

УДК 004.932.2

ИСПОЛЬЗОВАНИЕ ОКОННОГО ПРЕОБРАЗОВАНИЯ ХАФА ДЛЯ ПОИСКА ПРОТЯЖЕННЫХ ГРАНИЦ НА ИЗОБРАЖЕНИИ

© 2020 г. Е. И. Панфилова^{1,2,3,*}, И. А. Кунина^{1,2,4}¹ Институт проблем передачи информации им. А.А. Харкевича РАН
127051 Москва, Большой Каретный переулок, 19, Россия² ООО “Смарт Энджинс Сервис” 117312 Москва, проспект 60-летия Октября, 9, Россия³ Институт проблем управления им. В.А. Трапезникова РАН 117997 Москва, Профсоюзная улица, 65, Россия⁴ Московский физико-технический институт (НИУ) 141701 Долгопрудный, Институтский переулок, 9, Россия

*E-mail: e.panfilova@smartengines.com

Поступила в редакцию 30.03.2019 г.

После доработки 21.04.2019 г.

Принята к публикации 29.04.2019 г.

В работе предложен алгоритм нахождения протяженных границ на изображении. Рассматривается случай, когда граница может быть приближена ломаной с ограничением на угол излома. Задачи поиска таких границ возникают при детектировании линий дорожной разметки, построении карты дорог по космическим снимкам Земли, поиске дислокаций в кристаллах на отдельных проекциях рентгеновской топо-томографии. Для поиска звеньев ломаной изображение обрабатывается скользящим окном, в каждом новом положении которого ищется прямолинейный сегмент при помощи быстрого преобразования Хафа. Далее найденные сегменты в зависимости от взаимного расположения объединяются в группы, покрывающие искомые границы, и аппроксимируются ломаными. Предложенный алгоритм был использован в задаче детектирования линий дорожной разметки для определения беспилотным транспортным средством (БПТС) своего положения на заданной векторной карте дорог. Алгоритм был протестирован на реальных данных, собранных с фронтальной камеры БПТС при проезде на полигоне “Калибр” (г. Москва). Точность детектора линий дорожной разметки составила 43%, полнота – 73%. Точность локализации БПТС с его использованием составила 0.2 м в евклидовой метрике на маршруте, что в 8 раз меньше, чем локализация без использования информации о дорожной разметке. Также алгоритм был протестирован на отдельных изображениях ДЗЗ и топо-томограммах.

Ключевые слова: детектирование границ, линии дорожной разметки, протяженные дислокации, быстрое преобразование Хафа, оконный анализ изображений

DOI: 10.31857/S0235009220030075

ВВЕДЕНИЕ

Задача поиска протяженных границ на изображении возникает во многих прикладных системах компьютерного зрения. Например, протяженной границей можно задать положение линии дорожной разметки на изображении дороги (Ding et al., 2017) (рис. 1, *в*), отдельной полосы автодороги из дорожной сети на данных дистанционного зондирования Земли (ДЗЗ) (Kunina et al., 2019) (*а*) или дислокации в кристалле (*б*).

Часто задачу поиска границ решают с помощью алгоритмов поиска краев, таких как детектор краев Кэнни (Canny, 1986) или других менее распространенных детекторов, например, приведенных в работах (Mousavi et al., 2019; Fan et al., 2019). Такой подход отражен в работах (Li et al., 2016; Ma et al., 2012). Если же граница является

линией на изображении, то для поиска таких границ используют алгоритмы выделения хребтов (Lopez et al., 2005; Тропин и др., 2019).

Заметим, что такие алгоритмы никак не ограничивают кривизну (угол излома) и протяженность искомых границ, что требуется, например, при детекции дорожной разметки или автодорог.

Подходы, в основе которых лежат алгоритмы активных контуров (Wang et al., 2004; Anil et al., 2010), учитывают возможную кривизну границы, но качество таких подходов зависит от выбора контрольных точек (не всегда автоматического) и наличия априорной информации.

Еще один подход к решению таких задач – использование методов обучения машин (Kirthika et al., 2011; Xiao et al., 2016). Однако качество работы таких методов зависит от вариативности и разме-

