

УДК 004.932.72

## АЛГОРИТМ ШУМОФИЛЬТРАЦИИ ВЕРТИКАЛЬНОЙ ПРОЕКЦИИ СИМВОЛЬНОЙ СТРОКИ

**Д. В. ЗАЕРКО**

магистр физико-математических наук, аспирант

Белорусский государственный университет информатики и радиоэлектроники

**В. А. Липницкий**

доктор технических наук, профессор

Военная академия Республики Беларусь

*Распознавание символьных строк является важной частью проблематики распознавания изображений. Для сегментации зашумленной символьной строки, как растрового изображения, стало традиционным использование вертикальной проекции. В работе предлагается дополнить эту процедуру процессом шумофильтрации. Представлено детальное описание специального проекционного фильтра, как основного инструмента шумофильтрации. Совместно с анализируемой проекцией рассматриваются модельные и скелетная проекции, обеспечивающие эквивалентную замену зашумленных фрагментов и хранящиеся в специализированных базах. Представлен процесс создания баз и обогащения их новыми данными. Созданный алгоритм шумофильтрации реализован на программном языке Java и встроен в библиотеку Java ANPR. Разобран модельный пример фильтрации зашумленной строки, на основе использования этой библиотеки.*

**Ключевые слова:** растровые изображения, вертикальная и горизонтальная проекции, база проекций, алфавитная база проекции, средняя ширина символьной проекции, границы проекции, шумофильтрация символьных строк, оценка допустимого искажения.

### Введение

Со времени возникновения цифровых систем обработки, передачи и хранения информации появилась и проблема надежности этих систем, достоверности их работы. Возникавшие сложности постепенно обобщились в единую проблему распознавания сигналов и изображений [1; 2; 3]. Оба направления имеют уже солидную историю, свои, существенно различные, методы и средства исследований.

Распознавание номерных знаков транспортных средств относится к проблематике распознавания изображений. На первый взгляд, номера эти носят линейную и строго ограниченную структуру. Реально же мы имеем дело, как правило, с некачественным двумерным снимком исследуемого номера. Задача заключается в максимальном приближении изображаемого номера к реальному качественному снимку. Имеем классическую задачу восстановления изображения [4].

Технологии распознавания номерных знаков автомобилей востребованы на современных автомагистралях. Она является основой контроля автомобильных потоков и автоматического отслеживания правонарушений. Технические средства, задействованные для визуального наблюдения, зачастую используют системы распознавания, несовершенные в алгоритмическом плане и в плане безошибочности распознавания номерных знаков. Последнее особенно актуально при обработке изображений, выполненных в неблагоприятных погодных условиях, которые негативно влияют на качество изображения и вид самого номерного знака.

© Заерко Д. В., 2020

© Липницкий В. А., 2020

Весь процесс распознавания можно разбить на отдельные, но тесно взаимосвязанные этапы, не успешность одного из которых повлечет серьезные ошибки или невозможность выполнения последующих этапов.

Сегментация и коррекция символов являются одним из фундаментальных этапов распознавания номерных знаков, т. к. остальные этапы полностью зависят от полученных результатов. Если сегментация выполнена неуспешно, то анализируемый символ может быть, например, разделен на две или более частей, что делает невозможным сопоставить отдельные части и, как следствие, провести дальнейшую обработку.

Несомненно, что этап очистки от шумов на этапе анализа строки был бы очень полезен, т. к. позволял бы избегать дальнейших трудоемких и бессмысленных действий на последующих этапах.

В научной литературе слабо представлены алгоритмы, используемые на этапе сегментации символов из символьной строки [3, 5 – 7], а очистка от помех на этапе сегментации и вовсе сводится лишь к отделению границ номера от символьной строки.

Уточним, что современный номерной автомобильный знак содержит кроме символьной строки еще и другие элементы, такие как рамка номерного знака, номер региона или области и т. д. (см., например, рис. 1).

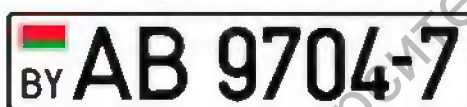


Рис 1. Номер автомобиля из Республики Беларусь

Существует несколько способов сегментации однострочных номеров. Наиболее часто используемые – это методы горизонтальной проекции и сегментация на их основе. Для однострочных номеров сегментация представляет собой способ нахождения границ между символами по горизонтали. Сегменты реального изображения номера содержат, кроме ожидаемых символов, еще и шумы – элементы, такие как точки, символы знаков тире между символами, детали, не относящиеся к номерному знаку. Следующие фазы после сегментации – это “усиление” сегментов и непосредственное распознавание символов строки, однако, эти фазы возможны только в случае корректного завершения фазы сегментации символьной строки.

