

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный университет»

Кафедра Системного Программирования

Малыгин Евгений Сергеевич

Устойчивая к шумам сегментация
автомобильных номеров в низком
разрешении

Бакалаврская работа

Допущен к защите.
Зав. кафедрой:
д. ф.-м. н., профессор Терехов А. Н.

Научный руководитель:
к. ф.-м. н. Вахитов А. Т.

Рецензент:
Николаев А. И.

Санкт-Петербург
2015

SAINT PETERSBURG STATE UNIVERSITY

Chair of Software Engineering

Eugene Malygin

Noise-robust segmentation of low resolution
license plate images

Bachelor's Thesis

Admitted for defence.

Head of the chair:

Professor Andrey Terehov

Scientific supervisor:

Ph. D. Alexander Vakhitov

Reviewer:

Alexander Nikolaev

Saint-Petersburg

2015

Оглавление

Введение	4
1. Постановка задачи	6
2. Алгоритм сегментации	7
2.1. Общая схема	7
2.2. Бинаризация и препроцессинг	8
2.2.1. Выравнивание изображения по яркости	8
2.2.2. Адаптивная бинаризация	8
2.2.3. Удаление границ номерного знака	9
2.3. Подготовка символьных компонент	10
2.3.1. Поиск всех компонент связности	10
2.3.2. Удаление неподходящих компонент	10
2.3.3. Разделение компонент	11
2.3.4. Склеивание компонент	12
2.3.5. Улучшение левой компоненты	12
2.4. Перебор гипотез	15
2.4.1. Снижение времени перебора	15
2.4.2. Определение типа номерного знака	16
2.4.3. Улучшение правой компоненты	17
2.4.4. Проверка допустимости гипотезы	18
2.4.5. Оценка правдоподобности гипотезы	19
3. Эксперименты	22
Заключение	24
Список литературы	25

Введение

В настоящее время системы автоматической идентификации автомобильных номеров активно внедряются в дорожную инфраструктуру для автоматического взимания платы за проезд или за парковку, для мониторинга дорожного движения и для обнаружения нарушений ПДД. Идентификация автомобильных номеров является востребованной и постоянно развивающейся областью.

Есть технические ограничения для систем автоматической идентификации номеров: пропускные способности каналов связи ограничены и находятся в противоречии с необходимостью получать кадры высокого разрешения с необходимой частотой, поэтому идентификацию приходится проводить в плохих условиях.

Идентификация номеров обычно может быть разбита на 4 этапа: получение кадра, поиск номерного знака, сегментацию, распознавание.

Сегментация – это извлечение из изображения номерного знака информации о местоположении отдельных символов. Сегментация считается сложнейшей задачей при автоматической идентификации номеров, потому что от качества её результатов зависит успешность распознавателя [10].

Описание одной из первых успешных систем визуальной автоматической идентификации номеров приведено в 1991 году [12]. Изображение, подаваемое с камеры, приводилось к чёрно-белому виду, а затем на нём производился поиск компонент связности, среди которых оставались лишь те, что подходили по нескольким шаблонам. В итоге система достигала общей точности распознавания более чем в 99% при ограниченных условиях применимости.

В более поздней работе [3], в которой сегментация также основана на компонентах связности, основной акцент делался на многообразии условий работы метода. Был предусмотрен механизм откидывания неподходящих компонент связности, а для слипшихся, разделившихся или ушедших компонент были предусмотрены соответствующие операторы восстановления. Точность метода достигла 93.7%.

Для сегментации также можно использовать вертикальные и горизонтальные проекции изображений номеров [11]. Горизонтальные проекции позволяли находить уровень высоты, на котором находятся символы, а с помощью вертикальных проекций проводились границы между буквами. Авторы статьи создали прототип приложения, использующего предложенный ими подход.

Похожий подход был использован в другой работе [4]. Отличие заключается в том, что горизонтальные проекции дополнительно использовались для обнаружения двухстрочных номерных знаков, а вертикальные проекции применялись в связке с ограничениями на пропорциональность символов, чтобы улучшить работу системы в неидеальных условиях. Итоговая точность системы составила 92.85%.

В методе [6] сегментация опирается на заранее известные параметры номера – его

размеры, различающиеся при обзоре из разных точек. При этом под сегментацией понимается не получение частей изображения, содержащих отдельные символы, а получение части изображения, содержащей всю символьную строку. Полученная в итоге система в реальном времени распознаёт около 60% проезжающих машин, что для задачи мониторинга дорожного движения является удовлетворительным результатом.

Есть другой подход, опирающийся на заранее известные данные [1]. При таком подходе изображение номерного знака разделяется на прямоугольники с символами в известных местах. Это делает сегментацию устойчивой к шумам, но любые отклонения границ номерного знака влекут за собой ошибки в сегментации.

Для сегментации было предложено использовать алгоритм выделения контуров [7]. Эксперименты показали, что сегментация с помощью алгоритма выделения контуров даёт лучшие результаты, чем сегментация, основанная на информации о расположении символов.

Нахождение контуров также было применено для сегментации в другой работе [5]. Сначала с помощью Fast Marching было получено грубое первичное приближение контуров символов, а затем происходило уточнение, пока 95% точек контура не было зафиксировано. Такой подход оказался способен находить контуры даже распавшихся на части символов, а точность распознавания системы достигла 98.96%.

