

---

---

**ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ  
ПРОГРАММИРОВАНИЕ**

---

---

УДК 004.92+004.94

## ПРОГРАММНАЯ СИСТЕМА ОБРАБОТКИ ИЗОБРАЖЕНИЙ С ПАРАЛЛЕЛЬНЫМИ ВЫЧИСЛЕНИЯМИ

© 2020 г. К. И. Кий<sup>a,\*</sup>, Д. А. Анохин<sup>a,\*\*</sup>, А. В. Подоприсветов<sup>a,\*\*\*</sup>

<sup>a</sup> *Институт прикладной математики им. М.В. Келдыша РАН,  
Москва, Миусская пл., д. 4, 125047 Россия*

\*E-mail: kikip\_46@mail.ru

\*\*E-mail: anodmitry@gmail.com

\*\*\*E-mail: llecxis@gmail.com

Поступила в редакцию 17.12.2019 г.

После доработки 05.02.2020 г.

Принята к публикации 11.04.2020 г.

В работе описывается программная система обработки изображений с параллельными вычислениями, основанная на методе геометризованных гистограмм, разработанным для содержательного описания и сегментации цветных изображений и для создания систем понимания изображений реального времени. Параллельная обработка опирается на то, что в отличие от большинства существующих методов сегментации изображений, указанный метод построен так, что наиболее трудоемкие операции работы с пиксельным массивом могут выполняться в нем при помощи  $n$  независимых потоков. Описываются принципы построения программ обработки данных в отдельных потоках, в которых создается содержательное сжатое описание изображения (кадра видеопоследовательности), сохраняющее геометрические связи исходного изображения, но имеющее размерность на несколько порядков меньше, чем у него. Основные операции сегментации и систем понимания изображений выполняются без использования пиксельного массива изображения — с построенным сжатым содержательным описанием. На эти операции требуется малое (в среднем менее 10 мс на кадр для всей совокупности задач) время исполнения на стандартных современных персональных компьютерах, даже для видео высокого разрешения. В работе описана многопоточная реализация создания сжатого описания изображения (кадра), которая позволяет довести и без того быструю работу системы до рекордных образцов быстродействия. Также описываются применения к системам понимания дорожных сцен, таким как системы нахождения области дороги и ее окрестности, области неба и детектирования и понимания дорожной разметки (постоянной белой и цветной, временной), а также к поиску сигнальных огней вертолетов. Приводятся и обсуждаются примеры обработки конкретных дорожных сцен и даются оценки быстродействия системы на видеопоследовательностях реальных дорожных сцен.

DOI: 10.31857/S0132347420060047

### 1. ВВЕДЕНИЕ

Технологии обработки изображений для обнаружения объектов или событий часто требуют обработку потока кадров. Вследствие этого появляется множество подходов к обработке видеорядов, в том числе в реальном времени. Их можно разделить как по основным целям, например: автономное вождение, медицинское применение, системы безопасности, отслеживание и сортировка объектов, так и по методам и подходам. Классические методы сегментации [1], методы многоклассовой сегментации, основанные на обучении [2, 3] и методы, основанные на нейросетях [3, 4], дают несколько подходов к решению задач сегментации и большого круга задач понимания изображений в реальном времени. Данные

методы в качестве существенного этапа содержат некоторую процедуру кластеризации в пространстве признаков. Это автоматически приводит к потере малых объектов на изображении. Необходимо также отметить, что при определении близости значений признаков не используется информация о локализации пикселей, в которых принимаются данные значения признаков. Это не позволяет работать с изображениями с сильно неоднородным освещением. Следует также отметить, что возможность распараллеливания не предусмотрена исходно в методах [1–3]. Также важно, что для решения задач управления объектами с обратной связью по зрению, сама пиксельная сегментация является лишь промежуточным результатом, и требуется дополнительная боль-

шая работа, чтобы получить данные для управления.

Кроме того, имеется много подходов, ориентированных на конкретные специфические задачи, такие как, например, задача распознавания граничных разметок полосы движения автомобиля [4, 5]. Подробный обзор таких методов содержится в [6]. Также данной проблеме посвящена [7], и она содержит полезную библиографию по данной проблеме. У всех вышеперечисленных подходов есть существенные ограничения на область применения, поскольку необходимые вычислительные мощности, обеспечивающие достаточную точность, часто превышают рамки ресурсов, выделенных под задачу. Это обусловлено большим объемом информации, получаемой из видеопоследовательностей и, как следствие, сложностью алгоритмов обработки. Для ускорения обработки видеоряда часто используются многопоточные алгоритмы, выполняемые как на CPU, так и на GPU [8, 9]. Данный подход широко применим, поскольку алгоритмы обработки, работающие с матрицами пикселей, могут быть преобразованы в многопоточные алгоритмы не только на уровне языка программирования, но и на функциональном уровне. Благодаря развитию современных компьютерных архитектур, параллельное программирование является областью науки с большим потенциалом. Использование технологий многопоточного программирования может значительно сократить время расчетов, ускорив достижение основной цели программы.

Дополнительные возможности построения эффективных систем понимания изображений обеспечивает подход, при котором алгоритмы решения задачи сегментации заранее подразумевают возможность их дальнейшей адаптации для работы в многопроцессорных системах, а результаты сегментации записаны в сжатом виде, непосредственно подходящем для практического применения в задачах, использующих компьютерное зрение. Такими свойствами обладает метод геометризованных гистограмм, разработанный в [10–12] и обрабатывающий изображение как в черно-белом, так и в цветном форматах. В данной работе будет рассматриваться этот метод, его программная реализация, а также его адаптация для многопоточного исполнения на CPU. Данный метод является достаточно гибким и позволяет решать как задачи обнаружения и слежения за объектами, так и задачи распознавания. С помощью метода геометризованных гистограмм удается находить на изображениях и кадрах видеопоследовательностей в реальном времени как большие, так и малые объекты: цветные метки (движущиеся и неподвижные), дорогу, обочины, сигнальные огни автомобилей и верто-

летов, дорожную разметку (временную и постоянную), дорожные знаки и сигналы светофоров [11–14]. Все эти задачи можно решать в реальном времени для видео высокого разрешения ( $1280 \times 720$  пикселей и более) на основе сжатого описания изображений. В разделе 2 будет дано описание метода геометризованных гистограмм и его программной реализации. В разделе 3 будет описано применение метода многопоточной обработки для реализации сжатого описания данных видеоряда по методу геометризованных гистограмм. В разделе 4 будет дана блок-схема полученной системы обработки, сегментации и понимания изображений в реальном времени. Также будут описаны применения разработанной системы, даны результаты обработки видеопоследовательностей в реальном времени и приведены результаты оценки быстродействия системы на реальных видеопоследовательностях дорожных сцен.

