

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа экономики и управления
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА
Рецензент, _____
/ _____ /
« _____ » _____ 20 _____ г.

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой, д.т.н., с.н.с.
/ Б.М. Суховилов /
« _____ » _____ 20 _____ г.

Система контроля доступа автотранспорта
на территорию предприятия

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.03.2017. 12-1220-2614. ВКР

Консультант, _____
/ _____ /
« _____ » _____ 20 _____ г.

Руководитель, Сартасов ЕМ
/ _____ /
« _____ » _____ 20 _____ г.

Автор
студент группы зЭиУ – 505
/ Агафонов Д.В./
« _____ » _____ 20 _____ г.

Нормоконтролер, _____
/ _____ /
« _____ » _____ 20 _____ г.

АННОТАЦИЯ

Агафонов, Д.В. Система контроля доступа автотранспорта на территорию предприятия
– Челябинск: ЮУрГУ,
ВШЭУ-505, 50 с., 33 ил., библиогр. список – 8 наим.

Выпускная квалификационная работа выполнена с целью разработки системы контроля доступа автотранспорта на территорию предприятия.

В работе обоснована актуальность выбранной темы, сформулированы цели и задачи.

Система контроля доступа автомобиля разработана, отлажена и протестирована.

Оглавление

1. ОБЗОР СИСТЕМ РАСПОЗНАВАНИЯ НОМЕРОВ АВТОТРАНСПОРТА .	3
2. КОМПОНЕНТЫ СИСТЕМ РАСПОЗНАВАНИЯ НОМЕРА	5
3. АЛГОРИТМ Распознавание Номера	6
4. Морфологические преобразования	10
4.1 Общая информация	10
4.2 Роль примитивов	10
4.3 Основные операции	12
4.4 Перенос	14
4.5 Нарращивание, эрозия, замыкание, размыкание	14
4.6 Нарращивание	15
4.7 Эрозия	15
4.8 Размыкание	16
4.9 Замыкание	17
4.10 Условное наращивание	17
4.11 Выделение границ	19
5. Метод идентификации контуров	22
5.1 Алгоритм Фримена.....	22
6. Библиотека OPENCV	26
6.1 Обзор	26
6.2 Используемые функции	28
7. Работа приложения	31
8. Дополнительные функции.	
9. БИБЛИОГРАФИЧЕСКИЙ СПИСОК	38

1. ОБЗОР СИСТЕМ РАСПОЗНАВАНИЯ НОМЕРОВ АВТОТРАНСПОРТА

В настоящее время, в связи с активным ростом автомобилизации, появляется необходимость анализировать и контролировать поток автомобилей на определённых участках. В частности, для предприятий, чьи территории, как правило, являются закрытыми, а их деятельность так или иначе связана с постоянным автомобильным трафиком, появляется задача допуска ограниченного количества автомобилей, их учёт, учет времени пребывания на территории, разрешение ночной стоянки, часы въезда и выезда, управление автоматическим шлагбаумом, а также и другие задачи. Все поставленные задачи опираются на результат выполнения одной единственной операции – распознавании автомобильного номера проезжающего транспортного средства.

Системы распознавания номеров достаточно широко представлены на рынке. На государственном уровне они присутствуют в качестве карательных систем, таких как скоростной контроль, регистрация проезда в моменты, когда он запрещён («красный свет», пешеходный переход с пешеходами на нём и другие).

В качестве примера можно упомянуть систему выявления нарушений «Автодория», которая может измерять скорость проезда автомобиля (на пункте контроля – мгновенную, или с использованием 2 пунктов контроля - среднюю), Так же с помощью нескольких комплексов можно получать информацию о загруженности дорог и с помощью информационного табло информировать водителей.

Мобильный комплекс «Стрелка-М» позволяет быстро развернуть пункт контроля нарушений ПДД в любом месте.

Автоматические платные парковки взимают плату за стоянку, регистрируя время въезда и выезда. Подавляющее большинство систем в той или иной мере нацелено на решение задач по оборудованию ограничения допуска на ограниченную территорию. Исполнительными механизмами у таких системы могут быть шлагбаумы и заградительные барьеры, информационные табло (например, о количестве свободных мест на парковке). Среди них можно выделить систему **sPARK 2.3 Professional**, в которой есть инструменты для противодействия угону транспортного средства.

Среди наиболее интересных систем для предприятий можно упомянуть «**FortNet**». Особенностью является ограничение времени въезда-выезда, организация «гостевого» въезда, привязка документов на отгрузку к автомобилю, чтобы избежать ошибок склада отгрузки и препятствовать кражам.

2. КОМПОНЕНТЫ СИСТЕМ РАСПОЗНАВАНИЯ НОМЕРА

Среди компонентов таких систем обязательно присутствуют следующие элементы:

- Камера видеонаблюдения;
- Вычислительная система.

Выбор этих компонентов всецело определяет качество распознавания, причем камере принадлежит главная роль. Главными элементами камеры являются размер матрицы и светосила объектива. В общем случае здесь следует правило «чем больше, тем лучше». Немного иначе с разрешением матрицы. Слишком низкое разрешение приведёт к «квадратичности» изображения и невозможности распознавать номер с высокой степенью вероятности, в то же время, слишком высокое разрешение приведёт к очень высокой нагрузке на вычислительную систему, поэтому разрешение должно обеспечить близкий к оптимальному размер номерной планки.

Вычислительная система должна обладать необходимым быстродействием. Слишком низкое быстродействие приведёт к невозможности распознать часто проезжающие автомобили, слишком высокое – к излишнему удорожанию.

Номерные планки стандартизированы [1], размеры, толщина и цвет линий, расположение символов, цвета фона имеют чёткие требования, что повышает качество распознавания номера

3. АЛГОРИТМ РАСПОЗНАВАНИЕ НОМЕРА

В целом задача распознавания номера решается в сложных условиях эксплуатации: при различных погодных условиях, освещении, загрязнённости номерных знаков.

Процесс распознавания номерного знака состоит из нескольких этапов:

1. выделение номерной пластины в кадре;
2. наложение сетки на выделенную область;
3. распознавание текста.

На первом этапе работы алгоритма происходит выделение номерной пластины в кадре. Для этого используется особенность номерного знака, которая заключается в резких скачках яркости от белого к черному. Строится график зависимости яркости от координаты изображения – X. Номерная пластина будет находиться в области, где на графике наблюдаются резкие выбросы.

Для построения графика яркости производится построчное сканирование изображения [2,3]. По полученным результатам строится функция зависимости яркости изображения от координаты (1).

(1)

где $I(i)$ – значение яркости, вычисляемое по следующей формуле (2):

(2)

где R, G, B – это цвета красный (R), зеленый (G), синий (B), а значения 0,212671, 0,715160, 0,072169 – это коэффициенты веса для каждого цвета гаммы RGB.

Для нахождения вертикальных краев номерной пластины находится вторая производная функции (1). Исходя из данных, полученных на графике яркости, выделяется примерная область расположения пластины. Далее на выделенную область накладывается сетка – шаблон. В случае их совпадения выделенная область считается номерной пластиной.

