзаменационных бланков ЕГЭ (13312 символов, 68 классов – латинские и кириллические буквы, а также арабские цифры) и общеизвестном наборе рукописных цифр MNIST (70000 символов, 10 классов – арабские цифры).

Важным преимуществом алгоритмов на основе структурных моделей является возможность качественного распознавания образов при отсутствии большого количества эталонных изображений. Поэтому сравнение алгоритмов с использованием структурных моделей (SMI – метод пересечений, SMM – метод нахождения максимального паросочетания, SMT – вероятностный тест) производилось с алгоритмом на основе растрового варианта метода пересечений [16] (IM), а также классификаторами на основе машины опорных векторов [17] (SVM) и вероятностной нейронной сети [18] (PNN). В качестве вектора признаков в подходе на основе метода опорных векторов используются гистограммы ориентированных градиентов. В методе на основе вероятностной нейронной сети в качестве вектора признаков используется само бинарное изображение. Все алгоритмы настраивались и обучались с использованием лишь трёх ($E=3$), пяти ($E=5$) и семи ($E=7$) эталонных изображений для каждого класса. Результаты сравнения приведены в табл. 1. Для каждого из методов при $E=3$, $E=5$ и $E=7$ был определён процент правильно классифицированных символов.

Табл. 1. Результаты сравнения качества распознавания на наборе символов из ЕГЭ

E	SMI	SMM	SMT	IM	SVM	PNN
3	98,9	99,2	93,2	95,3	88,7	72,1
5	99,3	99,4	94,4	96,1	89,2	74,8
7	99,3	99,5	95,1	96,2	90,0	75,7

Как можно заметить, предложенные алгоритмы имеют более высокую точность распознавания, чем общеизвестные методы классификации, рассчитанные на наличие большего количества эталонных изображений. Сравнительно высокий процент верно распознанных графических представлений объясняется намерением исполнителей начертания соответствовать приведённым образцам.

Для базы рукописных цифр MNIST общий уровень качества распознавания несколько ниже.

Табл. 2. Результаты сравнения качества распознавания на наборе символов набора MNIST

E	SMI	SMM	SMT	IM	SVM	PNN
3	89,1	93,2	78,1	87,4	83,8	70,9
5	90,8	95,1	79,4	88,7	85,5	73,2
7	91,2	95,1	80,0	88,9	87,0	74,2

Это объясняется большим разнообразием почерков и вариаций начертания одних и тех же символов. По этой же причине на этом наборе вероятностный тест оказался недостаточно эффективен, в то время как подход с нахождением максимального паросоче-

тания значительно превосходит по точности распознавания рассматриваемые алгоритмы.

Предположение о необходимости в использовании дополнительной операции устранения плоских окончаний было сделано на основе анализа результатов тестирования предложенного алгоритма. Однако подробный анализ каждой из построенных моделей используемых наборов данных затруднителен, ведь единственный надёжный способ оценки качества построенной структурной модели – это оценка визуального сходства, которая может быть произведена лишь человеком. Тем не менее, можно использовать оценку точности распознавания в качестве параметра, который косвенно отражает качество построения структурных моделей. Сравнение точности распознавания производилось для каждого из трёх предложенных алгоритмов для количества эталонных изображений $E=15$ на наборе рукописных цифр MNIST [19] и подготовленном наборе символов из бланков, аналогичных бланкам ЕГЭ.

Табл. 3. Точность распознавания предложенных алгоритмов с использованием операции устранения плоских окончаний и без нее на символах набора MNIST

	SMI	SMM	SMT
С устранением	91,4%	95,3%	80,3%
Без устранения	74,3%	82,6%	63,9%

Табл. 4. Точность распознавания предложенных алгоритмов с использованием операции устранения плоских окончаний и без нее на символах из бланков

	SMI	SMM	SMT
С устранением	99,4%	99,5%	95,5%
Без устранения	82,1%	86,9%	74,7%

Как можно заметить, для каждого из трёх алгоритмов точность распознавания существенно улучшается вследствие применения операции устранения плоских окончаний.

Еще одним критерием, по которому можно оценить полезность операции устранения плоских окончаний, является соотношение количества пикселей скелетизированного изображения до и после применения данной операции. Для удобства восприятия такого рода информации человеком изменение количества пикселей следует измерять в процентах.