## 2. ОСНОВНЫЕ ПРИНЦИПЫ МЕТОДА ГЕОМЕТРИЗОВАННЫХ ГИСТОГРАММ

В отличие от основных методов сегментации изображений, метод геометризованных гистограмм построен так, что основная обработка видеоданных будет вестись параллельным образом. В основе метода геометризованных гистограмм лежит основная идея математической редукции сложных задач. Эта идея предполагает введение новых понятий и логических систем, таких, что вложение в эти системы исходных задач и переформулирование их в новых терминах делает их разрешимыми с помощью новой техники (современная теоретическая математика) или дает эффективные методы их численного решения (вычислительные науки и программирование). Так как пиксельная сегментация является лишь промежуточным продуктом в построении систем понимания изображений и плохо годится для решения задач управления с обратной связью по зрению, то в методе геометризованных гистограмм она в основном не используется.

### 2.1. Конструкция геометризованных гистограмм

С помощью техники, описанной в [10–12], каждому цветному изображению ставится в соответствие структурный граф цветовых сегментов  $STG$ . Чтобы построить  $STG$ , изображение разбивается на полосы одинаковой ширины  $Sr_n$  со сторонами, параллельными горизонтальной или вертикальной оси плоскости изображения  $Ox$ . Вводится понятие геометризованной гистограммы изображения [10], которая является далеким обобщением обычной гистограммы. Однако обычная гистограмма является слабым инвариантом изображения, так

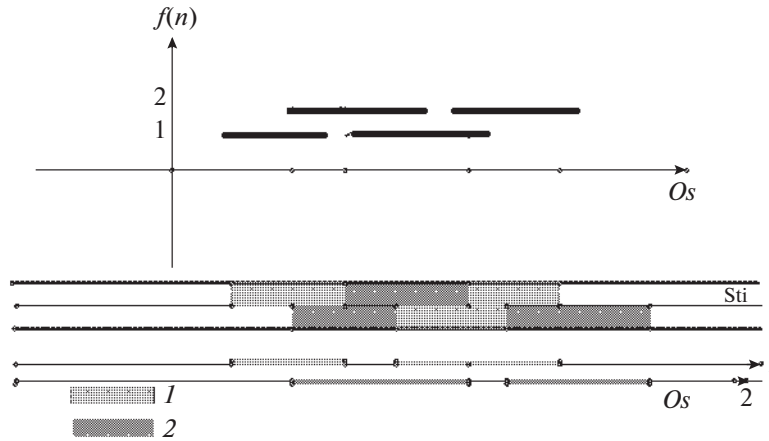


Рис. 1. Получение интервалов геометризованной гистограммы для модельного изображения.

как она остается инвариантной при любом взаимно-однозначном преобразовании прямоугольника изображения. В ней совершенно не учитывается геометрия объектов изображения. В то время как геометризованная гистограмма остается инвариантной только относительно преобразований внутри полос, при которых точки передвигаются перпендикулярно оси  $O_s$ . Так как мы имеем дело с узкими полосами, эти преобразования почти не меняют геометрии объектов, принадлежащих изображению.

Геометризованная гистограмма достаточно точно описывает распределение значений функции, задающей изображение в прямоугольнике кадра. Получается геометризованная гистограмма с помощью проектирования пикселей полосы на ее нижнее основание. Мы будем обозначать подмножества полосы  $St_n$ , в которых функция, задающая изображение, принимает фиксированное значение  $z - L_z$  и называть их множествами уровня  $z$ . Для монохромного изображения [10], ввиду дискретного характера изображений, проекция любого  $L_z$ , на нижнее основание полосы есть объединение интервалов  $Pr(L_z) = \cup_k I_{kz}$  на нижней оси полосы.

Заметим, что описание геометрии множеств  $L_z$  на плоскости изображения есть описание структуры распределения значений функции  $f(x, y)$ , задающей изображение. Описание этой геометрии на всей плоскости носит сложный характер. Использование в задачах сегментации описания распределения значений функции  $f$  выведет решение задачи о сегментации изображения на новый уровень по сравнению с изучением частотных закономерностей в области ее значений. Задача приближенного описания множеств  $L_z$  в узких полосах легко поддается решению. Это решение дается геометризованной гистограммой полосы. Чтобы сравнивать

множество уровней в разных полосах, можно считать, что все интервалы лежат на оси  $O_s$ . Рисунок 1 иллюстрирует получение интервалов геометризованной гистограммы в полосе. В полосе задана функция изображения с двумя значениями, которая постоянна в выделенных прямоугольниках. Под полосой расположены два экземпляра оси  $O_s$ . На них показаны системы интервалов, соответствующие областям одинакового значения функции изображения. На самом деле, эти системы интервалов лежат на одной оси. Два экземпляра взяты для наглядности. При проектировании для каждого интервала  $Sg$  вычисляются его границы  $beg_{Sg}$  и  $end_{Sg}$  на оси  $O_s$ , а также его мощность  $Card^{Sg}$  — количество точек полосы, в которых функция изображения принимает заданное значение и которые проектируются на данный интервал. Объединение систем интервалов по всем полосам хорошо описывает распределение значений монохромной функции, задающей изображение. Эта конструкция обобщается на случай векторной функции, задающей цветное изображение [10]. Удастся при проектировании разделить все множество точек полосы на подмножества, в которых насыщение, оттенок, и яркость варьируются в некоторых диапазонах. Эти подмножества проектируются на интервалы на оси  $O_s$ . В каждой полосе получаются системы интервалов, каждый из которых ( $Sg$ ) характеризуется следующими параметрами:

- Положение интервала  $[beg_{Sg}, end_{Sg}] Sg$  на оси  $O_s$ .
- Диапазон  $\Delta H^{Sg} = [H_{min}^{Sg}, H_{max}^{Sg}]$  и среднее значение оттенка  $H_{mean}^{Sg}$ .
- Диапазон  $\Delta S^{Sg} = [S_{min}^{Sg}, S_{max}^{Sg}]$  и среднее значение  $S_{mean}^{Sg}$  насыщения.

- Диапазон  $\Delta I^{Sg} = [I_{\min}^{Sg}, I_{\max}^{Sg}]$  и среднее значение яркости  $I_{mean}^{Sg}$ .

- Мощность интервала  $Card^{Sg}$  (приблизительно равно числу точек полосы, лежащих в полосе над интервалом, с цветовыми характеристиками, заключенными в интервалах, указанными выше для  $Sg$  на оси  $O_s$ ).

Обозначим  $dens(Sg) = Card^{Sg}/(end_{Sg} - end_{Sg} + 1)$  плотность интервала  $Sg$ . Естественно, если изображение в полосе состоит из однородно закрашенных цветовых пятен с фиксированными цветовыми характеристиками, то подобно рис. 1, каждому такому цветовому пятну при проектировании будет соответствовать интервал геометризованной гистограммы с теми же фиксированными цветовыми характеристиками. Однако изображения сцен реального мира с тенями, областями неоднородного освещения и влажными частями намного сложнее. Геометризованные гистограммы полос таких изображений содержат сложные системы пересекающихся интервалов с заданными диапазонами и средними значениями яркостно-цветовых характеристик. Для того, чтобы давать семантическое описание изображений и решать задачи их понимания на основе геометризованных гистограмм, требуется введение большого числа новых понятий и аксиоматизированных конструкций; также необходимо разработать эффективные программные средства для построения объектов, удовлетворяющих постулированным свойствам и их интерпретации.