Для перехода к этапу распознавания текста на номерной пластине необходимо применить к полученному изображению ряд фильтров, чтобы изображение было более четким.

Данный алгоритм определения положения номерной пластины довольно устойчив к помехам, единственное, что может помешать определить положение номерного знака – это полное его загрязнение или его отсутствие.

На втором этапе работы алгоритма осуществляется распознавание текста на номерном знаке. Этот этап содержит в себе три подэтапа:

- создание бинарной матрицы символов;
- векторизация бинарной матрицы;
- распознавание символа по составленному набору векторов.

Составление бинарной матрицы производится по алгоритму нахождения К-среднего. Белому пикселю ставится в соответствие единица, а черному пикселю ноль. Строится бинарная матрица (см.Рисунок 1)

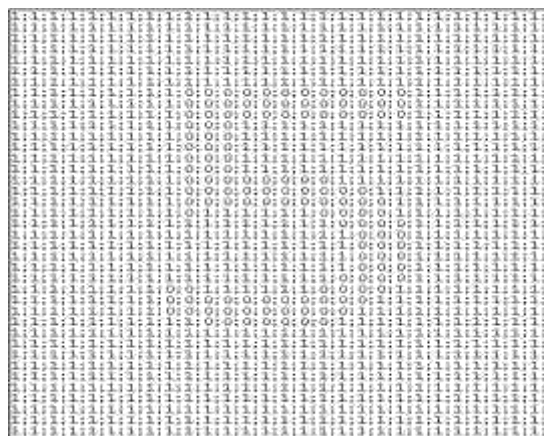


Рисунок 1 Бинарная матрица изображения цифры 5

Процесс векторизации заключается в составлении набора векторов, который будет описывать символ, представленный в бинарной матрице. Для векторизации используется восемь направлений – восемь векторов, которые пронумерованы от 0 до 7 (см.Рисунок 2)

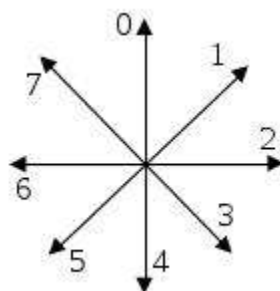


Рисунок 2 Роза ветров

Обход символа по контуру начинается с левого верхнего угла, а заканчивается когда алгоритм пройдет в ту точку, из которой начинался обход. Иногда в ходе векторизации получаются несколько подряд идущих векторов одного направления. В данном случае они объединяются в один вектор (см. Рисунок 3).

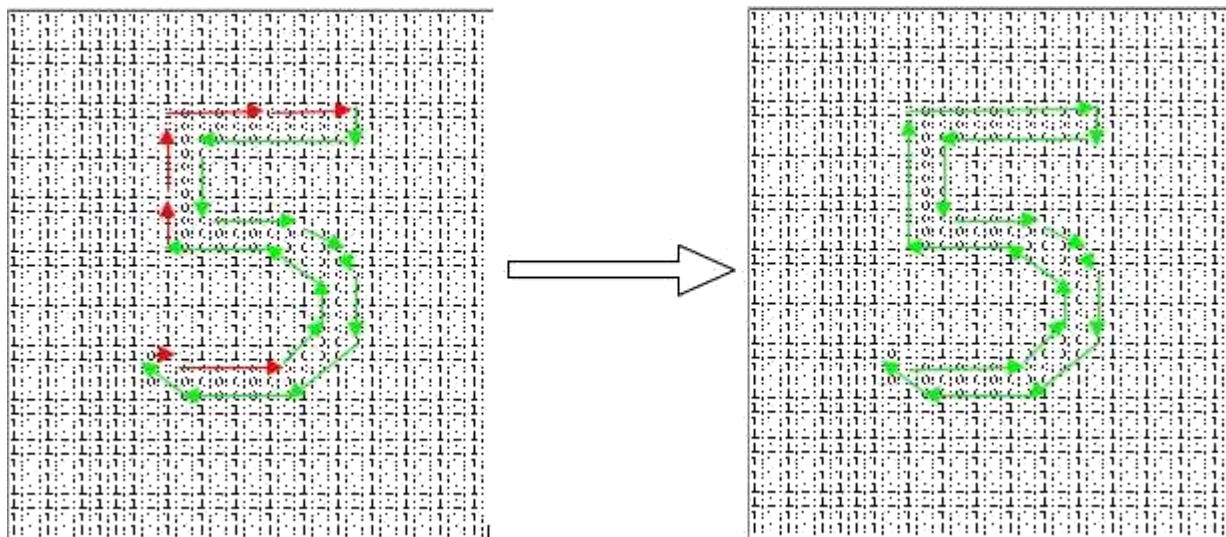


Рисунок 3 Пример склеивания векторов одного направления

В итоге векторизации получаем набор векторов, которые описывают символ бинарной матрицы.

Метод векторизации символов хорош тем, что он позволяет в дальнейшем сравнивать и распознавать символы независимо от того какого размера распознаваемые символы.

На последнем этапе работы алгоритма производится сравнение полученного набора векторов с набором эталонных векторных описаний символов. Сравнение производится с использованием расстояния Левенштейна, что позволяет с большой точностью и при небольших затратах времени и ресурсов распознать символы.

По сравнению с другими алгоритмами распознавания номерных знаков данный алгоритм отличается простотой реализации, высокой производительностью. Основным его преимуществом является то, что при его использовании не важен размер распознаваемого символа. То есть он является универсальным для камер с любой разрешающей способностью.

4. MORFOЛОГИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ

4.1 Общая информация

Термин морфология [4] относят к описанию структуры и свойств формы структуры графических объектов. В применении к машинному зрению этот термин относится к описанию свойств формы областей на изображении. Изначально операции математической морфологии рассматривались как операции над множествами, но с появлением ЭВМ выяснилось, что они оказались полезными в задачах обработки множества точек в двумерном пространстве, которые представляют из себя, в частности, оцифрованные изображения. В математической морфологии под множеством понимаются объекты на изображении. Легко заметить, что одним из вариантов полного описания бинарного изображения является множество всех фоновых пикселей этого изображения. В первую очередь математическую морфологию необходимо использовать при извлечении некоторых свойств изображения, нужных для его описания и представления. Например, выпуклых оболочек, контуров, остовов. Также очень полезными окажутся некоторые морфологические методы, используемые на разных стадиях предварительной и итоговой обработки изображений. К примеру, морфологическая фильтрация, утолщение или утоньшение.

4.2 Роль примитивов

Исходными данными для механизма математической морфологии являются два изображения: обрабатываемое и специальное, зависящее от вида операции и решаемой задачи [5]. Такое специальное изображения принято называть примитивом или структурным элементом. Структурный элемент, как правило, намного меньше обрабатываемого изображения, что позволяет его считать описанием области с некоторой формой. Форма может быть любой, главное, чтобы её можно было представить в виде бинарного изображения заданного размера. Во многих пакетах обработки изображений наиболее распространенные структурные элементы имеют специальные названия: $BOX[H,W]$ –прямоугольник

заданного размера, $DISK[R]$ — диск заданного размера, $RING[R]$ — кольцо заданного размера.