Было выделено семь наиболее значимых интервалов изменения количества пикселей:

- два интервала уменьшения количества пикселей: $-(5\%; \infty)$, $-(0\%; 5\%)$;
- один интервал неизменности количества пикселей: $\{0\%$ };
- четыре интервала увеличения количества пикселей: $+(0\%; 5\%)$, $+(5\%; 15\%)$, $+(15\%; 25\%)$, $+(25\%; \infty)$.

Табл. 5. Изменение количества чёрных пикселей при использовании операции устранения плоских окончаний

	$-(5; \infty)$	$-(0; 5]$	$\{0\}$	$+(0; 5]$	$+(5; 15]$	$+(15; 25]$	$+(25; \infty)$
MNIST	0,12%	1,94%	60,63%	23,07%	7,60%	5,42%	1,22%
Бланки	0,27%	2,39%	50,52%	38,12%	4,52%	3,29%	0,89%

Как можно заметить, достаточно редко использование такого рода операции приводит к уменьшению количества пикселей бинарного изображения в результате скелетизации. Как правило, если такое уменьшение происходит, то оно составляет не более 5% от исходного количества пикселей. В свою очередь, увеличение количества пикселей, как правило, соответствует устранению эффекта чрезмерной скелетизации – удаления пикселей, которое приводит к нарушению топологии символа. Как видно, примерно в одном проценте случаев сохраняется более 25% пикселей. На практике такое существенное увеличение количества пикселей после скелетизации всегда соответствует сохранению какого-либо структурного элемента топологии символа.

Результаты проведённых экспериментов позволяют сделать выводы о целесообразности применения операции устранения плоских окончаний элементов на бинарном изображении, однако необходимо исследовать и необходимость в использовании алгоритма скелетизации Ву–Цая для устранения недоскелетизированных участков результатов работы алгоритма Зонга–Суня. Оценить такую необходимость в терминах точных значений существенно проще. Количество пикселей, удалённых в ходе работы алгоритма Ву–Цая, соответствует количеству пикселей, которые должны быть удалены алгоритмом Зонга–Суня, но были пропущены.

Табл. 6. Распределение изображений используемых наборов по количеству пропущенных пикселей

	0	1	2-3	4-7	8-15	16-∞
MNIST	73,67%	11,92%	9,91%	3,27%	1,17%	0,06%
Бланки	66,78%	16,02%	9,92%	4,58%	2,56%	0,14%

Как видно из табл. 6, дополнительная обработка алгоритмом Ву–Цая необходима примерно четверти изображений набора MNIST и трети изображений набора из бланков, аналогичных бланкам ЕГЭ. На графическом представлении результатов экспериментов заметно, что для большинства изображений количество пропущенных пикселей, которые необходимо удалить, находится в диапазоне от 0 до 7. Крайне редко возникает необходимость в удалении большего числа пикселей.

Полученные результаты позволяют сделать вывод о необходимости использования для существенной части изображений обработки алгоритмом Ву–Цая с целью устранения пикселей, которые образуют области на бинарном изображении, являющиеся причиной неоднозначности при выборе порядка обхода изображения алгоритмом Ли.

Заключение

Статья посвящена разработке оригинального алгоритма построения структурной модели рукописного символа по его изображению и последующему использованию этой модели для решения задачи распознавания этого символа. Существенным преимуществом такого рода алгоритмов является возможность

корректной работы без большого количества эталонных изображений для каждого из классов.

Основные результаты исследования, описанного в статье:

- разработан оригинальный подход к скелетизации бинарного изображения на основе совместного использования алгоритмов скелетизации Зонга–Суня и Ву–Цая, а также операции предварительной обработки с целью устранения плоских окончаний исходного начертания;

- разработан оригинальный подход к извлечению структурных составляющих на скелетизированном бинарном изображении на основе выделения ключевых точек, изгибов, соединяющих и композитных рёбер;

- предложены три метода оценки степени схожести структурных моделей двух символов: метод на основе поиска максимального паросочетания минимальной стоимости, подход на основе метода пересечений и алгоритм вероятностного тестирования на схожесть.

Было установлено, что в условиях малого количества эталонных изображений разработанные алгоритмы обеспечивают достаточно высокий процент верно классифицированных изображений рукописных символов. Наиболее высокое качество классификации достигается при использовании метода на основе поиска максимального паросочетания наименьшей стоимости.