Существенным свойством алгоритма получения геометризованной гистограммы в полосе является то, что она может быть получена за один проход массива точек полосы. Это обеспечивает ее получение в реальном времени. Также это является ключевым моментом параллельной реализации метода. Даже для монохромной функции изображения, построение геометризованной гистограммы за один проход изображения является значительной программистской задачей, которая может предлагаться на олимпиадах по программированию. Она реализуется кодом, объем которого более 300 строк. Код программы построения геометризованной гистограммы монохромной функции изображения выложен на [18], поэтому ввиду размера код в статье не приводится. Построение геометризованной гистограммы цветного изображения за один его проход является существенно более сложной программистской задачей. Код ее реализации содержит более 2000 строк. Существенное место в коде занимает сортировка пикселей согласно с их яркостно-цветовыми характеристиками. При сортировке мы используем цветовые координаты ( $G/(G+B)$ ,  $G/(G+R)$ ,  $I$ ), введенные

первым автором в [10]. Здесь  $R$ ,  $G$ ,  $B$  – обычные цветовые координаты, а  $I$  – полутоновая интенсивность. Введем характеристическую функцию  $CF$ . Заметим, что функции  $G/(G+B)$ ,  $G/(G+R)$  задают цветовые координаты на цветовом треугольнике [10]. Если оттенок пиксела принадлежит желтой части цветового треугольника, тогда  $CF$  совпадает с  $G/(G+B)$ . Когда мы движемся в следующий цветовой диапазон (зеленый, голубой, красный), значение  $G/(G+B)$  сдвигается на  $M$ , где  $M$  – число градаций функции  $G/(G+B)$ . Геометризованная гистограмма  $CF$ , дополненная для каждого ее интервала  $I_{kz}$  классической гистограммой другой цветовой координаты  $G/(G+R)$  и диапазоном, и средним значением полутоновой компоненты спроектированных на интервал пикселей, называется геометризованной гистограммой цветного изображения в полосе  $St_i$ . На основе полученных данных для каждого интервала вычисляются диапазоны и средние значения координат  $H$  (оттенок),  $S$  (насыщение),  $I$  (полутоновая интенсивность). Далее характеристики интервалов применяются в этой цветовой системе координат.

Пример геометризованной гистограммы выделенной полосы приведен на рис. 2. Точка в правой части изображения, расположенного в верхней части рисунка, выделяет горизонтальную его полосу. В нижней части рисунка показаны все интервалы геометризованной гистограммы этой полосы. По горизонтальной оси откладываются значения характеристической функции геометризованной гистограммы  $CF$ . По вертикальной оси откладываются координаты концов интервалов (измеренные по горизонтальной оси изображения). Овалами (при движении слева направо) выделены интервалы геометризованной гистограммы, соответствующие: 1) частям белого автомобиля между габаритами и зонами сигналов торможения; 2) частям зоны стоп-сигналов на белом автомобиле, которые имеют красный цвет; и 3) частям соответствующих зон растительности на обочинах (верхний и нижний овалы). Квадраты в верхней части визуализации геометризованной гистограммы показывают цвет того из интервалов, определяемых стоп-сигналами, к которому подведена стрелка мыши.

Так как мы имеем дело с реальными объектами, то им соответствуют наборы интервалов геометризованной гистограммы, яркостно-цветовые характеристики которых варьируются. На сайте [18] выложена программа с инструкциями, которая позволяет построить и визуализировать, как на рис. 2, геометризованную гистограмму любого изображения в формате BMP.

Программная реализация построения геометризованных гистограмм цветного изображения

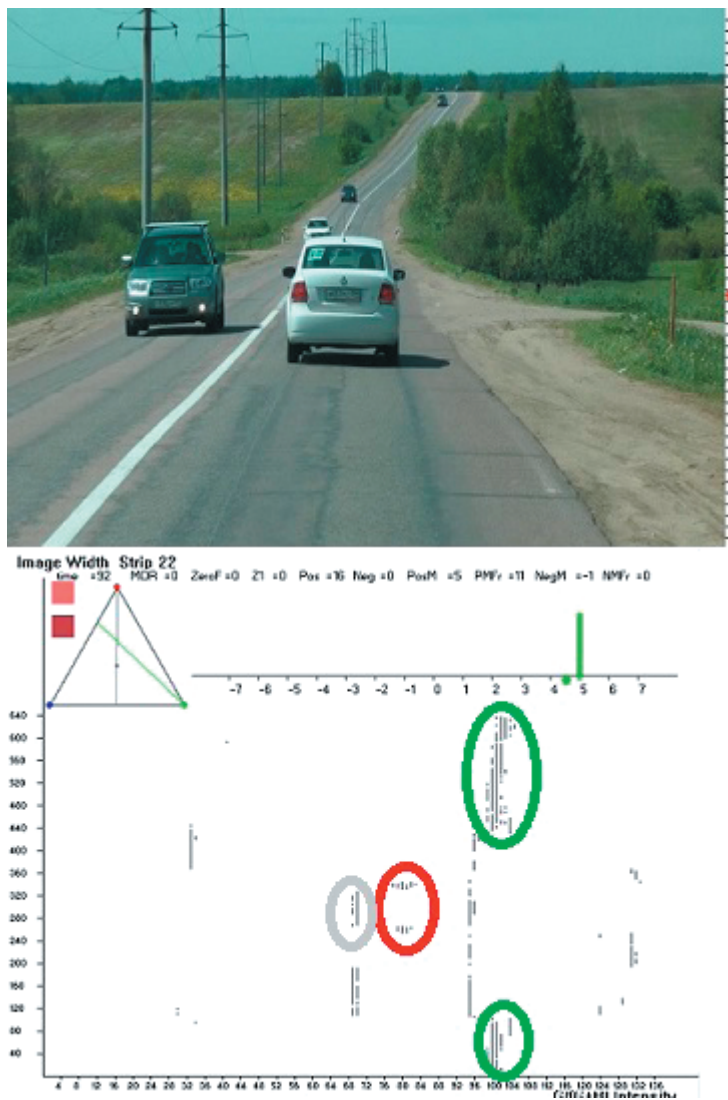


Рис. 2. Геометризованная гистограмма выделенной полосы.

(кадра видеопоследовательности) и его полутоновой компоненты находится в Блоке 1 итоговой блок-схемы программной системы из раздела 4. Для реальных изображений это является правилом, что реальным объектам соответствуют несколько геометрически близких интервалов геометризованной гистограммы, у которых яркостно-цветовые характеристики имеют близкие значения. Объединение геометризованных гистограмм полос дает геометризованную гистограмму всего изображения.