Примеры структурных объектов приведены на рисунках ниже:

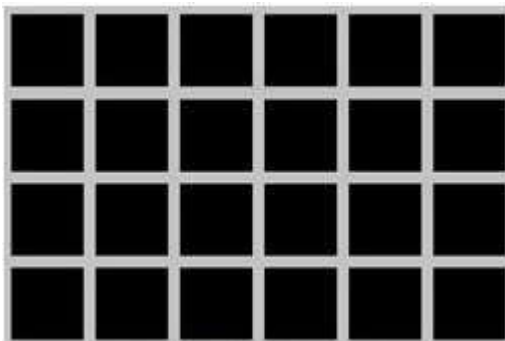


Рисунок 4 $BOX[4,6]$

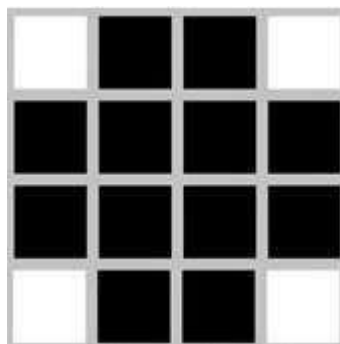


Рисунок 5 $DISK[4]$

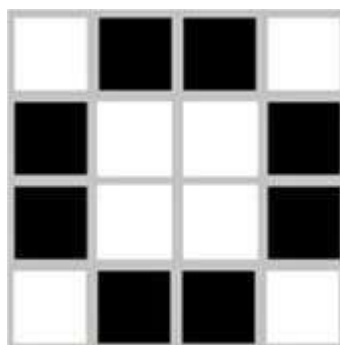


Рисунок 6 $RING[4]$

Результат операций морфологической обработки зависит от размера и конфигурации исходного изображения, так и от структурного примитива. Размер структурного элемента, как правило, равен 5x5, 4x4 или 3x3 пикселей. Это определяется главной идеей морфологической обработки, в процессе которой отыскиваются характерные детали изображения. Искомый объект описывается

примитивом, и в результате морфологической обработки можно подчеркнуть или удалить такие детали на всём изображении. Среди основных преимуществ морфологической обработки – её относительная простота: входные данные и результат процедуры обработки представляет из себя бинаризованное изображение. Ряд методов, как правило, из исходного изображения сначала получают полутоновое, которое затем приводится к бинарному с помощью пороговой функции.

4.3 Основные операции

Основными операциями математической морфологии являются наращивание, эрозия, замыкание и размыкание. По названиям можно определить назначение операций: наращивание увеличивает область изображения, эрозия делает её меньше, операция замыкания замыкает внутренние дыры области и устраняет заливы вдоль границы, операция размыкания помогает избавиться от маленьких, выступающих наружу, фрагментов.

Рассмотрим теоретико-множественные операции, лежащие в основе математической морфологии. Объединение двух множеств A и B , которое обозначают $C=A \cup B$, есть множество всех элементов, принадлежащих хотя бы одному множеству A или B . Аналогично, пересечение двух множеств A и B , которое обозначается $C=A \cap B$, есть множество всех элементов, принадлежащих одновременно множествам A и B . Дополнение множества A есть множество элементов, не содержащихся в A : $A^c = \{w | w \notin A\}$. Разность двух множеств A и B обозначается $A \setminus B$ и определяется так: $A \setminus B = \{w | w \in A, w \notin B\} = A \cap B^c$. Это множество состоит из элементов A , которые не входят в множество B . Рассмотрим все вышеперечисленные операции на конкретном примере.

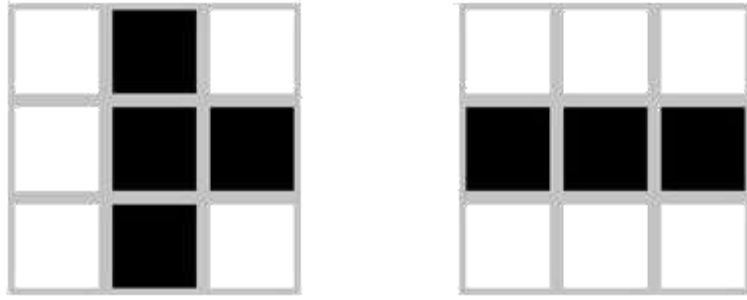


Рисунок 7 Множества А и В

U

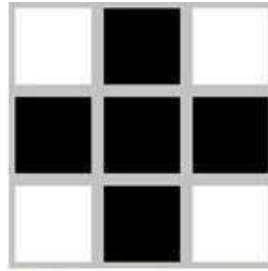


Рисунок 8 Операция объединения

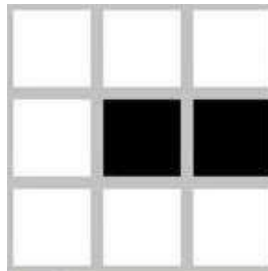


Рисунок 9 Операция пересечения

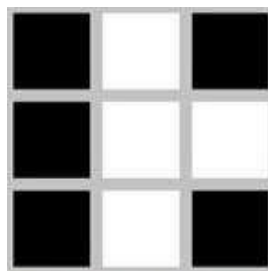


Рисунок 10 Операция дополнения

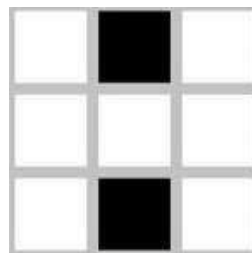


Рисунок 11 Операция разности

4.4 Перенос

Операция переноса X_t множества пикселей X на вектор t задаётся в виде $X_t = \{x+t | x \in X\}$. Следовательно, перенос множества единичных пикселей на бинарном изображении сдвигает все пиксели множества на заданное расстояние. Вектор переноса t может задаваться в виде упорядоченной пары $(\Delta r, \Delta c)$, где Δr – компонент вектора переноса в направлении строк, а Δc — компонент вектора переноса в направлении столбцов изображения.