Из вышесказанного следует, что наиболее перспективным и многообещающим будет изучение символьной строки, представляющей собой набор монохромных пикселей, из которых состоят отдельные символы, а также проекции этой строки на ось  $Ox$ . Они и будут главными объектами нашего исследования. Предметом изучения будет описание и применение алгоритма проекционной фильтрации монохромной, пиксельной строки на этапе ее построения.

Таким образом, первоочередным и фундаментальным этапом является этап сегментации с последующей очисткой сегмента от помех. В статье, представленной ниже, подробно рассмотрен метод горизонтальной проекции, а также представлен один из методов очистки сегментов символьной строки от шумов на этапе построения проекции. Предложенный метод впервые вводит в рассмотрение очистку отдельных сегментов символьной строки, представленных в виде фрагментов проекции, а не только отделение анализируемой строки от ее границ, что было рассмотрено прежде [8].



5) символьная строка может быть любой длины, однако, условимся, что она конечна.

Горизонтальная проекция позволяет выполнить сегментацию посимвольно, основываясь на определении пробелов между символами. Адаптивный порог используется для разделения темных областей на переднем плане от светлого фона с неравномерным освещением. Учитывая пороговое значение, можно вычислить горизонтальную проекцию  $p_f(x)$  для номера  $f(x, y)$ . Использование проекции позволяет определить горизонтальные границы между сегментированными символами.

Основная цель алгоритма сегментации – это поиск пиков (локальных экстремумов) функции  $p_f(x)$ , которые связаны с пробелами между символами. Охарактеризуем несколько важных значений на графике горизонтальной проекции  $p_f(x)$ :

–  $v_m$  – максимальное значение горизонтальной проекции, определяемое как  $v_m = \max_{0 \leq x < w} \{p_f(x)\}$ , где  $w$  это ширина номера в пикселях;

$v_a$  – усредненное значение горизонтальной проекции  $p_f(x)$ , рассчитываемое как  $v_a = \frac{1}{w} \sum_{x=0}^{w-1} p_f(x)$ ;

–  $v_b$  – значение, используемое как базовое для оценки вертикальных пиков; базовое значение  $v_b$  – всегда рассчитывается как  $v_b = 2v_a - v_m$ ; разумеется, значение  $v_a$  должно лежать на вертикальной оси между значениями  $v_b$  и  $v_m$ .

Итерационный алгоритм сегментирования найдет максимумы в графике функции вертикальной проекции. Пики представляют собой границы промежутков – разрывов между символами. После нахождения пиков производится обнуление пиков и итерационное повторение алгоритма до тех пор, пока не будет найден глобальный экстремум. Все вышеизложенное может быть представлено в виде следующего пошагового алгоритма.

1. Определение индекса максимального значения горизонтальной проекции  $x_m = \arg \max_{0 \leq x < w} \{p_f(x)\}$ .
2. Определение левой и правой границы пиков (поиск минимального и максимального значения проекции относительно локального максимума в  $x_m$ ):

$$x_l = \max_{0 \leq x \leq x_m} \{x \mid p_f(x) \leq c_x p_f(x_m)\};$$

$$x_r = \min_{x_m \leq x < w} \{x \mid p_f(x) \leq c_x p_f(x_m)\}.$$

3. Сегментация номера по горизонтали по точкам  $x_m$ .
4. Обнуление вертикальной проекции  $p_f(x)$  на интервале  $\langle x_l, x_r \rangle$ .
5. Если  $p_f(x_m) < c_w v_m$ , алгоритм оканчивается (найлены все пики на графике проекции), т. е. переходим на шаг 7. Проверяем, является ли точка  $x_m$  значимой, т. е. не относится к разряду шумов.
6. Переход к шагу 1.
7. Окончание алгоритма.

В приведенном алгоритме участвовали, в частности, две следующие никак не отмеченные ранее константы:

$c_x$  – используется для определения границ пиков  $x_m$  (оптимальное  $c_x = 0.7$ ). Другими словами  $c_x$  позволяет оптимально определить левую и правую границы относительно локального максимума  $p_f(x_m)$  в точке  $x_m$ . Под оптимальным значением имеется в виду средняя величина, полученная статистическими методами после анализа множества изображений [8]. Однако эта величина, как и другие “оптимальные”, может иметь иные значения в зависимости от качества обрабатываемых изображений.

$c_w$  – определяет минимальную высоту пика, связанную с максимальным значением проекции  $v_m$ , если высота пика ниже этого минимума, то пик не будет считаться как пробел между символами.