### 2.2. Построение цветовых сгустков и графа STG

Для того чтобы выделить на геометризованной гистограмме полосы части реальных объектов, в ней необходимо применить некоторые процеду-

ры для объединения ее интервалов, им соответствующим. С помощью оригинальной операции кластеризации [10] интервалы геометризованной гистограммы объединяются в кластеры – цветовые сгустки, которые характеризуются такими же, как и у интервалов, яркостно-цветовыми параметрами, мощностью и плотностью. На этом шаге проводится объединение интервалов геометризованной гистограммы, интервалы локализации которых имеют сильное пересечение [10].

Программная реализация этой процедуры разделяется на следующие шаги:

1. Все интервалы геометризованной гистограммы выбрасываются на три отрезка прямой, соответствующей средней линии полосы. На первом отрезке остаются номера интервалов максимальной плотности, на втором – интервалы наиболь-



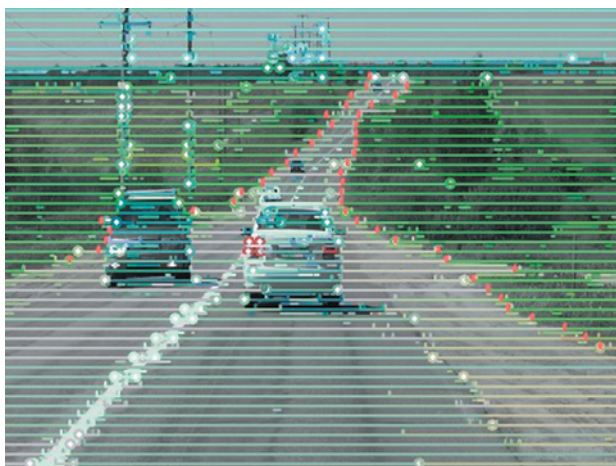


Рис. 3. Цветовые сгустки изображения из рис. 2. Черточками выделены границы зеленой обочины, полученные с помощью программной системы.

шего локального насыщения, на третьем — интервалы максимального значения полутоновой компоненты.

2. Интервалы с номерами, “выжившими” на всех трех отрезках средней линии, являются зародышами, с которых стартует процедура кластеризации.

3. К каждому зародышу присоединяются интервалы гистограммы, которые имеют с ними сильное пересечение [10] и близкие яркостно-цветовые характеристики.

Далее работает процедура объединения цветových сгустков. При этом объединяются цветовые сгустки, имеющие соседние интервалы локализации и сходные яркостно-цветовые параметры. Итоговые цветовые сгустки  $b$  (color bunches в английском варианте термина) характеризуются интегральными параметрами  $\Delta H^b$ ,  $H_{mean}^b$ ,  $\Delta S^b S_{mean}^b$ ,  $\Delta I^b$ ,  $I_{mean}^b$ ,  $Card^b$ , получаемыми суммированием и усреднением соответствующих интервалов геометризованной гистограммы, из которых они состоят. Введем так же, как и для интервалов геометризованной гистограммы, плотность цветового сгустка как  $dens(b) = Card^b / L([beg_b, end_b])$  (мощность, поделенная на длину интервала). Процедуры построения цветовых сгустков и их объединения принадлежат Блоку 2 программы.

Цветовые сгустки объединяются в граф. В полосе соединяются ребром соседние цветовые сгустки (с соседними интервалами локализации), а в соседних полосах — цветовые сгустки, интервалы локализации которых пересекаются. Неформально, каждый сгусток дает описание некоторой части реального объекта в полосе, его проекцию на ось  $Ox$  и описание

значений численных цветовых характеристик этой части объекта.  $STG$  можно интерпретировать геометрически с помощью наложения отрезков его сгустков ( $[beg_b, end_b]$ ) на центральную линию соответствующей полосы и окрашивания этих отрезков в цвет, определяемый  $H_{mean}^b$ ,  $S_{mean}^b$ ,  $I_{mean}^b$ . Пример множества цветовых сгустков изображения с рис. 2, наложенных на его полутоновую компоненту, можно посмотреть на рис. 3. Примеры цветовых сгустков, наложенных на цветные изображения, можно также посмотреть в открытом доступе в [13, 14].

Цветовые сгустки являются аналогом суперпикселей, применяемых в классических методах сегментации. Также с помощью программы из [18] можно построить и визуализировать множество цветовых сгустков для любого изображения в формате BMP. Приведенные изображения цветовых сгустков и получаемые с помощью программы из [18] показывают, что граф цветовых сгустков хорошо описывает яркостно-цветовые свойства изображений и геометрию реальных объектов в полосах и в изображении в целом.

### 2.3. Построение поисковой решетки на $STG$

На множестве цветовых сгустков строится “решетка поиска”  $SearchLat(STG)$  [12], которая позволяет производить глобальный анализ изображения. Если мы положим на среднюю линию каждой полосы разбиения изображения интервалы геометрической локализации  $[beg_b, end_b]$  всех цветовых сгустков полосы, то получим некоторое ее покрытие.

**Определение 2.1.** Цветовые сгустки, имеющие в некоторой точке средней линии максимальную плотность, называются доминирующими цветовыми сгустками. Остальные сгустки называются доминируемыми.

Ясно, что доминирующие цветовые сгустки образуют покрытие средней линии. Оказывается [12], что всегда можно выбрать линейно-упорядоченную последовательность базисных цветовых сгустков, которые образуют покрытие средней линии. Алгоритм для построения  $SearchLat(STG)$  состоит из следующих шагов:

1. Для каждого доминирующего сгустка найдутся доминирующие сгустки слева и справа, интервалы локализации являются соседними для интервала локализации выбранного сгустка.

2. Начиная с левого конца, строится упорядоченная система доминирующих цветовых сгустков, соседних друг другу.

Доминирующие цветовые сгустки, включенные в линейно-упорядоченное покрытие, назы-

ваются базисными цветовыми сгустками. Базисные цветовые сгустки всех полос образуют решетку поиска изображения  $\text{SearchLat}(STG)$  [12].

В дополнение к поисковой решетке строится структура  $\text{BunchLoc}(STG)$ . В каждой полосе для любой точки  $x$  оси  $O_s$  данная структура указывает номера всех цветовых сгустков, которые проходят через данную точку. Программа, реализующая построение структуры  $\text{BunchLoc}(STG)$ , реализуется в два шага:

1. Перебираются все цветовые сгустки, и для каждой точки  $x \in O_s$  определяется число сгустков, проходящих через данную точку. Вычисляется  $N_b$  — максимальное число сгустков, которые могут проходить через произвольную точку оси  $O_s$ .

2. Строится массив, высота которого равна  $N_b$  и который содержит для каждой точки  $x \in O_s$  список цветовых сгустков, проходящих через эту точку.

Структура  $\text{BunchLoc}(STG)$  позволяет для каждого цветового сгустка  $b$  определить, какие цветовые сгустки в соседних полосах связаны с данным сгустком ребром в  $STG$  (имеют интервалы локализации, пересекающиеся с интервалом локализации  $b$ ). Структуры  $\text{SearchLat}$  и  $\text{BunchLoc}$  позволяют находить все соседние цветовые сгустки для выбранного сгустка  $b$ . После выполнения операций сегментации, цветовым сгусткам присваиваются метки объектов, которым они принадлежат. Эти структуры позволяют строить отношения соседства для выделенных на изображении объектов и выполнять глобальный анализ их взаимодействия [12]. Их конструирование выполняется в Блоке 3.