Ссылки на другие методы сегментации и общий обзор области автоматической идентификации номеров можно найти в обзорной статье [2].

В рамках работы над системой автоматической идентификации номеров была решена задача сегментации. Используемый в существующей системе алгоритм был основан на проекциях и пытался разбить номер на интервалы равной длины, сумма проекций на границах которых была бы минимальна. Такой алгоритм работал недостаточно хорошо, потому что шумы и границы номера влияли на проекции.

Был необходим алгоритм, который будет работать лучше старого. В качестве базы для алгоритма было выбрано использование компонент связности. Такой подход сравнительно прост для реализации и подбора параметров, но его слабое место – сегментация слипнувшихся или разделившихся символов.

1. Постановка задачи

Цель данной работы – повысить качество распознавания номеров в системе, измеряемое на размеченных наборах данных, за счёт усовершенствования алгоритма сегментации.

Для достижения данной цели были поставлены следующие задачи:

- Предложить алгоритм сегментации, эффективно работающий на зашумлённых изображениях.
- Реализовать алгоритм в рамках системы автоматической идентификации номеров.
- Подготовить наборы данных для тестирования.
- Провести сравнительное тестирование нового и старого подходов.

2. Алгоритм сегментации

Для построения эффективного алгоритма были изучены статьи, относящиеся к предметной области, и было решено работать с компонентами связности.

От алгоритма требовалось, чтобы он работал недолго, но при этом число ложных сегментаций было минимально.

В процессе итеративной разработки алгоритма был произведён анализ возникающих при сегментации проблем, и были предложены пути для их решения. В итоге была получена окончательная схема алгоритма.

Государственные регистрационные номерные знаки в РФ можно разделить на 2 части: та часть, что идёт до разделителя региона, и сам регион. Эти части в контексте данной работы будут называться просто «номер» и «регион» соответственно, хотя первое обозначение и отличается от общепринятого. Весь номерной знак в целом будет называться «номерным знаком».

Сегментация номера и региона идут отдельно, но по одной общей схеме.

2.1. Общая схема

Алгоритм можно разделить на 3 этапа:

- Бинаризация и препроцессинг.

На вход подаётся повернутое горизонтально масштабированное изображение номерного знака.

На выходе – двоичное (чёрно-белое) изображение номерного знака, на котором по возможности удалены левая и правая границы рамки.

Проблемы, которые нужно решить на этом этапе: переменная яркость изображений в течение дня, неравномерное освещение номерных знаков, слипающиеся с границами рамки символы.

- Подготовка символьных компонент.

На вход подаётся двоичное изображение, полученное на прошлом этапе.

На выходе – множество прямоугольников, которые могут соответствовать реальным символам на изображении номерного знака.

Проблемы, которые нужно решить на этом этапе: шумы, границы номерного знака как отдельные компоненты, разделившиеся символы, слипшиеся символы, левое крепление и разделитель региона для компонент в номере, разделитель региона и правое крепление для компонент в регионе.

- Перебор гипотез.

Гипотезой будем называть упорядоченный набор прямоугольников, соответствующий предполагаемому положению символов на изображении номерного знака, а также идентификатор предполагаемого типа номерного знака. Под типом номерного знака подразумевается, относится ли номерной знак к частному автомобилю, к такси, к транспортному средству МВД РФ и т.д.

На вход подаётся множество прямоугольников, полученное на прошлом этапе.

На выходе – упорядоченный набор гипотез, ранжирование в котором производится по степени уверенности в том, что эта гипотеза соответствует истине.

Проблемы, которые нужно решить на этом этапе: чрезмерное время перебора, определение типа номерного знака, заведомо неверные гипотезы, ранжирование.

Можно рассмотреть алгоритм подробнее, разбив каждый его этап на последовательные подзадачи.

2.2. Бинаризация и препроцессинг

2.2.1. Выравнивание изображения по яркости

На вход подаётся горизонтально повёрнутое приведённое к размеру 136x39 пикселей изображение номерного знака.

Для выравнивания изображения по яркости был использован реализованный в OpenCV метод `equalizeHist` [8].

2.2.2. Адаптивная бинаризация



Рис. 1: Исходные изображения номерных знаков, изображения после эквализации с отмеченными разделителями регионов, бинаризованные изображения.

Изображение, полученное на прошлой стадии, необходимо привести к двоичному виду.

Для этого был использован реализованный в OpenCV метод `adaptiveThreshold` [9]. При использовании адаптивной бинаризации перепады яркости в пределах одного изображения не являются помехой (Рис. 1).

Подбором были выбраны подходящие параметры бинаризации:

- `adaptiveMethod = ADAPTIVE_THRESH_MEAN_C`
- `blockSize = 11`
- `C = 20`

2.2.3. Удаление границ номерного знака

Левая и правая границы номерного знака на двоичном изображении иногда приклеиваются к ближайшим символам за счёт левого и правого механизма крепления соответственно. Для таких ситуаций был предусмотрен алгоритм, по мере возможности удаляющий рамки и крепления, если они образовали одну компоненту связности с символом.

В каждой точке левого края изображения запускается рекурсивный алгоритм, который сначала считает высоту вертикального столбца из чёрных пикселей, продолжающегося вверх и вниз из этой точки. Если эта высота не превышает высоту столбца на прошлом шаге рекурсии, то текущий столбец удаляется, то есть заполняется белым цветом, а после этого рекурсивный алгоритм запускается из точек, что расположены правее точек удалённого столбца. Предусмотрен механизм, не допускающий многократный пересчёт высоты одного и того же столбца.