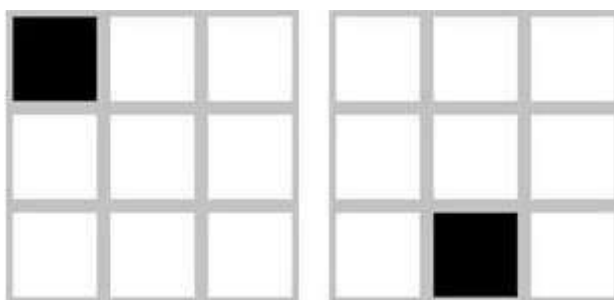


Рисунок 12 Пример переноса при $t=[2,1]$

4.5 Нарастивание, эрозия, замыкание, размыкание

Следующие операции рассмотрены на конкретном примере. Пусть есть следующее бинарное изображение и структурный элемент:

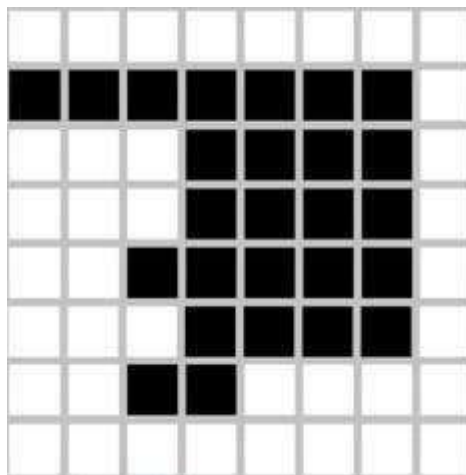


Рисунок 13 Бинарное изображение В

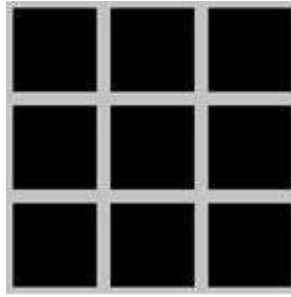


Рисунок 14 Структурный элемент S

4.6 Нарращивание

Структурный элемент S применяется ко всем пикселям бинарного изображения. Каждый раз, когда начало координат структурного элемента совмещается с единичным бинарным пикселем, ко всему структурному элементу применяется перенос и последующее логическое сложение с соответствующими пикселями бинарного изображения. Результаты логического сложения записываются в выходное бинарное изображение, которое изначально инициализируется нулевыми значениями.

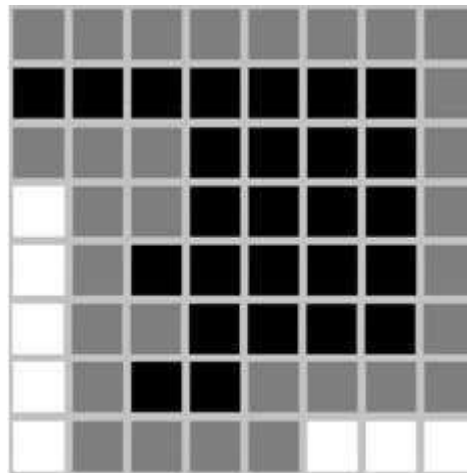


Рисунок 15 Нарращивание изображения В структурным элементом S

4.7 Эрозия

При выполнении операции эрозии структурный элемент тоже проходит по всем пикселям изображения. Если в некоторой позиции каждый единичный пиксель структурного элемента совпадет с единичным пикселем бинарного изображения, то выполняется логическое сложение центрального пикселя структурного элемента с соответствующим пикселем выходного изображения.

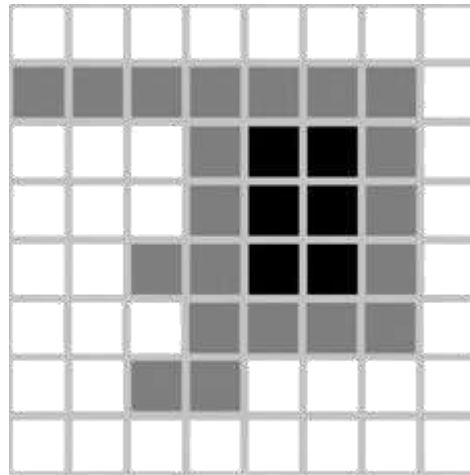


Рисунок 16 Эрозия изображения В структурным элементом S

В результате применения операции эрозии все объекты, меньшие, чем структурный элемент, стираются, объекты, соединённые тонкими линиями становятся разъединёнными и размеры всех объектов уменьшаются.

4.8 Размыкание

Операция эрозии полезна для удаления малых объектов и различных шумов, но у этой операции есть недостаток – все остающиеся объекты уменьшаются в размере. Этого эффекта можно избежать, если после операции эрозии применить операцию наращивания с тем же структурным элементом.

Размыкание отсеивает все объекты, меньшие, чем структурный элемент, но при этом помогает избежать сильного уменьшения размера объектов. Также размыкание идеально подходит для удаления линий, толщина которых меньше, чем диаметр структурного элемента. Также важно помнить, что после этой операции контуры объектов становятся более гладкими.

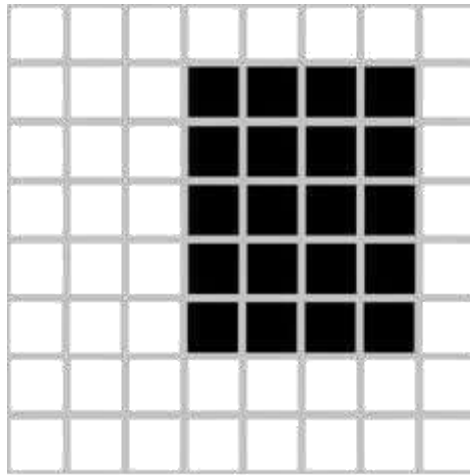


Рисунок 17 Размыкание изображения В структурным элементом S

4.9 Замыкание

Если к изображению применить сначала операцию наращивания, то мы сможем избавиться от малых дыр и щелей, но при этом произойдет увеличение контура объекта. Избежать этого увеличения позволяет операция эрозия, выполненная сразу после наращивания с тем же структурным элементом.

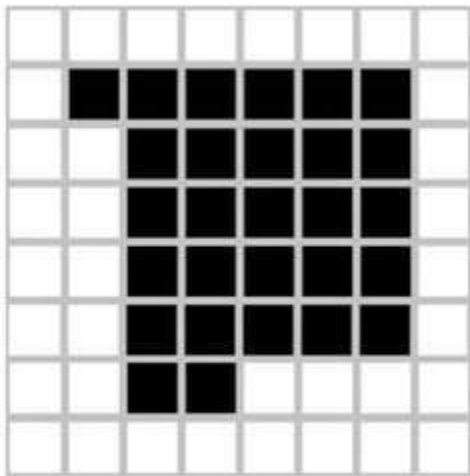


Рисунок 18 Замыкание изображения В структурным элементом S

4.10 Условное наращивание

Одним из типичных применений бинарной морфологии является выделение на бинарном изображении компонент, у которых форма и размеры удовлетворяют заданным ограничениям. Во многих подобных задачах возможно построение структурного элемента, который после применения к бинарному изображению удаляет не удовлетворяющие ограничениям компоненты и оставляет несколько

единичных пикселей, соответствующих удовлетворяющим ограничениям компонентам. Но для последующей обработки могут потребоваться компоненты целиком, а не только их фрагменты, оставшиеся после эрозии. Для решения этой проблемы была введена операция условного наращивания. Множество полученное в результате эрозии циклически наращивается структурным элементом S , и на каждом шаге результат уменьшается до подмножества пикселей, которые имеют единичные значения на исходном изображении B . Операция условного наращивания пояснена на рисунке ниже. На этом рисунке бинарное изображение B было подвергнуто эрозии элементом V для выделения компонент, содержащих вертикальные фрагменты высотой 3 пиксела. На полученном изображении C есть две таких компоненты. Чтобы выделить эти компоненты целиком, изображение C условно наращивается элементом D относительно исходного изображения B .