#### 2.4. Построение глобальных объектов

Чтобы решать задачи поиска объектов с заданными геометрией и яркостно-цветовыми характеристиками на изображениях и задачи понимания изображений, необходимо строить образы частей предполагаемых реальных глобальных объектов в  $STG$ . Под глобальными объектами мы понимаем объекты, расположенные более чем в одной полосе изображения. Реальные глобальные объекты могут быть двух типов. К первому типу относятся областные объекты. Они в полосах, как правило, при проектировании на  $STG$  индуцируют доминирующие цветовые сгустки. Глобальные объекты второго типа — это малые и тонкие объекты, такие как дорожная разметка, строительные конусы для обозначения запретных зон на дороге, знаки аварийной остановки, дорожные знаки, стоп-сигналы автомобилей и т.д. Объекты такого типа могут при проектировании давать как доминирующие (на близком расстоянии), так и доминируемые цветовые сгустки. Целый объект

или его часть, лежащие в нескольких полосах, дают в некотором смысле непрерывные системы цветовых сгустков в  $STG$ . Для нахождения образов таких объектов, необходимо формализовать понятие непрерывной системы цветовых сгустков. Многие задачи по поиску ориентиров и объектов в кадре, можно переформулировать строго как задачи поиска некоторых непрерывных абстрактных объектов на графе цветовых сгустков.

Для этих целей в [11] введены понятия левых и правых контрастных кривых или левых и правых ростков глобальных объектов на  $STG$ , и определен двудольный граф левых и правых контрастных кривых  $LRG$ . Если изображение разбито на горизонтальные полосы, то, неформально, левая или правая контрастная кривая (росток глобального объекта) есть цепочка доминирующих цветовых сгустков в соседних полосах с подобными цветовыми характеристиками. При этом левые или правые концы цветовых сгустков меняются от полосы к полосе “непрерывно”, и соседние в той же полосе слева или справа цветовые сгустки имеют контрастные цветовые характеристики. Эта цепочка строится снизу вверх, переходя из полосы в полосу [11]. Для того, чтобы избежать операций полного перебора, непрерывное продолжение цветового сгустка в соседние полосы (построение левых и правых ростков глобальных объектов) осуществляется с использованием структуры  $\text{BunchLoc}$ . Конструкция левых и правых непрерывных цепочек контрастных доминирующих цветовых сгустков (левых и правых ростков глобальных контрастных объектов) выполняется в Блоке 4 программной системы. Также в этом блоке строится двудольный граф связи между левыми и правыми ростками глобальных объектов. Двудольный граф  $LRG$  показывает, какие левые и правые ростки в полосах можно соединить цепочками цветовых сгустков без контраста. Следовательно, эти ростки могут быть частями одного объекта. Для решения различных задач понимания изображений необходимо находить прямолинейные участки в граничных элементах построенных левых и правых ростков. Методы, основанные на методе наименьших квадратов, являются неустойчивыми относительно ошибок сегментации. Устойчивый метод, основанный на анализе гистограмм углов наклона отрезков, соединяющих концы интервалов локализации сгустков, входящих в построенный росток глобального объекта, разработан в [12]. Часть программы, реализующая предложенный метод, находится в Блоке 5 диаграммы. Этот блок заканчивает часть программного комплекса, который дает аппарат для решения различных задач понимания изображений. Общий объем кода больше 40000 строк. Предполагается



**Рис. 4.** Непрерывная слева система цветовых сгустков, соответствующая части зеленой обочины.

выложить части кода в интернет. Также будет выложена в [18] dll, которая реализует все перечисленные модули для общего анализа изображений. Программный комплекс, который позволяет строить ростки левых и правых глобальных объектов для любого цветного изображения выложен на сайт [18]. Пример левого ростка глобального множества для изображения рис. 2 приведен на рис. 4.

Подобные определения вводятся для поиска малых и тонких объектов. Единственная разница состоит в том, что в цепочках цветовых сгустков участвуют контрастные доминируемые сгустки. На этом языке могут быть переформулированы многие классические задачи понимания изображений.

Например, так можно сформулировать задачу детектирования и понимания дорожной разметки [14]. При этом удастся отыскивать как белую, так и цветную разметку, производить нахождение временной цветной разметки при наличии старой белой и анализировать криволинейную разметку также хорошо, как и прямолинейную.

Поисковая решетка SearchLat активно используется при сегментации и решении задачи понимания изображений. С помощью специальных массивов описывается, какие левые и правые ростки проходят через любой цветовой сгусток. Также отмечается тот факт, что таких ростков не имеется. С помощью SearchLat легко строятся отношения соседства для ростков глобальных множеств [12], что при обычных подходах представляет серьезную проблему.

Таким образом, готовится процедура объединения ростков. Также с помощью SearchLat строится система рассуждений, которая позволяет выделять кандидатов на часть дорожной разметки в полосе.

Так как кандидаты на части дороги всегда принадлежат SearchLat, то для любого цветового сгустка достаточно проанализировать его связи в яркостно-цветовой и в геометрической областях с соседними базисными сгустками SearchLat, чтобы отнести его к кандидатам на часть разметки в полосе.

Поисковая решетка SearchLat и структура BunchLoc также позволяют выделять мелкие контрастные объекты на изображениях. Используя структуру BunchLoc, для каждого доминируемого цветового сгустка  $b$  можно определить все базисные цветовые сгустки из SearchLat, интервалы локализации которых пересекают интервал локализации  $b$ .

**Определение 2.2.** Доминируемый цветовой сгусток  $b$  является контрастным цветовым сгустком в  $STG$ , если он имеет контрастные цветовые характеристики со всеми базисными сгустками из SearchLat, с которыми он имеет пересекающиеся интервалы локализации.

*Замечание.* Для доминирующих сгустков определение контрастности было дано в [11]. Оно утверждает, что доминирующий сгусток – контрастный, если его яркостно-цветовые характеристики контрастны к яркостно-цветовым характеристикам соседних доминирующих сгустков.

Используя это определение, с помощью формальных процедур, на  $STG$  находятся образы контрастных малых объектов из изображения, таких как фары автомобилей, сигнальные огни летательных объектов, сигналы светофоров, стоп-сигналы и сигналы поворотов автомобилей, и т.д. Для подтверждения истинности найденных малых объектов используются данные глобальной сегментации. Эти данные подтверждают, что малый контрастный объект находится рядом с большим контрастным объектом (сигнальные огни вертолета) или внутри него (стоп-сигналы автомобиля). Взаимодействие локального и глобального анализа обеспечивает надежность идентификации.