Аналогичный рекурсивный алгоритм с противоположным направлением распространения запускается и на правом крае изображения.

Такой подход позволяет избавиться от рамок тогда, когда номерной знак хорошо подогнан под свои границы. Если же номерной знак занимает меньшую часть изображения, то рамка не находится на краях изображения, и попытки её удалить где-то во внутренней части изображения будут нести риск того, что будет удалён какой-либо значимый символ.

Кроме того, такое стирание чёрных пикселей на краях в частных случаях решает проблему левого крепления и иногда понижает число рассматриваемых компонент связности, что особенно важно для сегментации региона.

В дальнейшем можно улучшить удаление границ за счёт целенаправленного поиска большой компоненты связности на изображении окрестностей номерного знака.

2.3. Подготовка символьных компонент

2.3.1. Поиск всех компонент связности

На бинарном изображении, полученном после удаления границ номерного знака, производится поиск всех компонент связности.

При этом компоненты связности наращиваются переходами по 4-м направлениям на изображении, то есть переходы по диагоналям не используются.

Каждой компоненте связности ставится в соответствие прямоугольник, который вмещает эту компоненту.

2.3.2. Удаление неподходящих компонент

На этом этапе в качестве изображения выбирается сокращённое изображение номерного знака: только вырезанная картинка номера для сегментации номера, только вырезанная картинка региона для сегментации региона.

Из полученных на прошлом этапе прямоугольников удаляются те, которые заведомо не могут соответствовать компоненте связности настоящего символа, а также те, которые не дадут никакой пользы при распознавании.

При этом условия для откидывания прямоугольников не должны быть слишком строгими, чтобы была возможность в дальнейшем восстановить прямоугольники, соответствующие истинному положению символов.

Удаление прямоугольников может базироваться как на внутренних параметрах прямоугольника (например, на его ширине и высоте), так и на внешних (соотношения параметров прямоугольника и параметров изображения).

Удаление компонент следует проводить до попыток восстановления, потому что в противном случае число символьных компонент может значительно возрасти, и при этом будет совершенно непонятно, какие компоненты следует рассматривать, а какие – нет.

Введу обозначения:

rectangle.width – ширина прямоугольника в пикселях,

rectangle.height – высота прямоугольника в пикселях,

image.width – ширина изображения в пикселях,

rectangle.area – площадь прямоугольника в квадратных пикселях,

image.area – площадь изображения в квадратных пикселях.

Итоговый набор альтернативных условий для удаления прямоугольника в номере оказался таков:

- $rectangle.width < 5 \ || \ rectangle.height < 7$

Эти условия выводят из рассмотрения мелкие компоненты, которые чаще всего

соответствуют шумам, механизмам крепления, распавшимся границам номерного знака.

- $\frac{rectangle.width}{rectangle.height} > 4 \parallel \frac{rectangle.height}{rectangle.width} > 8$

Эти условия выводят из рассмотрения излишне непропорциональные компоненты. При этом условия выбраны так, чтобы пропустить возможно слипнувшиеся символы.

- $\frac{rectangle.width}{image.width} > \frac{1}{3} \parallel \frac{rectangle.area}{image.area} > \frac{1}{4}$

Первое условие в основном удаляет верхнюю и нижнюю рамки, а второе условие удаляет уголки – компоненты, составленные из вертикальных частей рамки и горизонтальных.

Максимальная возможная ширина прямоугольника регулируется 3-й группой условий, в то время как максимальная высота ничем не ограничена. Это связано с тем, что существуют номера, где некоторые символы занимают почти всю высоту изображения. Добавление условия на максимальную высоту лишь ухудшит работу системы.

2.3.3. Разделение компонент

Разделение компоненты на 2 части происходит в том случае, если:

$$\frac{rectangle.width}{rectangle.height} > 1.5$$

При этом возникает 2 новых прямоугольника. Старый прямоугольник при этом откидывается, потому что даже для широких одиночных символов с прилипшей грязью или креплением такая пропорция на практике не наблюдалась.

Было опробовано 2 метода разделения символа:

- Разделение компоненты в ближайшей к центру точке, где вертикальная проекция (на рассматриваемом прямоугольнике) минимальна.
- Разделение компоненты строго по центру.

Первый способ казался наиболее правильным, но на деле оказалось, что минимум проекции совсем не обязательно достигается в месте склеивания (Рис. 2).



Рис. 2: Номерные знаки со склеившимися символами.

Поэтому было решено использовать разделение компоненты строго по центру. В дальнейшем можно опробовать комбинированный подход: зафиксировать область в окрестности центра прямоугольника, и в ней уже искать минимум проекции.

В данный момент реализовано разделение 2-х слипнувшихся символов. Случаи, когда число объединившихся символов достигает 3-х или более, практически не представлены, поэтому пока что специально не обрабатываются. В регионе слияния более чем 2-х компонент происходит чаще, поэтому там предусмотрено многократное разделение.

2.3.4. Склеивание компонент

Во время бинаризации некоторые символы могли разделиться на несколько компонент связности (Рис. 3). Встречаются случаи, когда отделение части символа не является проблемой, потому что прямоугольник, в котором располагается компонента связности, захватывает отколовшуюся часть (например, когда отделяется нижняя часть буквы М на одной из сторон).