Очень важным является выбор значения константы  $c_w$ . Чрезмерно малое значение указывает на множество пиков, которые будут представлены как пробелы и, как следствие, символы будут разбиты на части. Чрезмерно большое значение  $c_w$  приведет к тому, что не все пики будут рассматриваться как пробелы, и символы будут склеены между собой. Наиболее приемлемое значение  $c_w = 0.86$  [8]. Чтобы обеспечить правильное поведение алгоритма эти константы должны удовлетворять следующему соотношению

$$\begin{cases} \forall(x_l, x_m, x_r) \in P : c_w v_m > p_x(x_l), \\ \forall(x_l, x_m, x_r) \in P : c_w v_m > p_x(x_r), \end{cases}$$

где  $P$  – это множество всех пиков  $x_m$  со связанными пиками  $x_l, x_r$ .

## 2. Зашумленность символьной строки.

### Анализ и фильтрация проекции символьной строки

Проекция растровых изображений часто используется для структурного анализа изображений с целью выявления закономерностей, которые в свою очередь будут использоваться при решении практических задач распознавания.

Растровая проекционная фильтрация предполагает анализ фрагментов проекции на наличие шаблонов, т. е. наборов последовательно повторяемых признаков на фиксированном фрагменте проекции. Под признаками будем иметь в виду последовательность максимумов и минимумов проекции или совпадение множества пикселей между чередующимися экстремумами, с некоторой допустимой погрешностью. Подчеркнем то, что использование фильтров будет возможно только в случае наличия не отдельных признаков, а в случае наличия целого их спектра.

Уточним понятие шума, а также признаков, указывающих на наличие его в фиксированном фрагменте проекции. Понятие шума может существенно изменяться в зависимости от начальных данных (исходной символьной строки) и от постановки задачи (воздействующих условий).

Далее, для примера, проанализируем возможные “шумы” на номерных автомобильных знаках. В контексте данной работы, под шумами с физической точки зрения будут рассматриваться: пятна грязи, т. е. песок, мазут и т. д. (рис. 3), царапины и трещины на полотне знака (рис. 4), выцветшая или отслоившаяся краска символов (рис. 5). На изображении эти шумы могут быть представлены в виде смазанных символов по вертикали и горизонтали (потoki грязи); фрагмента символа, отдельных точек в составе символа, четкого контура символа, но с выцветшим центром до цвета заднего фона (выцветание и царапины). Исходя из вышесказанного, простейшими составляющими фрагментами шума на изображении являются: несвязанные, отдельно стоящие точки, вертикальные линии, выступающие по размеру относительно большинства символов и не выступающие за границы номера, а также горизонтальные линии, впереди и позади которых нет символов номера. Все три типа шума могут быть выявлены путем анализа экстремумов вертикальной проекции с последующим составлением их портрета или шаблона (набора обязательно присутствующих признаков).

Следующий ключевой этап после анализа проекции на наличие шумов в нашем исследовании – это этап корректировки проекции. Этап корректировки предполагает преобразование исходного зашумленного фрагмента проекции к фрагменту без шума, путем изменения графика вертикальной проекции. Рассмотрим два примера возможных зашумленных символьных строк.



Рис 3. Пятна грязи на номерном знаке



Рис 4. Царапины на номерном знаке



Рис 5. Отслоившиеся символы на номерном знаке

В первом случае рассмотрим ситуацию, при которой один из знаков, а именно, символ тире имеет зашумленность, выраженную в виде набора несвязанных точек (рис. 6). Наличие шума может привести к росту экстремумов на этом отрезке (амплитуды проекции) и, как следствие, восприятие этого символа, либо как единое целое с соседними символами: “В-0”, “В-”, “-0”, либо как отдельный, и вовсе не распознаваемый символ алфавита. В случае восприятия как единое целое, алгоритм распознавания сегментированных фрагментов, выполняемый на следующем этапе, может быть легко зациклен, т. к. зачастую он циклический и, в лучшем случае, ограничен числом итераций, зависящим от заданного числа символов в символьных строках, а в худшем, не имеет ограничений как таковых. При нераспознаваемом символе – неверное распознавание номера. Во втором случае проанализируем ситуацию, когда символ тире первый в номерном знаке, что является недопустимым. Следовательно, такой символ не должен восприниматься как символ регистрационного номерного знака, а только как его визуальное оформление (сам номер располагается между знаками тире). Как и в первом примере, распознавание такой символьной строки, как номерного знака на этапе распознавания символов приведет либо к закликиванию, либо к распознаванию лишь части номера.



Рис. 6. Пример проекции монохромного изображения

Опишем следующий набор предположений, или набор признаков, характеризующий особые свойства фрагментов проекции и позволяющий сегментировать и корректировать зашумленные символы.