Дальнейшие операции сегментации и построения систем понимания изображений выполняются без обращения к массиву изображения, путем использования только графа цветовых сгустков и левых, правых ростков глобальных объектов. При этом вместо миллионов пикселей изображения используются только несколько тысяч цветовых сгустков. Отметим, что на данной стадии (сборка объектов из частей) может быть использована семантическая информация, так как мы знаем описание геометрии частей и их яркостно-цветовые характеристики. Также очень важны отношения соседства, полученные с помощью SearchLat. Время, затраченное на сегментацию и решение большого числа задач понимания изображений, при, например, разбиении на 60 полос изображений с



разрешением порядка  $1600 \times 1200$ ,  $1280 \times 720$ , менее 100 мс. Основное время тратится на построение графа цветных сгустков. При таких разрешениях и при последовательной обработке полос обеспечивается быстродействие 8–12 кадров в секунду. Ясно, что при большом числе процессоров и распараллеливании обработки полос изображения, данное время может быть сильно сокращено, почти до нуля, так как время обработки одной полосы и так небольшое.

В работе предлагаются методы этого сокращения и даются оценки полученного сокращения для нескольких типов современных персональных компьютеров с процессорами с несколькими ядрами и несколькими логическими процессорами. Заметим, что даже простейшие реализованные методы распараллеливания приводят к рекордным результатам решения задач сегментации и построения систем понимания изображений на видео высокого разрешения.

### 3. МНОГОПОТОЧНАЯ АДАПТАЦИЯ МЕТОДА ГЕОМЕТРИЗОВАННЫХ ГИСТОГРАММ И ОПИСАНИЕ ИТОГОВОЙ СИСТЕМЫ

Данный раздел посвящен простейшей параллельной адаптации метода геометризованных гистограмм. В последующих работах планируется рассмотреть другие способы распараллеливания представленного алгоритма обработки изображений.

Стоит отметить, что для достижения хорошего прироста к скорости обработки видеопоследовательностей, было необходимо в дополнение к алгоритмическому описанию, приведенному в предыдущем разделе, провести комплексный анализ программной реализации рассматриваемого метода анализа изображений.

Приведем краткую характеристику приложения, содержащего исходную реализацию метода геометризованных гистограмм.

Данное приложение написано на языке программирования C++ версии 2003, для отображения графического интерфейса и воспроизведения видеопоследовательностей используется пакет Microsoft Foundation Classes (MFC). Разработка ведется в среде Microsoft Visual Studio 2017.

Для обеспечения возможности запуска приложения на других операционных системах было принято решение перейти к использованию классов STL для работы с файловой системой, библиотеки OpenCV – для работы с видеофайлами, а для сборки исполняемых файлов – использовать утилиту CMake.

Для реализации многопоточного исполнения приложения были применены относительно но-

вые возможности стандартной библиотеки C++, появившиеся, начиная с 11 версии, а именно, классы из пространств имен `std::thread`, `std::mutex` и их методы, определенные в заголовочных файлах `<thread>` и `<mutex>`, соответственно.

Было определено, что наибольшее время занимает исполнение функций, отвечающих за этап сегментации изображений видеопоследовательности. Подробно ознакомиться со связанными метриками можно в следующем разделе.

Метод геометризованных гистограмм изначально спроектирован так, что каждая полоса, на которую разделяется изображение, может быть обработана по отдельности. Поэтому было решено попробовать применить многопоточное программирование именно к обработке этих полос. Стоит отметить, что метод не сводится только к обработке каждой полосы. На данном этапе осуществлена параллельная реализация той части процедуры сегментации, которая занимает 95% времени выполнения в однопоточном режиме. Поэтому не удалось добиться ускорения программы в количестве раз, линейно зависящее от числа физических ядер процессора.

За хранение полос изображения отвечает отдельный класс `CStrip`. В однопоточной реализации на этапе сегментации выполняется несколько циклов, однако есть только один, наиболее требовательный к вычислительным ресурсам, назовем его “главным циклом”. Он производит необходимые действия над объектами указанного класса. В свою очередь, внутри “главного цикла” производится вызов нескольких других функций (около четырех), время исполнения которых в среднем варьируется от 0.04 мс на кадр до 45 мс (при обработке изображения с разрешением  $640 \times 480$  пикселей).

Данный цикл был заменен на одновременное исполнение некоторого количества независимых потоков. В ходе экспериментов было выявлено, что для достижения наиболее оптимального прироста скорости обработки изображений, необходимо запускать число потоков, равное числу полос, на которые разрезается исходное изображение (в зависимости от разрешения, этот параметр устанавливается различным, но чаще всего равен 60 или 72). Стоит отметить, что выбор такого количества потоков (большего, чем количество физических ядер центрального процессора) обусловлен необходимостью обеспечения достаточной универсальности для работы программы с видеопоследовательностями различных разрешений на различных компьютерах, в том числе, на бортовом сервере беспилотного автомобиля.

Для обеспечения корректного одновременного доступа к данным изображения (объект класса

```

auto begin = std::chrono::steady_clock::now();
vision.segment(resized_fr, fr_number); //start segmentation
vision.visualize_markings(resized_fr); //visualize markings
//calculate one frame's time for current frame
float secs = std::chrono::duration_cast<std::chrono::milliseconds>
    (std::chrono::steady_clock::now() - begin).count() / 1000.0;
sum += secs; //all time counter increment

```

Рис. 5. Метод расчета времени выполнения функций с помощью *std::chrono*.

*cv::Mat* библиотеки OpenCV) используются блокировки с помощью *std::mutex*.

Стоит отметить, что даже на ноутбуке средней ценовой категории 2017 года, имеющем процессор Intel Core i7 с четырьмя физическими ядрами, время обработки одного кадра видеопоследовательности, получаемой в реальном времени с камеры высокого разрешения, уменьшилось более чем в 3 раза и составило менее 30 мс.

#### 4. ПРИМЕНЕНИЯ, РЕЗУЛЬТАТЫ И ОЦЕНКИ

Для оценки скорости работы программы применяются возможности стандартной библиотеки C++, реализованные в классах из пространства имен *std::chrono*. Способ получения временного значения в миллисекундах представлен на рис. 5.

В ходе тестов исходной версии приложения производился замер времени выполнения всей процедуры сегментации в целом, а также более мелких функций, вызываемых из “главного цикла”. Также производилось сравнение с результатами, полученными при запуске многопоточной версии.

Для большей наглядности на рис. 6 приведена схема работы алгоритма, а также выделены те его части, которые теперь выполняются в многопоточном режиме.

Данные собирались при обработке трех качественно разных видеорядов на трех ноутбуках с процессорами уровня Intel Core i5/i7 (седьмого и

восьмого поколений), обладающими четырьмя физическими ядрами. Затем производилось усреднение данных. В приведенных далее графиках и таблицах время обработки кадра является вычисленным средним арифметическим по 1000 кадрам из каждого видео.

Считывание и предобработка исходной видеопоследовательности, в частности, уменьшение разрешения, производилось с помощью библиотеки OpenCV.