Для остальных случаев необходимо предусмотреть механизм склеивания.



Рис. 3: Номерные знаки с разделившимися символами.

Склеивание прямоугольников происходит тогда, когда расстояние между ними меньше 3, и при этом средняя высота склеиваемых прямоугольников больше их суммарной ширины. Это делается для того, чтобы по возможности исключить склеивание компонент в широкие прямоугольники, которые с большой степенью уверенности ошибочны, то есть не соответствуют реальному положению одиночного символа.

При склеивании 2-х компонент сохраняются все 3 участвующие компоненты: 2 исходные и 1 вновь полученная. Полученные с помощью склеивания новые прямоугольники в дальнейшем процессе склеивания не участвуют.

Такой подход показал свою эффективность в решении наиболее частой проблемы расколовшегося символа: раздробления буквы Н посередине.

2.3.5. Улучшение левой компоненты

После бинаризации левый символ номера часто образует одну компоненту связности с левым креплением. Такую проблему нельзя разрешить с помощью ранее рассмотренного механизма разделения компонент.

Чтобы эффективно отсекают левое крепление от компоненты, нужно было определить особенности левого символа. Для распознающихся в данный момент в системе регистрационных номеров частных автомобилей и такси левый символ – буква, то есть символ из строки «АВЕКМНОРСТУХ».

Обычно крепление располагается чуть выше середины левого символа, поэтому символы «ВЕКМНОРС» гораздо чаще слипаются с левым креплением, чем символы «АТУХ». Алгоритм отсечения левого крепления опирается на эту информацию.

Символы «ВЕКМНОРС» на номерах похожи между собой тем, что в их крайних левых частях располагается высокий вертикальный столбец из чёрных пикселей, хотя у символов «ОС» этот столбец и не такой высокий.

Тогда первым шагом по улучшению левой компоненты становится проверка на наличие в прямоугольнике на изображении высокого вертикального столбца. Если такой столбец будет найден, то компоненту можно сокращать.

Эта проверка производится так: считаются вертикальные проекции чёрных пикселей на изображении, и если какая-то проекция превышает $\frac{4}{5}$ от высоты прямоугольника, то считается, что высокий вертикальный столбец есть.

Сокращение производится слева направо, пока не будет достигнута точка, в которой вертикальная проекция превзойдёт некоторый порог. Также сокращение прерывается в тот момент, когда отрезана определённая максимальная часть компоненты.

В качестве порога берётся $\frac{2}{3}$ от максимальной вертикальной проекции чёрных пикселей в прямоугольнике. От прямоугольника максимум отрезаем 0.3 ширины. Такие параметры оказались оптимальными.

При таком подходе может быть повреждена целая буква Т, находящаяся слева. Поэтому перед отрезанием левого крепления проводится проверка на наличие в прямоугольнике слева по центру чёрных пикселей.

Символы «ВЕКМНР» при запуске отрезания никак не повреждаются, если левое крепление к ним не прилипло.

В зависимости от масштабирования в символах «ОС» высокий вертикальный столбец может быть, а может и не быть. Поэтому потребовался дополнительный механизм отсечения.

Основой для такого механизма стала информация о том, что в символах «ОС» в середине вверху и в середине внизу находятся чёрные пиксели.

Цикл движется слева направо по прямоугольнику левой компоненты и считает остаточную ширину прямоугольника. После этого производится проверка цвета пикселей в середине вверху и в середине внизу в этом остаточном прямоугольнике. Если будет найдена точка, для которой эти пиксели чёрного цвета, то отрезание пройдёт по этой точке. Если же такая точка найдена не будет, или если перебор зайдёт за порог в 0.3 ширины картинки, то считается, что отрезать ничего не надо.

Такой подход позволил эффективно отрезать от символов «ОС» левое крепление

в тех случаях, когда первая стадия отсечения не работает. Другие символы при этом не портятся.

Результаты работы алгоритма отсечения левого крепления можно увидеть на примере (Рис. 4).

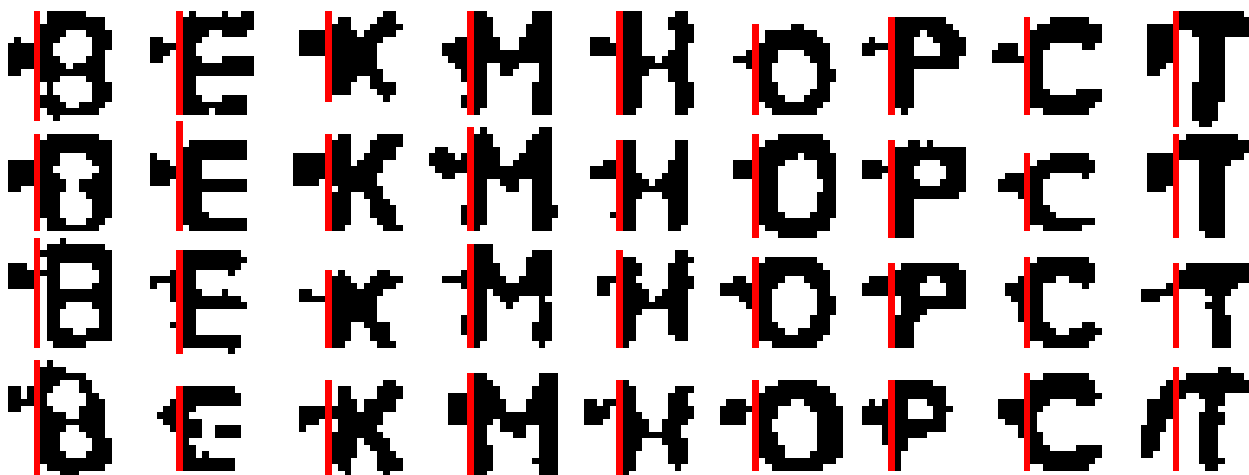


Рис. 4: Отсечение левого крепления, красным отмечена линия отсечения.