1. Если проекция на некотором фрагменте имеет ярко выраженный минимум, причем он находится между двумя локальными максимумами, не превышающими другие максимумы более чем на величину  $\epsilon > 0$ , то фрагмент потенциально характеризует отдельный знак, границы которого совпадают с локальными максимумами.

2. Существует предположение, что значение проекции (число пикселей) на определенном фрагменте для отдельного символа может быть четко сопоставлено с **конечным** набором модельных проекций, где каждая модельная проекция характеризует один единственный символ алфавита с некоторой допустимой погрешностью. Следовательно, исходя из вышесказанного, существует возможность, опираясь на множество модельных проекций, различать проекцию самого символа от шума, учитывая некоторую погрешность  $\epsilon \geq 0$ . Тем самым можно проводить замену фрагментов проекции (**коррекцию**) еще на этапе сегментации, либо в случае повторного этапа сегментации, когда этап распознавания, следующий за этапом сегментации, не дал удовлетворительных результатов.

3. Фрагмент проекции, минимум которого значительно превышает все остальные минимумы (порог  $\tau_{\min} > 0$ ) на всем протяжении проекции, потенциально не подлежит распознаванию, в связи с сильной зашумленностью.

Выполнение первого предположения позволяет провести сегментацию символьной строки исходя только из анализа проекции. Выполнение второго предположения позволяет сегментированные фрагменты сопоставить с модельной проекцией и провести отделение проекции шума от проекции символа. Другими словами, выделяются две основные задачи или два этапа: обнаружение зашумленного сегмента проекции и коррекция или замена целого сегмента на некоторый “модельный” сегмент без шума.

### 3. Скелетная и модельная проекции.

#### Алгоритм-коррекция вертикальной проекции

Ключевую роль в фильтрации проекции отдельно взятого сегмента играет конечное модельное множество проекций  $P_{\text{mod}}$ , где модельная проекция – это “размытая” проекция выделенного символа, т. е. такая проекция, которая описывает нечеткую или размытую

пиксельную границу символа, но сохраняющая основной пиксельный скелет (чистую проекцию символа) [9; 10].

Скелетная проекция или чистая проекция – это множество целых значений  $p_{frame}(x_{m,0}), p_{frame}(x_{m,1}), \dots, p_{frame}(x_{m+1,z(m,m+1)})$  функции проекции  $p_{frame}(x)$ , содержащее  $z(m, m+1) \in N$  целых значений проекций символа на сегменте и ограниченных границами символа  $[x_m, x_{m+1}]$ ,  $m \in [0, w-1]$ . Скелетная проекция – это проекция символа без помех, ее строят на основе искусственного шаблона.

Модельная проекция символа – множество значений  $p_{mod}(x_{m,0}), p_{mod}(x_{m,1}), \dots, p_{mod}(x_{m+1,z(m,m+1)})$  функции проекции  $p_{mod}(x)$ , слева и справа ограниченное предполагаемыми границами сегментации  $[x_m, x_{m+1}]$ ,  $m \in [0, w-1]$  и, также как и скелетная, содержащая  $z(m, m+1) \in N$  целых значений. Причем, модельная проекция кроме набора значений характеризует еще и величину  $sum(\epsilon_{mod})$ , где  $\sum_{m=0}^{w-1} \epsilon_{mod}(p_{mod}(x_m)) = sum(\epsilon_{mod}) \geq 0$ .

Эта суммарная оценка допустимого “искажения” проекции символа. Допустимого – в смысле сохранения основных уникальных черт распознаваемого символа на сегменте.

В свою очередь,  $\epsilon_{mod}(p_{mod}(x_m))$  – это допустимое искажение проекции для столбца  $x_m$  (в случае вертикальной проекции) относительно суммы всех не закрашенных пикселей в этом столбце. Предположим далее, что задано не жесткое ограничение “искажения”, т. е. важна суммарная оценка для каждой модельной проекции, а не точное совпадение ограничения в “искажении” для отдельно взятого столбца. Такая постановка вносит определенную свободу проекции, т. е. позволяет охватить более широкий спектр возможных проекций при фильтрации. Аналогично определяется  $p_{real}(x)$  для анализируемой символьной строки.

Очень важным является совпадение числа значений проекции, по которым вычислялась проекция на сегменте  $[x_m, x_{m+1})$ , как для сегмента анализируемой символьной строки, так и для модельного или скелетного сегмента. Иначе модельная проекция будет неприменима для оценки зашумленных сегментов. В связи с тем, что точное число значений проекции  $z(m, m+1)$  на сегменте  $[x_m, x_{m+1})$ ,  $m \in [0, w-1]$  зависит от ширины символа, далее будем использовать ее среднюю величину  $\bar{z}$ . Вычисление средней ширины сегмента проекции для символа – задача, решаемая с использованием простейших статистических методов для анализа вертикальной проекции символьных строк. Для статистического анализа начальные и конечные точки сегментов символьной строки могут быть извлечены из набора тестовых изображений. Важными характеристиками изображений для формирования статистической выборки являются их одинаковое разрешение, а так же однотипность символьных строк. Под однотипностью имеется в виду тот факт, что проводится анализ номерных автомобильных знаков одного государства и одного образца.