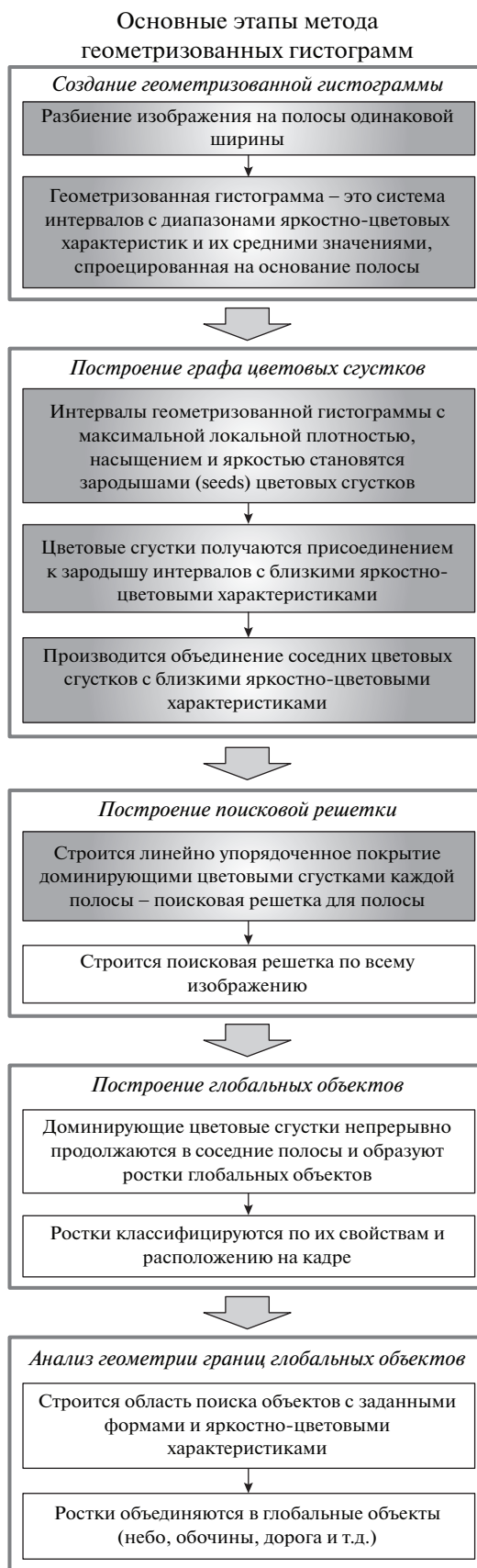
Стоит учитывать тот факт, что временные затраты на работу функций библиотеки OpenCV (включая считывание видеопотока, изменение разрешения изображения и различную визуализацию на кадре, например, отрисовку найденной методом геометризованных гистограмм дорожной разметки) также малы и составляют около 5–10 мс на один кадр.

Представленная на рис. 7 таблица 1 отражает полученную экспериментальную информацию о времени (в миллисекундах), необходимом на выполнение сегментации одного кадра разными версиями программы (столбец *multithreading state*), реализующей метод геометризованных гистограмм. Столбец *num of strips* отражает количество полос, на которые разрезается изображение, и соответствует количеству итераций “главного цикла” в исходной реализации метода и числу запускаемых потоков в многопоточной версии, соответственно.

На рис. 8 для наглядности представлены графики, содержащие аналогичные данные: по вер-

Таблица 1.

Multithreading state	Num of strips	640 × 480	1280 × 720	1920 × 1080
Off	48	25.381	59.1323	118.512
On	48	10.2064	18.8395	36.169
Off	72	29.9551	66.3633	124.652
On	72	16.3946	22.6448	38.6855
Off	108	35.3268	76.5239	136.415
On	108	24.1019	29.0153	41.0169



**Рис. 6.** Блок-схема этапов работы (блоков программного комплекса) метода геометризованных гистограмм, серым фоном выделены параллельно исполняемые части метода.

Multithreading state	Num of strips	640×480	1280×720	1920×1080
Off	48	25.381	59.1323	118.512
On	48	10.2064	18.8395	36.169
Off	72	29.9551	66.3633	124.652
On	72	16.3946	22.6448	38.6855
Off	108	35.3268	76.5239	136.415
On	108	24.1019	29.0153	41.0169

Рис. 7. Сравнение временных затрат на сегментацию одного кадра однопоточной и многопоточной реализациями метода геометризованных гистограмм.

тикальной оси – время на его обработку в миллисекундах, по горизонтальной – разрешение обрабатываемого изображения в пикселях.

Программный комплекс, реализованный на основе предложенного подхода, имеет много применений.

На учебном роботе Амуре было реализовано решение задачи поиска и распознавания движущейся трехцветной метки. Робот Амур, управляемый в реальном времени, с обратной связью по зрению, осуществлял движение за человеком, несущим в руке табличку с меткой. При этом человек стремился всячески запутать робота, выходя из кадра и пряча метку. Работа робота демонстрировалась на форуме роботов в присутствии в кадре других людей и различных плакатов на стенах с похожими цветными частями. Робот был запущен десятки раз и всегда осуществлял безошибочное слежение. Запись с камеры робота при движении может быть найдена на сайтах [16, 17].

На основе данного подхода реализуется подсистема компьютерного зрения системы управления беспилотного автомобиля “АвтоНива” [15, 18].

Для работы программы с видеопотоком реального времени в условиях задач, связанных с роботом-автомобилем Нивой, где камеры, применяемые в экспериментах, снимают с частотой 30 кадров в секунду, был выбран оптимальный режим работы программы, а именно, с разбиением кадра на 72 полосы и разрешением  $1280 \times 720$ , при котором время обработки составляет в среднем 22.6 миллисекунды на одно изображение. Именно этот режим в условиях ограниченных вычислительных мощностей дает наиболее детализированное распознавание кадра.

В настоящее время разработаны и опробованы на видеопоследовательностях дорожных сцен блоки поиска обочины, неба, дорожной разметки. Результаты обработки могут быть найдены на сайтах [16, 17]. Разработаны алгоритмы и отлаживаются программы поиска и распознавания дорожных знаков (в первую очередь, знаков, связанных с распознаванием выезда на перекрестки), поиска и распознавания сигналов светофоров, нахождения других участников движения в собственной и соседних полосах движения. Также разработана программа поиска сигнальных огней вертолета, и она успешно опробована на имеющихся видеозаписях. Это подтверждает способность метода находить очень маленькие окрашенные объекты на фоне объектов с похожими яркостно-цветовыми характеристиками (небо, растительность). Также применение методов глобального анализа и полная сегментация изображения подтверждают принадлежность сигнальной лампы большому объекту (вертолету, дрону, etc.). Сложный состав образа лампы (крайняя часть имеет красный цвет, а

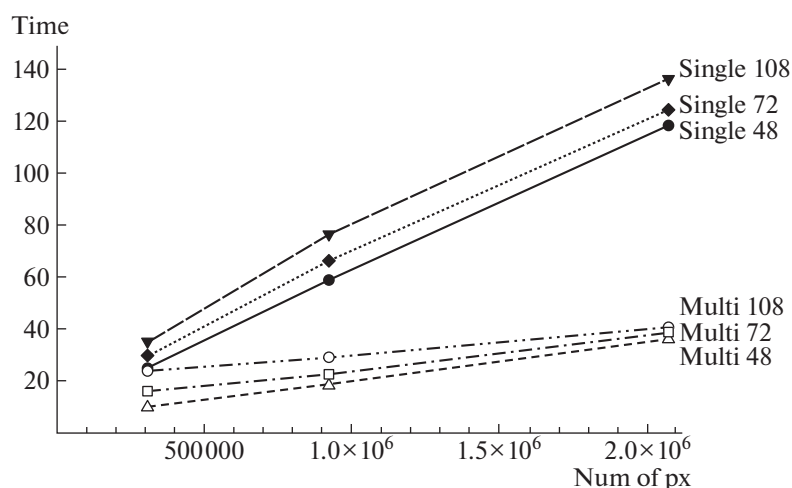


Рис. 8. Сравнение времени обработки одного кадра в однопоточной и многопоточной версиях приложения.