Оценка качества распознавания на наборе рукописных цифр MNIST в среднем ниже, чем оценка тех же самых методов на наборе символов из бланков ЕГЭ. Это обусловлено большей вариативностью форм начертания в наборе MNIST.

Стоит отметить, что предложенный подход на основе построения структурных моделей требует выполнения сравнений структурной модели классифицируемого изображения с каждой из структурных моделей эталонных изображений, что приводит к низкой эффективности алгоритма при большом количестве эталонных изображений. Кроме того, большое количество эталонных изображений позволяет качественно обучить признаковые классификаторы, что избавляет от необходимости использовать методы на основе выделения структурных составляющих.

Была экспериментально подтверждена эффективность предложенного подхода к скелетизации бинарного представления рукописного символа. В ходе исследования было установлено, что примерно в 25% случаев алгоритм Зонга–Суня оставляет не до конца скелетизированные участки изображения, утоньшение которых может быть выполнено с помощью алгоритма Ву–Цая, который, в свою очередь, ни разу в ходе проведённых экспериментов, не оставил ни одного такого участка. Целесообразность использования операции удаления плоских окончаний графического представления символа подтверждается существенным увеличением количества пикселей итогового

го скелета (на 5% или более) в результате её применения примерно в 15% рассмотренных случаев.

Литература

1. **Mori, Sh.** Historical review of OCR research and development / Sh. Mori, Ch. Y. Suen, K. Yamamoto // Proceedings of the IEEE. – 1992. – Vol. 80, Issue 7. – P. 1029-1058. – DOI: 10.1109/5.156468.
2. **Heliński, M.** Report on the comparison of Tesseract and ABYY FineReader OCR engines. Technical report / M. Heliński, M. Kmiecik, T. Parkoła. – Poznań, Poland: Poznań Supercomputing and Networking Center, 2012. – 24 p.
3. **Спицын, В.Г.** Применение вейвлет-преобразования Хаара, метода главных компонент и нейронных сетей для оптического распознавания символов на изображениях в присутствии импульсного шума / В.Г. Спицын, Ю.А. Болотова, Н.Х. Фан, Т.Т.Ч. Буй // Компьютерная оптика. – 2016. – Т. 40, № 2. – С. 249-257. – DOI: 10.18287/2412-6179-2016-40-2-249-257.
4. **Breuel, T.M.** High-performance ocr for printed english and fraktur using LSTM networks / T.M. Breuel, A. UHHasan, M. Al Azawi, F. Shafait // Proceedings of 12th International Conference on Document Analysis and Recognition. – 2013. – P. 683-687. – DOI: 10.1109/ICDAR.2013.140.
5. **Jaderberg, M.** Reading text in the wild with convolutional neural networks / M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman // International Journal of Computer Vision. – 2016. – Vol. 116, Issue 1. – P. 1-20. – DOI: 10.1007/s11263-015-0823-z.
6. **Wang, T.** End-to-end text recognition with convolutional neural networks / T. Wang, D.J. Wu, A. Coates, A.Y. Ng // Proceedings of the 21st International Conference on Pattern Recognition. – 2012. – P. 3304-3308.
7. **Lotfi, M.** Combining wavelet transforms and neural networks for image classification / M. Lotfi, A. Solimani, A. Dargazany, H. Afzal, M. Bandarabadi // 41st Southeastern Symposium on System Theory, Tullahoma, TN, USA. – 2009. – P. 44-48. – DOI: 10.1109/SSST.2009.4806819.
8. **Wang, P.S.P.** An improved structural approach for automated recognition of handprinted characters / P.S.P. Wang, A. Gupta // International Journal of Pattern Recognition and Artificial Intelligence. – 1991. – Vol. 05, Issue 01n02. – P. 97-121. – DOI: 10.1142/S0218001491000089.
9. **Otsu, N.** A Threshold Selection Method From Gray-level Histograms. / N. Otsu // IEEE Transactions on Systems, Man, and Cybernetics. – 1979. – Vol. SMC-9(1). – P. 62-66.
10. **Widiarti, A.R.** Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendraprasad-Wang-Gupta Thinning / A.R. Widiarti // International Journal of Computer, Electrical, Automation, Control and Information Engineering – 2011 – Vol. 5(6). – P. 20-24.
11. **Zhang, T.Y.** A Fast Parallel Algorithm for Thinning Digital Patterns / T.Y. Zhang, C.Y. Suen // Communications of the ACM. – 1984. – Vol. 27, Issue 3. – P. 236-239. – DOI: 10.1145/357994.358023.
12. **Wu, R.-Y.** A new one-pass parallel thinning algorithm for binary images / R.-Y. Wu, W.-H. Tsai // Pattern Recognition Letters. – 1992. – Vol. 13, Issue 10. – P. 715-723. – DOI: 10.1016/0167-8655(92)90101-5.
13. **Lee, C.Y.** An Algorithm for Path Connections and Its Applications. / C.Y. Lee // IRE Transactions on Electronic Computers. – 1961. – Vol. EC-10, Issue 3. – P. 346-365. – DOI: 10.1109/TEC.1961.5219222.
14. **Градштейн, И.С.** Таблицы интегралов, сумм, рядов и произведений / И.С. Градштейн, И.М. Рыжик. – 4-е изд. – М.: Наука, 1963 – 1108 с.
15. **Кристофидес, Н.** Теория графов. Алгоритмический подход / Н. Кристофидес. – Пер. с англ. – М.: Мир, 1978. – 432 с.
16. **Хаустов, П.А.** Генетический алгоритм поиска множества кривых для оптического распознавания символов с использованием метода пересечений / П.А. Хаустов, В.Г. Спицын, Е.И. Максимова // Современные проблемы науки и образования. – 2014. – № 6.
17. **Vapnik, V.N.** Support Vector Networks / V. Vapnik, C. Cortes // Machine Learning. – 1995. – № 20(3). – P. 273-297. – DOI: 10.1023/A:1022627411411.
18. **Хаустов, П.А.** Разработка системы оптического распознавания символов на основе совместного применения вероятностной нейронной сети и вейвлет-преобразования / П.А. Хаустов, Д.С. Григорьев, В.Г. Спицын // Известия Томского политехнического университета. – 2013 – Т. 323, № 5. – С. 101-105.
19. **LeCun, Y.** Gradient-Based Learning Applied to Document Recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. – 1998. – Vol. 86, Issue 11. – P. 2278-2324. – DOI: 10.1109/5.726791.