Рисунок 19 Структурный элемент V

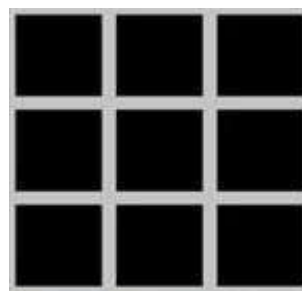


Рисунок 20 Структурный элемент D

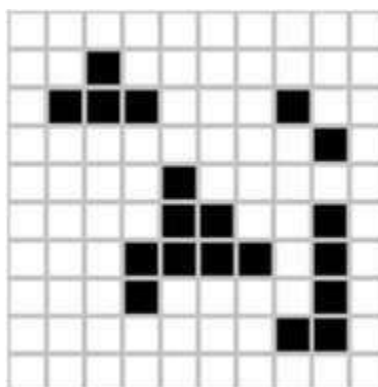


Рисунок 21 Бинарное изображение B

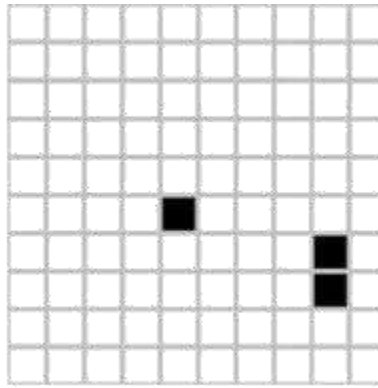


Рисунок 22 Эрозия изображения В структурным элементом V

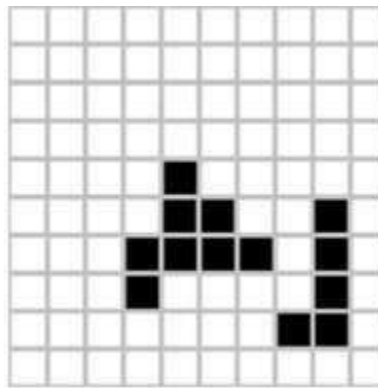


Рисунок 23 Условное наращивание изображение структурным элементом D

4.11 Выделение границ

Морфологические операции можно также использовать для выделения границ бинарного объекта. Это операция очень важна, потому что граница является полным, и в то же время весьма компактным описанием объекта. Легко заметить, что граничные точки имеют как минимум один фоновый пиксел в своей окрестности. Таким образом, применив оператор эрозии с структурным элементом, содержащим все возможные соседние элементы, мы удалим все граничные точки. Тогда граница получится с помощью операции разности множеств между исходным изображением и изображением, полученным в результате эрозии.

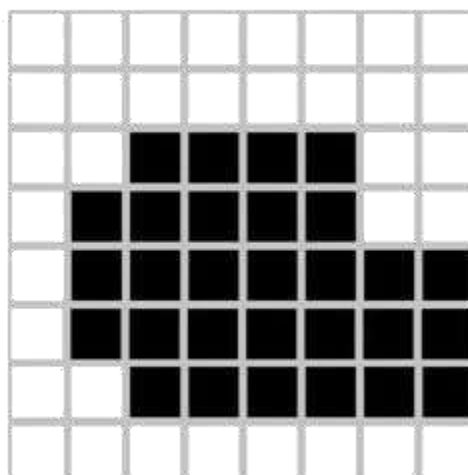


Рисунок 24 Исходное изображение

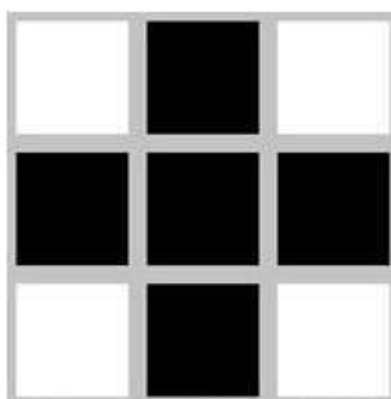


Рисунок 25 Структурный элемент для 4-компонентной окрестности

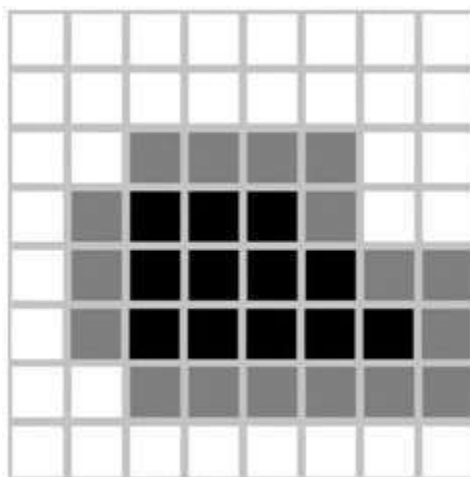


Рисунок 26 Изображение после эрозии

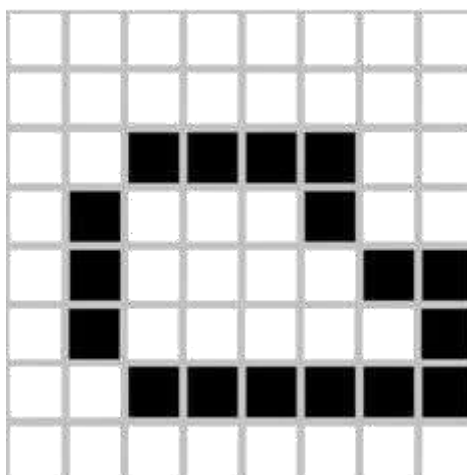


Рисунок 27 Границы объекта

Таким образом, рассмотрены основные операции математической морфологии, и несколько способов их применения.

5. МЕТОД ИДЕНТИФИКАЦИИ КОНТУРОВ

Контурное представление объектов представляет собой набор данных, дающих представление об форме объекта, его выпуклостях, пустотах. Такое представление объектов занимает меньше памяти, его проще обрабатывать. Частые операции сравнения объектов так же требуют меньше ресурсов.

5.1 Алгоритм Фримена

При выполнении многих операций над изображением интерес представляют точки цифрового изображения (x, y) , соседние по отношению к заданной

где

- матрица элементов изображения;

- количество бит (значение) каждого пикселя.

,

Множество точек, соседних по отношению к заданной (x, y) , называется *окрестностью* точки (x, y) и обозначается NS . Например, 4-х точечная окрестность определяется выражением:

Окрестность может содержать точку (x, y) , может не содержать её.

Операцией над изображением называется процесс преобразования изображения в изображение $I' = T(I)$.

Операция над изображением называется локальной, если отображение зависит от значения $I(x, y)$ в точке (x, y) .

Операция усреднения является локальной и применяется к изображению с целью сглаживания помех. Для этого берётся среднее арифметическое пикселей, принадлежащих к локальной окрестности. Например, операцию усреднения по пяти точкам с включением точки (x, y) можно представить таким образом:

Значения пикселей на границе изображения вычислены по 3-м или 4-м соседним точкам с добавлением нулей: одного или двух, чтобы довести количество точек до 5-ти. Чтобы уменьшить размывание изображения, используют метод селективного сглаживания. При этом значение конкретного пикселя заменяется средним по его соседям, если оно отличается от этого среднего по меньшей мере на заданное число градаций яркости R . По существу, этот метод пытается расклассифицировать пиксели на зашумлённые и не зашумлённые, сглаживаются при этом только зашумлённые пиксели.