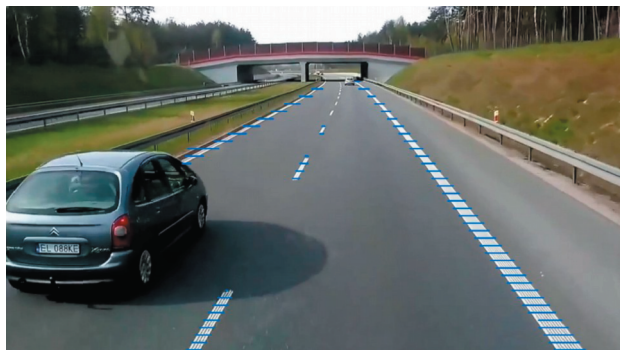


Рис. 9. Пример кадра с выделенной дорожной разметкой.

внутренняя — оранжевый) также удается фиксировать при анализе, что позволяет различить объект на фоне помех. Необходимо также отметить, что разработанные методы нахождения дорожной разметки могут быть применены для нахождения разметок взлетно-посадочной полосы для обеспечения автоматической посадки летательных объектов.

Рисунки 9, 10 показывают результаты обработки кадров видеопоследовательностей для поиска дорожной разметки.

Демонстрируемые примеры показывают работу системы детектирования и понимания разметки в случае заслонения части разметки другим объектом, и когда освещение сильно неоднородно, например, при заезде автомобиля под мост. Существенно, что система правильно находит разметку вне зависимости от интенсивности разметки и ее окружения. Также система демонстрирует хорошие результаты для частично стертой осенне-весенней разметки. Результаты обработки видеопоследовательностей с целью поиска разметки и других объектов на дороге будут размещаться на сайтах [17, 18].



Рис. 10. Пример распознавания дорожной разметки в разных условиях освещенности полотна дороги.

## 5. ЗАКЛЮЧЕНИЕ

В работе описана многопоточная программная система для сегментации и понимания изображений и кадров видеопоследовательностей в реальном времени. Описаны основные методы и алгоритмы системы. Приведены оценки времени работы системы при решении комплекса задач понимания дорожных сцен. Демонстрируются примеры работы подсистемы поиска разметки на дороге.

Адаптация метода геометризованных гистограмм для работы в многопоточном режиме позволила ускорить обработку одного кадра видеопоследовательности в среднем более чем в 3 раза и добиться хорошего показателя — менее 30 мс на кадр. Такое время обработки можно считать достаточно малым даже при анализе получаемой в реальном времени видеопоследовательности, поскольку большинство современных камер делают 15–60 кадров в секунду.

В дальнейшем планируется применить технологии параллельного программирования на более глубоких уровнях обработки изображений, а также провести замеры времени выполнения программы на более мощном оборудовании, обладающем большим количеством физических ядер процессора.

## СПИСОК ЛИТЕРАТУРЫ

1. Felzenszwalb P.F., Huttenlocher D.P. Efficient graph-based image segmentation // International Journal of Computer Vision. 2004. V. 59. № 2. P. 167–181.
2. Shotton J., Winn J., Rother C., Criminisi C.A. TextonBoost for image understanding multi-class objects recognition and segmentation by jointly modeling texture, layout, and context // International Journal of Computer Vision. 2004. V. 81. № 1. P. 2–23.
3. Badrinarayanan V., Kendall A., Cipola R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation // IEEE Trans. PAMI. 2017. V. 39. № 12. P. 2481–2495.
4. Tian Y., Gelernter J., Wang X. Lane marking detection via deep convolutional neural network // Neurocomputing. 2018. V. 280. P. 46–55.
5. Lin G., Santoso P.S., Lin C., Tsai C., Guo J. One Stage Detection Network with an Auxiliary Classifier for Real-Time Road Marks Detection // 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA. 2018. P. 1379–1382.
6. Norote S.P., Bhujbal P.N., Norote A.S., Dhane D.M. A review of recent advances in lane detection and departure warning system // Pattern Recognition. 2018. V. 73. № 2. P. 216–234.
7. Son J., Yoo H., Kim S., Sohn K. Real-time invariant lane detection for lane departure warning system Expert Systems with Applications. 2015. V. 42. P. 1816–24.
8. Chen L., Li J., Zhou J., Jiang M. “Multithreading Method to Perform the Parallel Image Registration”, 2009 International Conference on Computational In-



- telligence and Software Engineering, Wuhan. 2009. P. 1–4.
9. *Saaidon N., Sediono W.* “Multicolour object detection using multithreading”, 2015 IEEE Symposium on Computer Applications & Industrial Electronics (IS-CAIE), Langkawi. 2015. P. 48–52.
  10. *Kiy K.I.* A new real-time method for description and generalized segmentation of color images // Pattern Recognition and Image Analysis. 2010. V. 20. № 2. P. 398–401.
  11. *Kiy K.I.* Segmentation and detection of contrast objects and their application in robot navigation // Pattern Recognition and Image Analysis. 2015. V. 25. № 2. P. 338–346.
  12. *Kiy K.I.* A new method of global image analysis and its application in understanding road scenes // Pattern Recognition and Image Analysis. 2018. V. 28. № 3. P. 483–494.
  13. *Kiy K.I.* An image understanding system based on the geometrized histograms method: finding the sky in road scenes // CEUR Workshop Proceedings. V. 2210. P. 291–299. <http://ceur-ws.org/Vol-2210/paper38.pdf/>.
  14. *Dosaev R.V., Kiy K.I.* A new real-time method for finding temporary and permanent road marking // CEUR Workshop Proceedings. V. 2391. P. 86–96. <http://ceur-ws.org/Vol-2391/paper12.pdf/>
  15. *Подопросветов А.В., Павловский В.Е., Кий К.И., Анохин Д.А.* Робомобиль Нива: алгоритмы управления. Сборник трудов ПУМСС XXI Международной конференции. 2019. Т. II. С. 102–106.
  16. [http://video.mail.ru/mail/kikip\\_46/\\_myvideo/](http://video.mail.ru/mail/kikip_46/_myvideo/)
  17. <https://www.facebook.com/100004887018729/videos>
  18. <http://project1054516.tilda.ws/>