Сведения об авторе

Хаустов Павел Александрович, 1990 года рождения, в 2013 году получил степень магистра по направлению «Информатика и вычислительная техника» на кафедре вычислительной техники Национального исследовательского Томского политехнического университета. С 2013 г. по 2016 г. являлся аспирантом этой кафедры. Область научных интересов: алгоритмы теории графов, алгоритмы компьютерной геометрии, обработка графических изображений, программирование, разработка web-приложений. E-mail: exceibot@tpu.ru.

ГРНТИ 28.23.15.

Поступила в редакцию 3 июня 2016 г. Окончательный вариант – 22 декабря 2016 г.

ALGORITHMS FOR HANDWRITTEN CHARACTER RECOGNITION BASED ON CONSTRUCTING STRUCTURAL MODELS

P.A. Khaustov¹

¹ Tomsk Polytechnic University, Tomsk, Russia

Abstract

The article is devoted to the development of algorithms for handwritten character recognition based on constructing structural models. These algorithms do not require a large number of reference images

for the correct functioning. Also, an approach to a thinning of the binary character representation based on the joint use of Zhang-Suen and Wu-Tsai algorithms has been proposed. The effectiveness of the proposed approach is confirmed by the results of experiments.

The article includes a detailed description of all steps of the algorithm for constructing structural models. Results of the proposed algorithm's verification are provided, as well as their comparison with other character recognition algorithms. Algorithms that can operate under a limited number of reference images were used for the comparison.

Keywords: character recognition, structural components, structural models, computer vision, skeletonization, binarization.

Citation: Khaustov PA. Algorithms for handwritten character recognition based on constructing structural models. *Computer Optics* 2017; 41(1): 67-78. DOI: 10.18287/2412-6179-2017-41-1-67-78.