Операция порогового (бинарного) преобразования является локальной и трансформирует любое изображение в двухградационное. Величина порога t задаётся. Например,

тогда

Все рассмотренные изображения представлены в цифровом виде, полученном в результате дискретизации, квантования и кодирования первоначального непрерывного изображения. Разнообразие кодов для представления

изображения практически бесконечно. Рассмотрим инкрементный код. Этот код определяется исходной точкой и последовательностью элементарных перемещений (в частности, код Фримена). Подобное кодирование называют *цепным*. Суть цепного кодирования состоит в прослеживании линий или точек границы и в последовательном их кодировании по принципу:

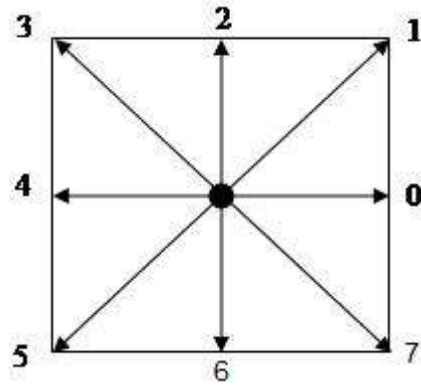


Рисунок 28 Схема возможных направлений при цепном кодировании изображений

Наклон может иметь только восемь направлений. Цепное кодирование предусматривает использование следующих понятий:

звено - , то есть направленный сегмент, ,

цепь - , то есть упорядоченная последовательность звеньев с включениями в неё кодов сигналов, например, таких, как

0400 - конец цепи:

0407__ - порядковый номер __ в цепи;

0424 - градация яркости;

0426 , 0427 - координаты начальной точки соответственно x и y .

Например:

Начальная точка имеет координаты (1, 3). Необходимо по изображению определить цепной код замкнутой кривой.

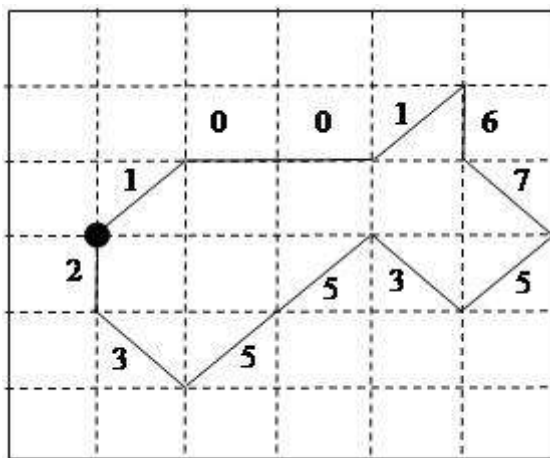


Рисунок 29 Пример изображения с заданием значений элементов кода

Для этой замкнутой кривой цепной код имеет вид:

040701 0424001 0426001 0427003 **100167535532** 0400.

где жирным шрифтом выделена часть кода, характеризующая только заданную на рисунке 29 кривую линию.

Обратная цепь геометрически конгруэнтна заданной цепи и направлена в другую сторону. При задании обратного кода в противоположную сторону меняется лишь его часть, характеризующая заданное изображение.

Например, для изображения на рисунке 29

040701 0424001 0426001 0427003 671171325445 0400

6. БИБЛИОТЕКА OPENCV

6.1 Обзор

OpenCV – (Open Computer Vision) — свободная библиотека компьютерного зрения с открытым исходным кодом, предоставляющая набор специальных типов данных и численных алгоритмов для обработки изображений алгоритмами компьютерного зрения. Реализована на C/C++.

Это библиотека изначально была разработана Центре разработки программного обеспечения Intel (российской командой). OpenCV написана на языке высокого уровня (C/C++) и содержит алгоритмы для: интерпретации изображений, устранение оптических искажений, калибровки камеры по эталону, анализ перемещения объекта, определение сходства, определение формы объекта и слежение за объектом, сегментация объекта, распознавание жестов, 3D-реконструкция и т.д.

Эта библиотека получила широкое распространение за счёт своей открытости и бесплатности как для учебных, так и для коммерческих целей. **Фактически, OpenCV состоит из наборов типов данных, функций и классов для обработки изображений.**

Основные модули библиотеки:

core — ядро, содержит базовые структуры данных и алгоритмы:

- базовые операции над многомерными числовыми массивами
- матричная алгебра, математические функции, генераторы случайных чисел
- Запись/восстановление структур данных в/из XML
- базовые функции 2D графики

CV - модуль обработки изображений и компьютерного зрения

- базовые операции над изображениями (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.)
- анализ изображений (выбор отличительных признаков, морфология, поиск контуров, гистограммы)
- анализ движения, слежение за объектами
- обнаружение объектов, в частности лиц
- калибровка камер, элементы восстановления пространственной структуры

Highgui

- модуль для ввода/вывода изображений и видео, создания пользовательского интерфейса
- захват видео с камер и из видео файлов, чтение/запись статических изображений.
- функции для организации простого UI (все демо приложения используют HighGUI)

SvauX - экспериментальные и устаревшие функции— пространств. зрение: стерео калибровки, самокалибровки

- поиск стерео-соответствия, клики в графах
- нахождение и описание черт лица CvCam
- захват видео
- позволяет осуществлять захват видео с цифровых видео-камер (поддержка прекращена и в последних версиях этот модуль отсутствует)

Часть функций оказалась очень полезна для задачи поиска и определения автомобильного номера.

6.2 Используемые функции

На начальном этапе изображение преобразуется в черно-белое (8бит, 1 цветовой канал) с помощью функции `cvCvtColor`. Эта функция выполняет операцию свёртки, то есть операцию, показывающую «схожесть» одной функции с отражённой и сдвинутой копией другой. Когда в качестве исходных данных функции принимается изображение, то свёртка — это операция вычисления нового значения заданного пикселя, при которой учитываются значения окружающих его соседних пикселей. Главный элемент свёртки - ядро свёртки — это матрица (произвольного размера и отношения сторон; чаще всего используется квадратная матрица (по умолчанию, размеры 3x3)).

Ядро свертки содержит важный параметр – якорь. Часто он расположен в центре ядра. При вычислении нового значения выбранного пикселя изображения, ядро свёртки прикладывается своим центром к этому пикселю. Соседние пиксели так же накрываются ядром. Вычисляется сумма произведений значений пикселей изображения на значения, накрывшего данный пиксель элемента ядра. Полученная сумма и является новым значением выбранного пикселя. При применении свёртки к каждому пикселю изображения все пиксели зависят от выбранного ядра свертки, и, таким образом, может быть обработано всё изображение. Операция «свёртка» выполняется функцией `cvFilter2D`.