Был опробован ещё один способ отсечения левого крепления. Цикл проходил по компоненте слева направо, пока вертикальная проекция не сменит возрастание на убывание, а затем снова на возрастание (нестрогие). Этот метод опирался на информацию о том, что крепление в виде шурупа имеет круглую форму, а поэтому его двоичная проекция будет сначала возрастать, а потом убывать. Такой подход из-за шумов и низкого разрешения изображения оказался не так эффективен, как метод с вертикальным столбцом.

Назову улучшением компоненты процесс приведения границ прямоугольника к реальным границам символа.

Чтобы улучшить компоненту левого символа, нужно знать, где она располагается. Этот вопрос просто не решить: на бинарном изображении могут присутствовать лишние компоненты, которые не соответствуют никакому действительному символу.

В таком случае единственным возможным вариантом на данном этапе становится просто улучшение самой левой компоненты по центру изображения. Если эта компонента соответствует реальному символу, то улучшение даст положительный вклад в дальнейший процесс поиска гипотез и в распознавание. Если же компонента не соответствует никакому символу, то её модификация не принесёт ощутимых отрицательных эффектов.

Нельзя заранее улучшать компоненты, находящиеся правее самой левой компоненты, потому что можно повредить цифры, к которым алгоритм отсечения левого крепления не приспособлен. Только получив гарантию, что компонента не соответствует цифре, можно попытаться её улучшить. Такую гарантию можно получить в

тот момент, когда фиксируется какая-то гипотеза. Поэтому улучшение левого символа запускается ещё и при выборе гипотезы для самого левого прямоугольника гипотезы.

Примеры получившихся к концу этого этапа символьных компонент номера можно увидеть на рис. 5.

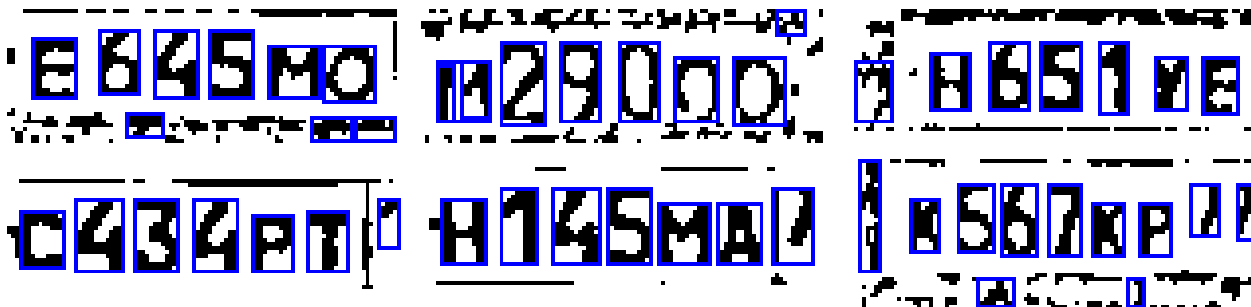


Рис. 5: Символьные компоненты для перебора гипотез в номере.

2.4. Перебор гипотез

Когда символьные компоненты выбраны, необходимо перебрать их конфигурации и найти наиболее похожие на истину.

Перебор осуществляется рекурсивно. Один за другим из списка отсортированных слева направо прямоугольников выбирается следующий и добавляется к строящейся гипотезе.

2.4.1. Снижение времени перебора

Если не ограничивать число символьных компонент, то рекурсия может занять непозволительно долгое время.

Поэтому перед запуском перебора происходит удаление компонент, пока число компонент не станет меньше установленного порога. В качестве максимально допустимого числа компонент выбрано значение 15.

Удаление компонент можно производить разными способами:

- Удалять те компоненты, чьи центры по вертикали дальше всего от центра масс всех компонент по вертикали.
- Удалять те компоненты, чьи центры по вертикали дальше всего от центра изображения по вертикали.
- Удалять компоненты, начиная с самых мелких.
- Найти промежуток по вертикали, на котором находятся компоненты, и удалять самые дальние от центра этого промежутка компоненты.

Первый способ был плох тем, что иногда центр масс сильно смещался в сторону мелких компонент, что приводило к необратимому удалению истинных компонент.

Третий способ плохо работает на тех изображениях номерного знака, где номерной знак меньше предполагаемого размера. Тогда могут быть ошибочно удалены компоненты действительных символов.

Четвёртый способ плох тем, что промежуток может смещаться в одну из сторон номерного знака.

В итоге был выбран второй способ удаления компонент.

2.4.2. Определение типа номерного знака

Система должна была различать государственные регистрационные знаки частных автомобилей и такси, при этом сам формат номерного знака у этих категорий различается: «a000aa|000» или «a000aa|00» для частного автомобиля, «aa 000|00» для такси.