References

- [1] Mori Sh, Suen ChY, Yamamoto K. Historical review of OCR research and development. *Proc IEEE* 1992; 80(7): 1029-1058. DOI: 10.1109/5.156468
- [2] Heliński M, Kmiecik M, Parkoła T. Report on the comparison of Tesseract and ABBYY FineReader OCR engines. Technical report. Poznań, Poland: Poznań Supercomputing and Networking Center; 2012.
- [3] Spitsyn VG, Bolotova YuA, Phan NH, Bui TTT. Using a Haar wavelet transform, principal component analysis and neural networks for OCR in the presence of impulse noise. *Computer Optics* 2016; 40(2): 249-257. DOI: 10.18287/2412-6179-2016-40-2-249-257.
- [4] Breuel TM, Ul-Hasan A, Azawi MAI, Shafait F. High-Performance OCR for Printed English and Fraktur using LSTM Networks. *Proc 12th International Conference on Document Analysis and Recognition* 2013: 683-687. DOI: 10.1109/ICDAR.2013.140.
- [5] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Reading Text in the Wild with Convolutional Neural Networks. *International Journal of Computer Vision* 2016; 116(1): 1-20. DOI: 10.1007/s11263-015-0823-z.
- [6] Wang T, Wu DJ, Coates A, Ng AY. End-to-End Text Recognition with Convolutional Neural Networks. *Proc ICRP 21* 2012: 3304-3308.
- [7] Lotfi M, Solimani A, Dargazany A, Afzal H, Bandarabadi M. Combining wavelet transforms and neural networks for image classification. *SSST* 2009: 44-48. DOI: 10.1109/SSST.2009.4806819.
- [8] Wang PSP, Gupta A. An improved structural approach for automated recognition of handprinted characters. *Int J Patt Recog Artif Intell* 1991; 05(01n02): 97-121. DOI: 10.1142/S0218001491000089.
- [9] Otsu NA. Threshold Selection Method From Gray-level Histograms. *IEEE Trans Sys, Man, Cybern* 1979; SMC-9(1): 62-66.
- [10] Widiarti AR. Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendraprasad-Wang-Gupta Thinning. *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 2011; 5(6): 20-24.
- [11] Zhang TY, Suen CY. A Fast Parallel Algorithm for Thinning Digital Patterns. *Communications of the ACM* 1984; 27(3): 236-239. DOI: 10.1145/357994.358023.
- [12] Wu R-Y, Tsai W-H. A new one-pass parallel thinning algorithm for binary images. *Pattern Recognition Letters* 1992; 13(10): 715-723. DOI: 10.1016/0167-8655(92)90101-5.
- [13] Lee CY. An Algorithm for Path Connections and Its Applications. *IRE Transactions on Electronic Computers* 1961; EC-10(3): 346-365. DOI: 10.1109/TEC.1961.5219222.
- [14] Gradshteyn IS, Ryzhik IM. Tables of integrals, series and products (Corrected and enlarged edition). London: Academic Press, Inc.; 1980. ISBN: 978-0-12-294760-5.
- [15] Christofides N. Graph Theory. An Algorithmic Approach. Orlando, FL, USA: Academic Press, Inc.; 1975. ISBN: 978-0121743505.
- [16] Khaustov PA, Spitsyn VG, Maksimova EI. Genetic algorithm of set of lines searching for optical character recognition based on the method of intersections [In Russian]. *Modern problems of science and education* 2014; 6.
- [17] Vapnik VN, Cortes C. Support Vector Networks. *Machine Learning* 1995; 20(3): 273-297. DOI: 10.1023/A:1022627411411.
- [18] Khaustov PA, Grigoryev DS, Spitsyn VG. The development of optical character recognition approach on the basis of joint application of probabilistic neural network and wavelet transform [In Russian]. *Bulletin of the Tomsk Polytechnic University* 2013; 323(5): 101-105.
- [19] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient based learning applied to document recognition. *Proceedings of the IEEE* 1998; 86(11): 2278-2324. DOI: 10.1109/5.726791.

Author's information

Pavel Aleksandrovich Khaustov (b. 1990) received a master's degree in "Computer Science and Technology" at Tomsk Polytechnic University in 2013. Studied as a postgraduate student of the Computer Engineering department in Tomsk Polytechnic University from 2013 to 2016. Research interests are graph theory algorithms, computer geometry algorithms, image processing, programming, web-development. E-mail: exceibot@tpu.ru.

Received June 3, 2016. The final version – December 22, 2016.