В статье [11] представлена пошаговая инструкция анализа и представлен пример вычисления для двух типов номерных знаков, однако аналогичный анализ возможно провести с большим числом типов. В данной статье первый тип представлен номерными знаками Словацкой Республики, где каждая строка разбита, как правило, на  $N = 12$  сегментов. Второй тип – номера Республики Беларусь (образца 2004 г.) – также 12 сегментов. Учитывая значения вариации для первого типа, среднее число значений, достаточное для описания одного сегмента проекции с одним символом  $\bar{z} = 11,28 \approx 11$  (детали расчета смотри в [11]).



Величина  $\mathcal{E}_{\text{mod}}(p_{\text{mod}}(x_{m,0}, x_{m,1}, \dots, x_{m+1,z}))$  – это множество поточечных разностей между множеством значений проекции модельного символа  $p_{\text{mod}}(x_{m,0}, x_{m,1}, \dots, x_{m+1,z})$  и множеством значений скелетной проекции этого же символа  $p_{\text{frame}}(x_{m,0}, x_{m,1}, \dots, x_{m+1,z})$ , где  $z$  – среднее число значений проекции на сегменте

$$\mathcal{E}_{\text{mod}}(p_{\text{mod}}(x_m)) = \mathcal{E}_{\text{mod}}(p_{\text{mod}}(x_{m,0}) - p_{\text{frame}}(x_{m,0}), p_{\text{mod}}(x_{m,1}) - p_{\text{frame}}(x_{m,1}), \dots, p_{\text{mod}}(x_{m+1,z}) - p_{\text{frame}}(x_{m+1,z})). \quad (1)$$

При наличии наборов  $A$  и  $N$  модельных проекций для символов латинского алфавита и арабских цифр можно провести замену зашумленного сегмента проекции на сегмент скелетной проекций. Замена возможна, при условии, что анализируемый фрагмент проекции  $\alpha$  может быть оценен исходя из имеющихся множеств модельных проекций, т. е.  $\alpha \in [A \cup N]$ . Оценка  $\alpha \in [A \cup N]$  предполагает то, что сумма поточечных разностей проекций для анализируемого сегмента меньше или равна наибольшей из всех поточечных сумм модельных проекций. Опишем шаги предлагаемого алгоритма шумофильтрации.

- 1) Если для сегмента проекции  $[x_m, x_{m+1}]$ ,  $m \in [0, w-1]$  выполняется условие  $\sum_{m=0}^{w-1} \sum_{z=0}^z p_{\text{real}}(x_{m,z}) - p_{\text{mod}}(x_{m,z}) = \text{sum}(\mathcal{E}_{\text{real}}) \leq (\mathcal{E}_{\text{mod}})$ , то шаг 2.
- 2) Замена сегмента  $[x_m, x_{m+1}]$ ,  $m \in [0, w-1]$  проекции на скелетную  $p_{\text{frame}}(x_{m,0}), \dots, p_{\text{frame}}(x_{m+1,z})$ . Если  $m \leq w-1$  то шаг 3, иначе шаг 5.
- 3) Переход к следующему сегменту проекции: шаг 1 при  $m = m+1$ .
- 4) Если  $\alpha \in [A \cup N]$ , то переход к следующему множеству модельных проекций для заданного алфавита (символы латинского алфавита и арабские цифры) и шаг 1, иначе сегмент проекции не анализируем:  $\alpha \notin [A \cup N]$  и следует пропустить его, т. е. перейти на шаг 3.
- 5) Завершение алгоритма.

#### 4. Об аспектах формирования словаря модельной и скелетной проекции на основе библиотеки Java ANPR. Средняя ширина символьной проекции

При реализации описанного выше алгоритма фильтрации особую роль играет подготовительный этап. На этом этапе в рамках выбранного множества  $[A \cup N]$  необходимо сформировать численное описание скелетной проекции для символов латинского алфавита (A-Z) и цифр в арабской записи (0-9) и провести запись этих значений проекций в специализированный словарь или базу данных. Рассмотрим подготовительный процесс для распознавания автомобильных номерных знаков.

В качестве хранилища скелетных и модельных проекций выбрана нереляционная, популярная база данных MongoDB. Выбор хранилища был обусловлен следующими преимуществами.