У частного автомобиля в номере шесть символов, а у такси – пять. Поэтому при построении гипотезы в рекурсии при достижении пяти прямоугольников нужно сделать выбор в сторону одной из этих категорий.

Для выбора между двумя категориями были использованы следующие отличия между форматами номеров:

- У такси на втором месте находится буква, а у частного авто – цифра.
- У такси на пятом месте находится цифра, а у частного авто – буква.
- У такси между вторым и третьим символами находится широкий пробел.

Функция заполнения типа номерного знака в гипотезе работает следующим образом.

Сначала находится средняя высота по всем пяти уже выбранным прямоугольникам в гипотезе.

Затем второй и пятый прямоугольники сравниваются по высоте со средней высотой. Второй прямоугольник должен быть ниже средней высоты, а пятый должен быть выше, чтобы гипотеза соответствовала такси.

После этого находится максимальная ширина пробела между соседними прямоугольниками, за исключением пробела между 2-м и 3-м символами. Чтобы гипотеза соответствовала такси, необходимо, чтобы пробел между 2-м и 3-м символами был больше либо равен 1.75 максимального пробела между остальными символами.

В итоге гипотеза считается соответствующей такси, если выполнены все 3 условия. В остальных случаях гипотеза соответствует частному авто. Такая строгая необходимость исполнения всех условий оправдывает себя, потому что такси проезжает зна-

чительно меньше, чем частных автомобилей. При этом определение типа номерного знака работает весьма эффективно.

В дальнейшем можно расширить функцию определения номерного знака на другие типы регистрационных номеров.

2.4.3. Улучшение правой компоненты

Когда тип номерного знака гипотезы установлен, и гипотеза получила последний необходимый до завершения прямоугольник, можно попытаться улучшить правую компоненту, которая могла исказиться из-за слияния с остатками разделителя региона.

Задача усложняется тем, что форма искажения не настолько понятна, как в случае с левым креплением: разделитель региона может быть представлен небольшой компонентой, а может протянуться до верхней или нижней рамки. Случаи объединения с разделителем региона редки, так что следует поставить строгие условия на необходимость улучшения.

Для улучшения правой компоненты было решено использовать информацию о других прямоугольниках в гипотезе. По этой информации строится прогноз возможной формы последнего прямоугольника, и этот прогноз сравнивается с выбранным прямоугольником.

Прогнозируемый прямоугольник имеет ту же высоту, что и предпоследний прямоугольник в гипотезе. Его ширина устанавливается равной его высоте, чтобы соответствовать самому широкому возможному символу – М. Горизонтальный отступ прогнозируемого прямоугольника равен горизонтальному отступу между предпоследними прямоугольниками в гипотезе, а вертикальная координата нижней грани равна вертикальной координате нижней грани предпоследнего прямоугольника.

Введу обозначения:

actual.width – ширина действительного правого прямоугольника в пикселях,

predicted.width – ширина прогнозируемого правого прямоугольника в пикселях,

$(actual \cap predicted).area$ – площадь пересечения действительного и прогнозируемого правых прямоугольников в квадратных пикселях,

predicted.area – площадь прогнозируемого правого прямоугольника в квадратных пикселях,

actual.area – площадь действительного правого прямоугольника в квадратных пикселях.

После построения прогнозируемого прямоугольника нужно проверить, следует ли улучшать правую компоненту:

- $actual.width \geq predicted.width$

Выполнение этого условия говорит о том, что действительный прямоугольник выходит за границы прогнозируемой ширины.

- $(actual \cap predicted).area \geq predicted.area/2$

Это условие говорит о том, что действительная компонента и прогнозируемая находятся примерно в одной области картинки. Дело в том, что нельзя улучшать правую компоненту, когда гипотеза с ней заведомо ложна. Иначе может получиться гипотеза, которая ложна, но после такого улучшения ещё допустима и правдоподобна.

- $actual.area \geq 1.25 * predicted.area$

Это условие дополняет проверку на ширину и откидывает незначительно отклоняющиеся от предсказанного компоненты.

Когда все три условия выполнены, действительный правый прямоугольник в гипотезе заменяется прогнозируемым.

2.4.4. Проверка допустимости гипотезы

Когда гипотеза сформирована, а входящие в неё прямоугольники улучшены, нужно узнать, может ли такая гипотеза в принципе соответствовать какой-то символической строке. Необходимо заранее откинуть большую часть неподходящих гипотез.

Для решения этой задачи сконструирована функция проверки допустимости гипотезы, объединяющая набор условий, которым должна удовлетворять гипотеза:

- В пробеле между 2-м и 3-м компонентами не должно быть крупных компонент (только для такси).

Это условие в основном откидывает те гипотезы, в которых номера частных автомобилей принимаются за такси с пропуском одного символа в центре.

Для проверки условия строится вспомогательный прямоугольник, чей вертикальный промежуток равен промежутку пересечения вертикальных размерностей компонент, образующих пробел, а горизонтальный промежуток соответствует пробелу.

Все символические компоненты по очереди пересекаются с этим вспомогательным прямоугольником, и если площадь пересечения превышает половину площади второй компоненты, то считается, что есть крупная компонента в пробеле.

- Пересечение вертикальных промежутков компонент гипотезы непусто.