`cvSmooth` – функция “сглаживания” изображения. Основная задача сглаживания — использование возможностей дискретного устройства вывода для показа оттенков цвета, которым нарисована кривая. Суть процесса в том, что пиксели, соседние с граничным пикселем изображения, принимают промежуточное значение между цветом изображения и цветом фона, создавая градиент и размывая границу. Таким образом, достигается устранение ступенек на прямых, не являющихся горизонтальными или вертикальными.

`cvErode` - Эрозия (размывание/сужение) изображения обычно используется для избавления от случайных вкраплений на изображении. Идея состоит в том, что вкрапления при размывании удаляются, тогда как крупные и, соответственно, более крупные детали остаются.

`cvDilate` – Расширение изображения, то есть укрупнение существующих деталей. При этом достигается уменьшение цифровых шумов (эффект снега).

`cvCanny` – процедура поиска краёв. Края(границы) — это такие кривые на изображении, вдоль которых происходит резкая смена яркости или других видов неоднородностей. Края указывают на важные особенности изображения и поэтому, целями преобразования изображения в набор кривых являются выделение существенных характеристик изображения и сокращение объема информации для последующего анализа. Самым популярным методом выделения границ является детектор границ Кенни.

`cvFindContours` – поиск контуров. Контурный анализ — один из важных и очень полезных методов описания, обработки и поиска графических объектов. Контур — это внешние очертания объекта. При проведении контурного анализа внутренние точки объекта не рассматриваются, имеет значение только границы объекта. Так же есть существенные ограничения на применимость этого метода: \при одинаковой яркости с фоном объект может не иметь чёткой границы, или может быть зашумлён помехами, что приводит к невозможности выделения контура и перекрывающиеся объекты или их группы приводит к тому, что контур определяется неверно. Но переход к рассмотрению только контуров объектов позволяет выйти из пространства изображения к пространству контуров, что существенно снижает сложность алгоритмов и вычислений. Таким образом, контурный анализ имеет довольно слабую устойчивость к помехам, и любое пересечение или лишь частичная видимость объекта приводит либо к невозможности детектирования, либо к ложным срабатываниям, но простота и быстрое действие контурного анализа, позволяют вполне успешно применять

данный подход (при чётко выраженном объекте на контрастном фоне и отсутствии помех).

`cvApproxPoly` – аппроксимация кривых, упрощает представление контуров.

`cvContourPerimeter` – вычисляет периметр заданного контура, позволяя оценить размеры объекта вместе с результатом следующей функции

`cvContourArea` – вычисляет площадь заданного контура.

`cvSetImageROI` – задаёт область обработки изображения. Все действия над изображением ограничиваются заданной областью.

`cvCreateImage` – резервирует в памяти объём, необходимый для хранения изображения указанного размера с указанной разрядностью цвета и количество каналов.

7. РАБОТА ПРИЛОЖЕНИЯ

Приложение выполнено в среде visual studio 2013 на языке visual c++.

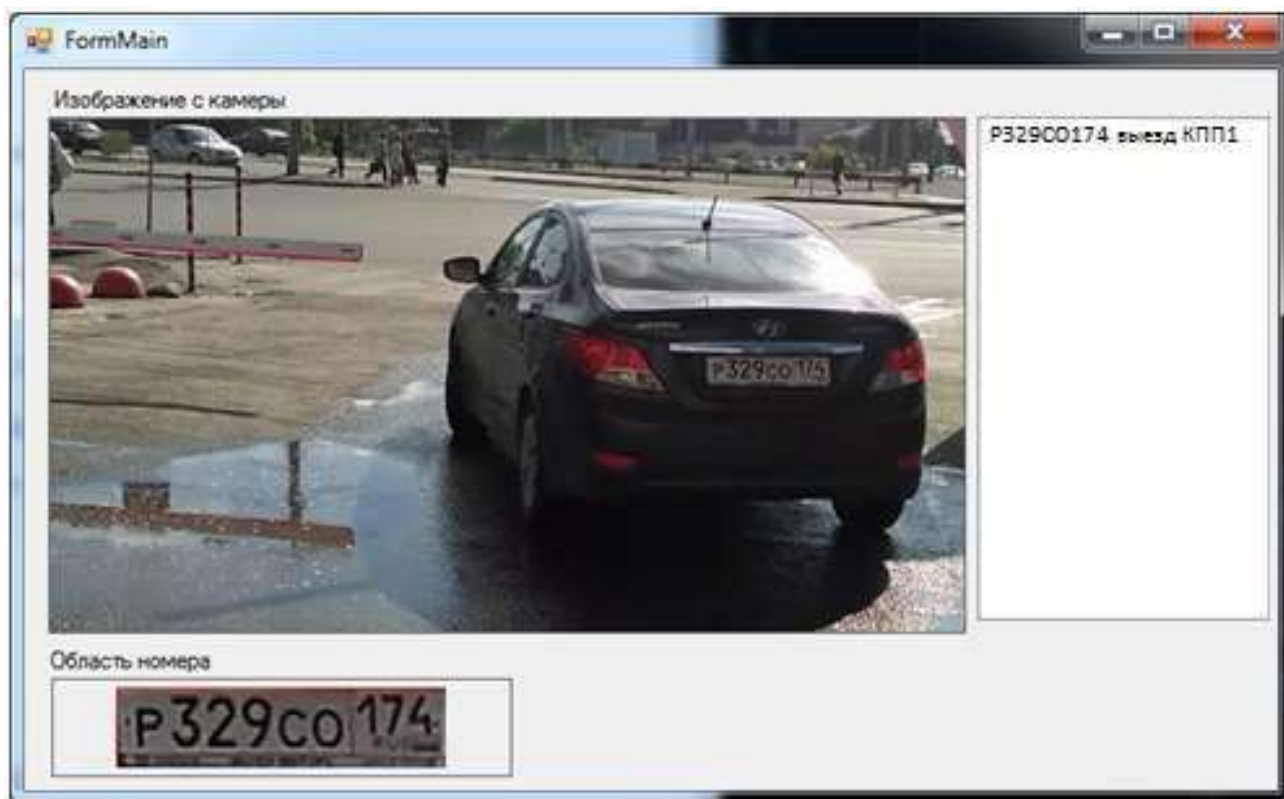


Рисунок 30 Основное окно приложения

На начальном этапе программа инициализирует источник видеосигнала (IP-камера, видеофайл и т.д) и начинает извлекать отдельные кадры для анализа графической информации.

```
CvCapture* capture = cvCreateFileCapture(filename);
```

Цикл извлечения кадров – это, в идеальном случае, бесконечный цикл

```
while (1){  
    char PressedKey = cvWaitKey(10);  
    if (PressedKey == 27) { // вызод по ESCape  
        break;  
    }  
}
```


Однако, в реальной жизни, потребуется закрывать программу по различным причинам. Поэтому добавлена возможность прекращения цикла по нажатию на клавишу ESCape.