1. Быстродействием. Особенно важен аспект “быстро” чтения данных. Операция чтения будет задействована намного чаще, чем операции записи/изменения/удаления, используемые на этапе формирования.
2. Легкой масштабируемостью, что особенно актуально при большом объеме данных и росте этих данных.
3. Хранением информации в документно-ориентированном виде, что означает отсутствие жесткой схемы документа.
4. Хранения данных в JSON формате.

Для получения численного описания **скелетной или модельной проекций** используем уже имеющиеся алгоритмы в проекте Java ANPR, применим алгоритм построения проекции к “не искаженным” однотипным изображениям символьных строк или к изображениям с искусственным искажением.

Здесь под типом понимается не только однострочность, но и принадлежность к определенному государственному формату. С практической точки зрения очень важным является определение приблизительных границ символа, точнее, границ его проекции на ось  $Ox$  или средней ширины сегмента для заданного типа. Используя вычисленную среднюю ширину сегмента [11], и число предполагаемых сегментов (стандарт номера по его типу задает число символов в строке), можно определить число значений проекции, необходимое для описания одного сегмента и представляющей собой символ. После вычисления провести запись значений проекции символа в базу.

Рассмотрим архитектуру приложения Java ANPR и детальное формирование скелетной и модельной проекции.

Проект Java ANPR построен по принципу распределения обязанностей между пакетами. Каждый пакет (package) содержит набор классов, обеспечивающий выполнение одного из этапов процесса распознавания. Наиболее важные из них:

neuralnetwork – содержит описание нейросети для распознавания символов;

recognizer – отвечает за непосредственную классификацию и распознавание символов;

imageanalysis – отвечает за первоначальную обработку изображения, выделение фрагментов с изображением номера, выравнивание и фрагментацию номера. Содержит классы, отвечающие за построение пиксельной матрицы, построение проекции и поиск пиков. Наибольший интерес представляет класс Plate, содержащий методы для формирования вертикальной проекции и вычисления точек сегментации проекции.

Применяя программные методы к тестовому изображению, и проведя визуальное сопоставление конкретного символа с ассоциированным с ним фрагментом проекции (множество значений проекции), можно сделать соответствующую запись (элемента) фрагмента проекции в БД. Далее этот фрагмент можно использовать, как **скелетную** проекцию символа. Аналогичным образом делается запись элемента фрагмента для **модельной** проекции. Здесь различие заключается лишь во внесении искусственного искажения на тестовое изображение с искусственно заданной величиной этого отклонения. Величина искажения для символа  $\epsilon_{mod}$  вычисляется как разность между скелетной проекцией и модельной. Если суммарная оценка  $sum(\epsilon_{mod})$  удовлетворяет предполагаемой, то проводится запись фрагмента модельной проекции в БД, с привязкой к ассоциированному символу и общему для всего множества модельных проекций  $sum(\epsilon_{mod})$ .

Отметим, что зашумленные символы должны подлежать распознаванию либо имеющимися алгоритмами Java ANPR, либо путем человеческого анализа с последующей привязкой распознанного значения зашумленного символа к набору значений его проекции. Другими словами, предполагается анализ и распознавание искаженного изображения до определенного предела, при котором сохраняются какие-либо признаки или контуры символа.

Таким образом построена программная реализация алгоритма шумофильтрации на языке Java, которая включена в состав пакета Java ANPR.

### 5. Применение алгоритма шумофильтрации при распознавании номерных знаков

Опираясь на алгоритм сегментации, реализуем программный алгоритм, описанный выше, на языке высокого уровня Java. Воспользуемся алгоритмом сегментации,

базирующейся на построении вертикальной проекции, которая представлена в пакете JavaANPR. В общем случае язык описания может быть любым, выбор Java обусловлен наличием готового базового алгоритма, удобством и перспективностью языка [12].

В качестве примера рассмотрим изображение номерного знака на рисунке (рис. 7). Акцентируем наше внимание на сегментации и проекционной фильтрации зашумленных фрагментов самого номера, всецело предоставляя процессы выделения, выравнивания по горизонтали (алгоритм Хафа) и размерной трансформации номерного знака, алгоритмам пакета Java ANPR.



Рис. 7. Зашумленный автомобильный номер и исходный

Учитывая модельные и скелетные представления проекций, реализуем алгоритм, описанный выше в виде псевдокода. Далее применим его для зашумленных изображений, последовательно увеличивая количество зашумленных символов и “степень” допустимого зашумления. Получение результаты представлены далее.