В правильной гипотезе символы должны находиться в одном коридоре по высоте. Если какие-то компоненты находятся в другом коридоре, то, скорее всего, гипотеза ложная.

- Дисперсия вертикальных координат нижних граней прямоугольников не должна превышать определённый порог.

В правильной гипотезе нижние грани символов располагаются примерно на одном уровне (изображение номерного знака было повернуто перед началом сегментации).

В качестве порога для дисперсии было выбрано значение:

$$bottomDispersionThr = 33$$

- Дисперсия координат центров прямоугольников по оси ОХ не должна превышать определённый порог (только для частных автомобилей).

В правильной гипотезе для частного автомобиля символы по горизонтали располагаются примерно равномерно.

В качестве порога для дисперсии было выбрано значение:

$$intermidDispersionThr = 9$$

- Расстояния между центрами прямоугольников по оси ОУ и средним значением центра не превышают определённый порог.

Такое условие проверяется отдельно для тех прямоугольников, которые по гипотезе должны соответствовать цифрам, и тем, что соответствуют буквам.

В качестве порога и для цифр, и для букв было выбрано одинаковое значение:

$$maxOffsetY = 5$$

Такой выбор условий оказался оптимальным для откидывания большей части ложных гипотез, возникающих из-за присутствия в символьных компонентах ложных компонент (Рис. 6).

2.4.5. Оценка правдоподобности гипотезы

Все допустимые гипотезы следует оценить по степени уверенности в том, что данная гипотеза соответствует символьной строке.

Для решения этой задачи сконструирована функция оценки. Эта функция собрала воедино все критерии, по которым гипотеза получает оценочные баллы:

- Проверка ожидаемой высоты цифр и букв.

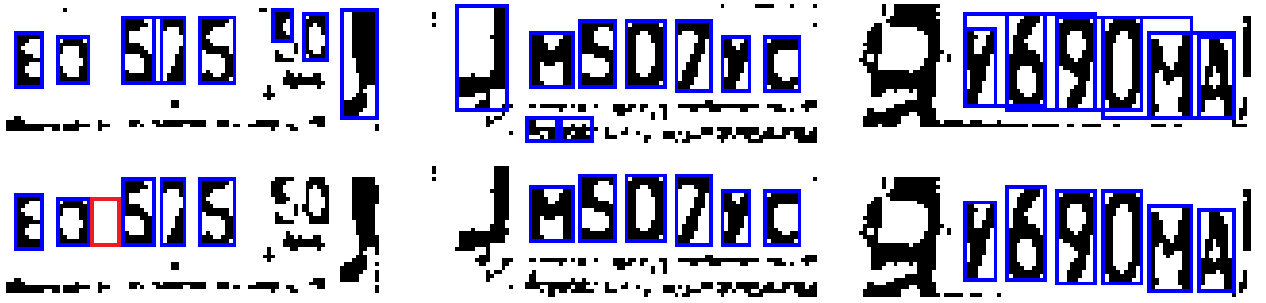


Рис. 6: Верхний ряд – все символьные компоненты нескольких различных номеров, нижний ряд – единственная допустимая гипотеза из этих компонент.

Сначала считается средняя высота всех прямоугольников в гипотезе. От прямоугольников, соответствующих букве, ожидается, что они будут ниже этой средней высоты. Прямоугольники для цифр должны быть выше средней высоты.

За каждый соответствующий ожиданиям прямоугольник гипотезе зачисляется балл.

- Сравнение максимальной ширины прямоугольника и минимальной.

Если максимальная ширина не более чем в 2 раза превышает минимальную, то гипотезе зачисляется балл.

- Расположение прямоугольников на одном уровне.

Находится среднее положение нижней грани прямоугольников.

Гипотеза получает баллы за те прямоугольники, нижняя грань которых отступает от среднего значения не более чем на:

$$offsetY = 3$$

- Равномерные отступы между прямоугольниками.

Находится средняя ширина пробела между соседними символами.

Гипотеза получает баллы за те пробелы между прямоугольниками, которые отличаются от среднего значения менее чем на:

$$offsetX = 6$$

Такая оценочная функция показала хорошие результаты для задачи сортировки гипотез.

В дальнейшем можно было бы улучшить оценочную функцию, добавив новые критерии оценивания. Также можно старые критерии брать с определёнными весами. От

целочисленной реализации оценочной функции можно перейти к дробной нормированной.

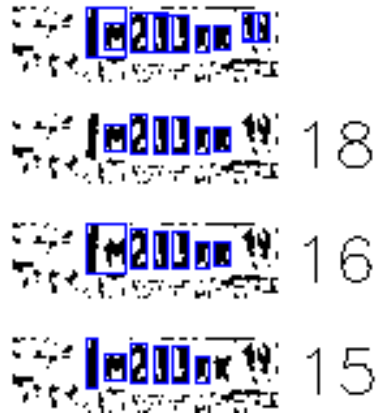


Рис. 7: Все символьные компоненты номера и отсортированные гипотезы со своими оценками.

Во время перебора гипотез остаётся лишь замечать избранные гипотезы более правдоподобными. В конце перебора следует отсортировать набор избранных гипотез, и сегментация будет завершена (Рис. 7).

3. Эксперименты

Для проведения экспериментов было необходимо определить формат наборов данных и разметить их.