Каждый кадр должен быть выведен на форму

```
ImageToForm = frame;
    pictureBoxCurrentFrame->Image = gcnw Bitmap(ImageToForm->width, ImageToForm-
>height, ImageToForm->widthStep,
    PixelFormat::Format24bppRgb, IntPtr(ImageToForm->imageData));
```

Подготовка к распознаванию производится с помощью следующих операций:

```
cvCvtColor(_image, temp, CV_BGR2GRAY);
```

Преобразование цветного изображения в черно-белое. Таким образом достигается уменьшение объема (в байтах) изображения, что ускоряет и упрощает анализ.

```
float kernel[9];
kernel[0] = -0.1;
kernel[1] = 0.2;
kernel[2] = -0.1;

kernel[3] = 0.2;
kernel[4] = 3;
kernel[5] = 0.2;

kernel[6] = -0.1;
kernel[7] = 0.2;
kernel[8] = -0.1;

CvMat kernel_matrix = cvMat(3, 3, CV_32FC1, kernel);
cvFilter2D(temp, temp, &kernel_matrix, cvPoint(-1, -1));
```

Это операция увеличения яркости. Номерная пластина состоит из чередования очень тёмных и очень светлых участков. При съемке в неблагоприятных условиях, например, слабая подсветка, съемка «против солнца», фон номера может оказаться тёмным.

```
cvSmooth(temp, temp, CV_GAUSSIAN, 3, 3, 0, 0);
```

Применив гауссовское сглаживание, можно избавиться от «ступенчатых» линий.

```
cvErode(temp, temp, NULL, 1);
```

Удаление самых мелких дефектов

```
cvDilate(temp, temp, NULL, 1);
```

удаление цифровых шумов

```
cvCanny(temp, temp, 100, 50, 3);
```

нахождение контуров. Номерная планка

```
cvFindContours(temp, storage, &contour, sizeof(CvContour), CV_RETR_EXTERNAL, 3,  
cvPoint(0, 0));
```

производится поиск контуров определённого размера

```
contourLow = cvApproxPoly(contour, sizeof(CvContour), storage, CV_POLY_APPROX_DP, 5, 12);
```

все найденные контуры аппроксимируются до прямоугольников.

После производится анализ найденных контуров, происходит фильтрация по ширине и отношению сторон. После с высокой долей вероятности находится номерная пластина. Найденный фрагмент передаётся в модуль распознавания Tesseract. Достоверность распознавания подтверждается, когда один номер распознается в смежных кадрах не менее четырёх раз из пяти. Хранение последних результатов организовано в компоненте TTreeView, где корневыми узлами отражаются версии модуля распознавания.

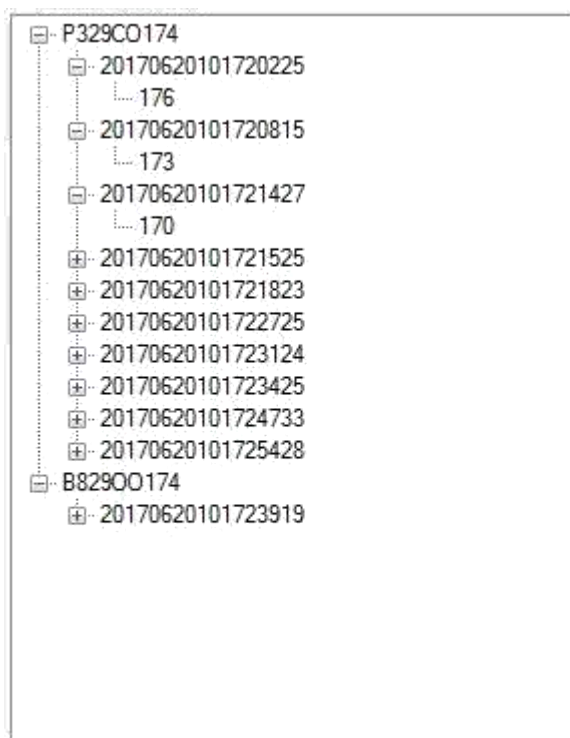


Рисунок 31 Хранилище последних результатов распознавания

В ветви каждой версии может присутствовать некоторое количество узлов с именем в формате “YYYYMMDDHHNNSSLLLL”, где

- YYYY – год
- MM – месяц
- DD – день
- HH - час
- NN - минута
- SS - секунда
- LLLL – тысячная доля секунды

Это имя кодирует момент времени, когда модуль распознавания определил версию номера, указанную в родительском узле. В каждом таком узле есть 1 дочерний узел, в котором указана ширина номерной рамки в пикселях. На основании изменений ширины рамки принимается решение о направлении движения автомобиля. В случае успешного определения направления выводится

сообщение в списке последних событий. После полученный номер заносится в базу данных и передаётся на обработку доступа на территорию.


На рисунке 32 приведён пример ситуации, когда автомобиль покидает территорию предприятия. В правой части окна отображена информация о результатах распознавания номера, а именно – моменты времени, когда оно произошло и ширина номерной рамки. Также есть одно ошибочное распознавание (внизу) – оно будет проигнорировано. Ниже выведен список событий, в котором так же выводится информация о владельце и в какой категории он находится.



Рисунок 32 Регистрация выезда автомобиля

FormMain

Изображение с камеры



H756TO174
 20170620102058145
 20170620102058848
 20170620102059520
 20170620102059860
 20170620102100120
 20170620102101836
 20170620102102374
 M156TO174
 20170620102058349
 20170620102059608
 H156TO174
 20170620102100870
 M155TO174
 20170620102101258

H756TO174 2471

H756TO174 въезд - Организация Обл. центр мед. проф

Область номера

Допуск **РАЗРЕШЕНО**

Информация о владельце

8. ФУНКЦИИ СИСТЕМЫ

Кроме основной своей задачи, программа так же позволяет:

- сохранять время и направление движения (от камеры или к камере) автомобиля с распознанным номером
- отслеживать перемещение транспортного средства по территории при наличии в зонах движения камер видеонаблюдения
- обслуживать нескольких пунктов въезда-выезда
- вести черный и белый список, в том числе с места, не находящегося на любом КПП
- выводить на экран информацию о владельце транспортного средства

Некоторые дополнительные функции требует предварительной настройки.

9. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гашников М.В., Глумов Н.И., Ильясова Н.И. Методы компьютерной обработки изображений. - М.: ФИЗМАТ ЛИТ, 2003.- 784 с.
2. Хорн Б.К.П. Зрение Роботов. - М.: МИР, 1989. - 487 с.
3. ГОСТ 50577-93. Знаки государственные регистрационные транспортных средств типы и основные размеры. Технические требования. Введен 01.01.94. - М.: Издательство стандартов. - 5 с.29
4. Л.Шапиро, Дж.Стокман. Компьютерное зрение. изд. — М.: БИНОМ. Лаборатория знаний, 2006. — 752 с.
5. Д.Форсайт, Ж.Понс. Компьютерное зрение. Современный подход. изд. — М.: Вильямс, 2004. — 928 с.
6. http://edu.sernam.ru/book_стой.php?id=26
7. Ш.-К. Чэн. Принципы проектирования систем визуальной информации. - М.: Мир, 1994. - 408
8. Ж. Эгрон. Синтез изображений. Базовые алгоритмы. - М.: Радио и связь, 1993. - 216 с.