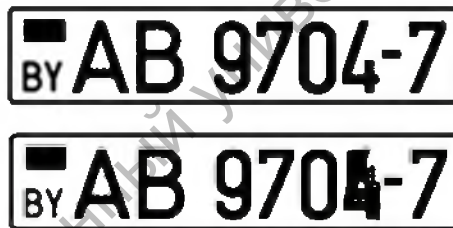


Рис. 8. Монохромное, растровое представление автомобильного номера

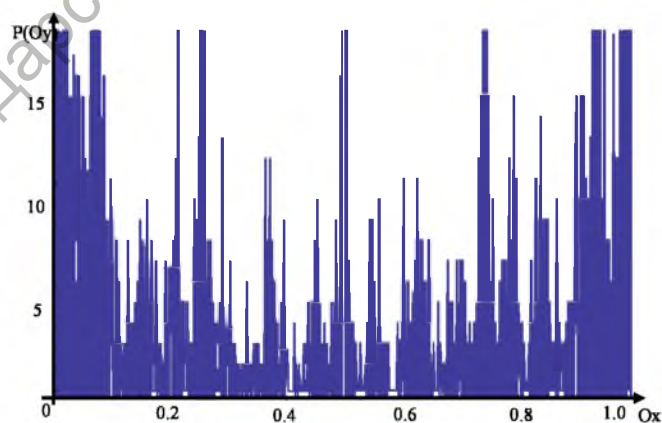


Рис. 9. Проекция строки до применения алгоритма фильтрации, для исходного номера

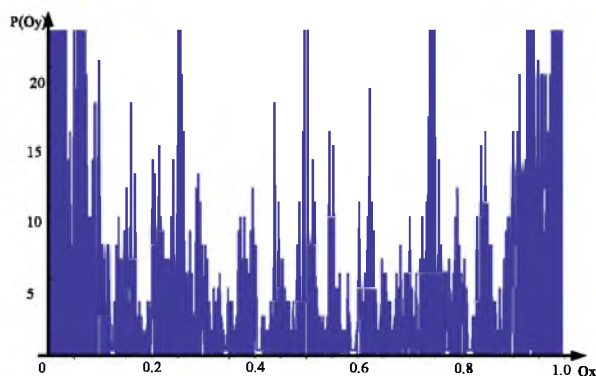


Рис. 10. Проекция строки до применения алгоритма фильтрации, для зашумленного номера

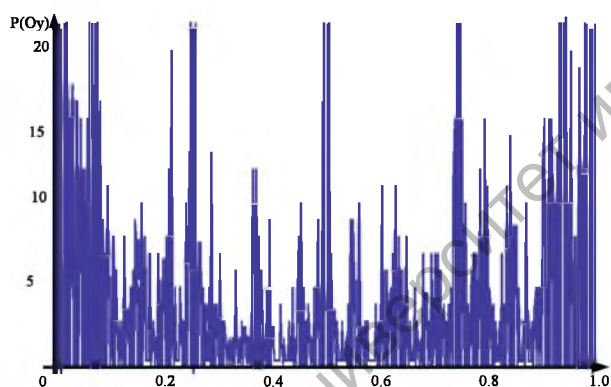


Рис. 11. Проекция строки после применения алгоритма фильтрации для зашумленного номера

На рисунках 11 и 12 наглядно продемонстрирована замена зашумленного фрагмента проекции, на фрагмент проекции скелетного символа, наиболее близко удовлетворяющий соотношению (1). Такая четкая и наглядная замена проекции произошла благодаря тому, что, во-первых, изображения были достаточно высокого качества для распознавания, а во-вторых, при создании модельных проекций использовалось не зашумленное изображение (рис. 7) и, в третьих, зашумлен был один из символов, а не группа соседних. Напомним, что основной целью примера было наглядная демонстрация основной идеи, описанной выше, применения алгоритма на изображении.

Предложенный алгоритм не отменяет последующих этапов распознавания. Напомним вид условия (1) и то, что он носит всего лишь приблизительную оценку, ограниченную имеющимся в наличии множеством модельных проекций. Представленный алгоритм не может являться последним этапом в цепочке распознавания символьной строки.

### Заключение

Распознавание автомобильных номерных знаков является одной из задач распознавания изображений, а точнее, распознавания символьных строк. Процесс распознавания символьных строк как растровых изображений включает в себя множество алгоритмов, среди них процесс сегментации является одним из фундаментальных, влияющим на дальнейший процесс. Как правило, сегментация проводится на основе построения проекции символьной строки. В работе процедура построения проекции дополнена процессом шумофильтрации.