Детекция – это ровно одно изображение номерного знака. Когда транспортное средство проезжает перед камерой, множество детекций, соответствующих этому транспортному средству, объединяется в проезд.

На автостраде была запущена система, собравшая большое количество проездов, которые были объединены в несколько наборов данных (Табл. 1). После этого к каждому проезду была добавлена информация об истинном значении символьной строки номерного знака автомобиля.

Набор данных	Число детекций	Число проездов
Day1	5924	380
Night	2075	112
Day2	4490	176

Таблица 1: Размеченные наборы данных.

Детекция была распознана верно, если система автоматической идентификации сопоставила детекции истинное значение символьной строки.

Проезд был распознан верно, если по результатам распознавания каждой из детекций этого проезда функция выбора среднего значения символьных строк выдала истинное значение символьной строки проезда.

При этом можно сузить символьную строку только до региона или только до номера, чтобы протестировать сегментации номера и региона отдельно.

Об успешности системы можно судить по двум метрикам: по проценту верно распознанных детекций и по проценту верно распознанных проездов.

Набор данных	Верных проездов, %	Верных детекций, %
Day1	62.11	71.00
Night	93.75	79.33
Day2	93.18	73.54

Таблица 2: Старый алгоритм сегментации, успешность по номерам.

Набор данных	Верных проездов, %	Верных детекций, %
Day1	88.95	85.97
Night	97.32	82.27
Day2	97.73	82.67

Таблица 3: Предложенный алгоритм сегментации, успешность по номерам.

Система была запущена на размеченных наборах данных со старым алгоритмом

сегментации и с предложенным. Были получены показатели успешности системы по обоим метрикам с разными алгоритмами сегментации (Табл. 2, 3).

При переходе к новому алгоритму сегментации возрасла успешность распознавания по отдельным детекциям. В результате улучшилось и среднее значение по символическим строкам – выросло число верно распознанных проездов.

Значительно улучшилась работа системы на наборе данных Day1, который содержал очень много проблемных детекций, на которых старый алгоритм сегментации работал плохо.

Алгоритм сегментации номера был адаптирован под сегментацию региона, и успешность системы по регионам и общая успешность по всему номерному знаку также возросли.

Заключение

В рамках данной работы были получены следующие результаты.

- Предложен алгоритм сегментации, эффективно работающий на зашумлённых изображениях.
- Алгоритм сегментации реализован в рамках системы автоматической идентификации номеров.
- Были подготовлены наборы данных для тестирования.
- Проведено тестирование нового и старого подходов: было показано, что качество работы системы при использовании нового алгоритма сегментации значительно лучше, чем со старым алгоритмом.

Все цели были достигнуты, а задачи выполнены.

В качестве дальнейшей работы рассматривается тестирование и, возможно, использование тех идей, что были указаны в тексте. Также планируются дальнейшие работы по улучшению сегментации региона.

Список литературы

- [1] Approach to recognition of license plate numbers using neural networks / I. Paliy, V. Turchenko, V. Koval et al. // IEEE International Joint Conference on Neural Networks, 2004. — Vol. 4. — P. 2965–2970.
- [2] Automatic License Plate Recognition (ALPR): A State-of-the-Art Review / Shan Du, M. Ibrahim, M. Shehata, W. Badawy // IEEE Transactions on Circuits and Systems for Video Technology. — 2013. — Vol. 23(2). — P. 311–325.
- [3] Automatic license plate recognition / Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, Sei-Wan Chen // IEEE Transactions on Intelligent Transportation Systems. — 2004. — Vol. 5(1). — P. 42–53.
- [4] Building an Automatic Vehicle License-Plate Recognition System / Tran Duc Duan, Tran Le Hong Du, Tran Vinh Phuoc, Nguyen Viet Hoang // Intl. Conf. in Computer Science – RIVF’05. — 2005. — P. 59–63.
- [5] Capar A., Gokmen M. Concurrent Segmentation and Recognition with Shape-Driven Fast Marching Methods // 18th International Conference on Pattern Recognition, 2006. ICPR 2006. — Vol. 1. — P. 155–158.
- [6] Feature based recognition of traffic video streams for online route tracing / C. Busch, R. Domer, C. Freytag, H. Ziegler // Vehicular Technology Conference, 1998. VTC 98. 48th IEEE. — Vol. 3. — P. 1790–1794.
- [7] Kim Kwang-Baek, Jang Si-Woong, Kim Cheol-Ki. Recognition of Car License Plate by Using Dynamical Thresholding Method and Enhanced Neural Networks // Computer Analysis of Images and Patterns. — 2003. — P. 309–319.
- [8] OpenCV. Histogram Equalization // OpenCV documentation. — URL: http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html (online; accessed: 23.05.2015).
- [9] OpenCV. Miscellaneous Image Transformations // OpenCV documentation. — URL: http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html (online; accessed: 23.05.2015).
- [10] Prabhakar P., Anupama P. A novel design for vehicle license plate detection and recognition // ICCTET 2014. — P. 7–12.
- [11] Rahman C.A., Badawy W., Radmanesh A. A real time vehicle’s license plate recognition system // IEEE Conference on Advanced Video and Signal Based Surveillance. — 2003. — P. 163–166.

- [12] Vehicle license-plate recognition by image analysis / K. Miyamoto, K. Nagano, M. Tamagawa et al. // IECON '91. — Vol. 3. — P. 1734–1738.