Для успешной шумофильтрации введены модельные и скелетная проекции, обеспечивающие эквивалентную замену зашумленных фрагментов и хранящиеся в специализированных базах. Алгоритм шумофильтрации реализован на программном языке Java и встроены в библиотеку Java ANPR. В завершении работы эффективность данного алгоритма успешно апробировано на конкретных примерах. Описанный алгоритм является универсальным и может применяться для различного типа символьных строк. Шаги, описанные для построения скелетной и модельной проекций, так же могут быть расширены на большее число алфавитов. Представленный алгоритм описывает фильтрацию проекции при очистке символьных строк от помех различного характера. Он послужит отправной точкой для дальнейших исследований.

#### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. **Gonzales, R. C.** Digital image processing / R. C. Gonzales, R. E. Woods. – 2-nd edition. – Prentice-Hall, 2002. – 793 p.
2. **Фисенко В. Т.** Компьютерная обработка и распознавание изображений / В. Т. Фисенко, Т. Ю. Фисенко. – СПб. : ИТМО, 2008. – 192 с.
3. **Sawaki, M.** Text-Line Extraction and Character Recognition of Document Headlines With Graphical Designs Using Complementary Similarity Measure / M. Sawaki, N. Hagita // IEEE Trans. PAMI, 1998, vol. 20, No. 10. – Pp. 1103–1109.
4. **Визильтер, Ю. В.** Обработка изображений и распознавание образов : методическое пособие / Ю. В. Визильтер. – М. : МИРЭА, 2010. – 464 с.
5. **Jin L. W. and Qin J. Z.** Car plate number characters recognition using Gabor orientation features and neural networks // IEEE Int. Conf. Neural Networks & Signal Processing, 2003. – Pp. 1628–1631.
6. Partha Sarathi Giri. Text information extraction and analysis from images using digital image processing techniques // Special Issue of International Journal on Advanced Computer Theory and Engineering (IJASTE), Volume 2, 2013. – Pp. 66–71.
7. Yungang Zhang and Changshui Zhang. A new algorithm for character segmentation of license plate // IEEE Intelligent Vehicles Symposium, 2003. – Pp. 106–109.
8. Ondrej Martinsky. Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems // B. Sc. thesis, Brno University of Technology, Faculty of Information Technology, Department of Intelligent Systems, 2007.
9. **Заерко, Д. В.** Применение модифицированных алгоритмов Java ANPR для автоматического распознавания номеров автомобилей / Д. В. Заерко, В. А. Липницкий // Информационные технологии и системы 2018 (ИТС–2018): материалы международной научной конференции, БГУИР, Минск, Беларусь, 25 октября 2018 г. – С. 286–287.
10. **Заерко, Д. В.** Фильтрация сегментной проекции символьной строки на этапе сегментации / Д. В. Заерко, В. А. Липницкий // Технические средства защиты информации : тезисы докладов XVII Белорусско-российской научно-технической конференции, Минск, 11 июня 2019 г. / Белорусский государственный университет информатики и радиоэлектроники : редкол.: Т. В. Борботько [и др.]. – Минск, 2019. – С. 29–30.
11. **Заерко, Д. В.** Алгоритм коррекции проекции однострочного номера / Д. В. Заерко, В. А. Липницкий // Информационные технологии и системы 2019 (ИТС 2019): материалы международной научной конференции, БГУИР, Минск, Беларусь, 25 октября 2019 г. = Information Technologies and Systems 2019 (ITS 2019) : Proceeding of The International Conference, BSUIR, Minsk, 25<sup>th</sup> October 2019 / редкол.: Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2019. – С. 210–211.
12. **Заерко, Д. В.** Вычисление среднего расстояния между пиками горизонтальной проекции / Д. В. Заерко, В. А. Липницкий // Системный анализ и прикладная информатика. – 2019. – № 4. – С. 4–8.

Поступила в редакцию 13.01.2020 г.

Контакты: zaerko1991@gmail.com (Заерко Денис Владимирович)

**Zaerko D., Lipnitsky V. NOISE FILTRATION ALGORITHM FOR VERTICAL PROJECTION OF SYMBOL STRING**

*Symbol string recognition is an important part of image recognition. The use of vertical projection is common for segmentation of distorted symbol string as a bitmap image. In the article the authors suggest supplementing this procedure with a noise filtration process. The article contains a detailed description of a special projection filter used as a main tool for noise filtration. The projection is analysed together with model and skeletal projections which provide the equivalent substitution of noisy fragments and are stored in specialized bases. The process of creating bases and enriching them with new data is presented. The algorithm of noise filtration is implemented in the software language Java and is built into the library Java ANPR. A model example of filtering string noise based on the use of this library is presented.*

**Keywords:** bitmap images, vertical and horizontal projections, projection base, alphabetical projection base, average width of symbol projection, projection border, noise filtration of symbol string, estimation of permissible distortion.