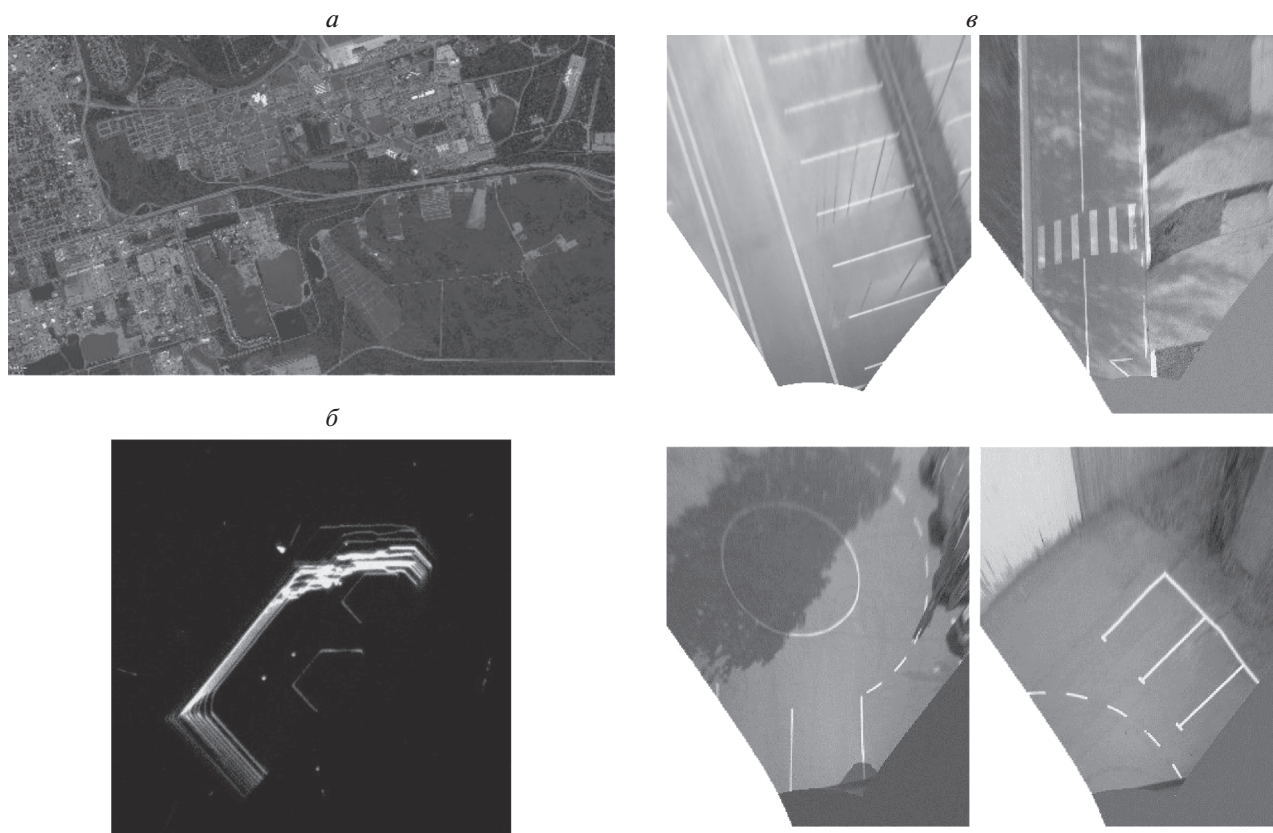


Рис. 1. Примеры данных для тестирования предлагаемого алгоритма.

a – снимок ДЗЗ города Фэрбанкс, Аляска, сделанный с высоты ≈ 7 км в видимом диапазоне; *b* – рентгеновская топограмма дислокаций в кристалле Si (111); *c* – снимки дороги с полигона “Калибр”, сделанные при проезде БПТС по заданному маршруту.

ра обучающей выборки, что не подходит в условиях малого количества входных данных.

В данной работе предлагается алгоритм поиска протяженных границ на изображении, ограниченных по протяженности и углу излома. Алгоритм принимает на вход изображение с выделенными границами и выдает множество ломаных, удовлетворяющих заданным ограничениям. В основе алгоритма – предположение, что отдельный участок границы можно приблизить прямолинейным сегментом. Для поиска таких сегментов используется преобразование Хафа (ПХ) (Hough, 1962), другое название которого – дискретное преобразование Радона (Radon, 1917).

Преобразование Хафа представляет собой процедуру голосования в пространстве параметров, определяющих прямую, оценивая таким образом количество точек, лежащих на каждой прямой на изображении. На настоящий момент существует множество модификаций классического ПХ (Hough, 1962), обзор которых можно найти в работе (Mukhopadhyay et al., 2015).

В нашей работе использована одна из модификаций ПХ – быстрое преобразование Хафа (БПХ)

в постановке Брейди (Brady et al., 1992). Несмотря на широкую известность классического ПХ, БПХ используется при решении множества разнообразных задач, таких как поиск точек схода (Sheshkus et al., 2020) и определение наклона шрифта (Limonova et al., 2017) в системах распознавания документов; создание панорамы из отдельных фотографий протяженного объекта прямоугольной формы (Limonova et al., 2018); быстрый подсчет лучевых сумм в вычислительной компьютерной томографии (Vulatov et al., 2020); отслеживание объектов, содержащих множество концентрических дуг, в видеопотоке (Котов и др., 2015) и другие.

Для поиска отрезков, которые будут локально приближать искомые границы, изображение обрабатывается скользящим окном. Затем при помощи БПХ в каждом новом положении окна ищется не более одной превалирующей прямой, которая задается границами рассматриваемого окна. Этот процесс мы и будем называть оконным преобразованием Хафа, идеи которого были заложены еще в 1959 г. в работе (Hough, 1959).

Дальнейшее изложение структурировано следующим образом: приводится общая схема предлагаемого алгоритма с последующим описанием аппарата БПХ и каждого из этапов алгоритма, затем приводятся качественные и количественные результаты тестирования алгоритма на используемом тестовом наборе данных.

СХЕМА ДЕТЕКЦИИ ПРОТЯЖЕННЫХ ГРАНИЦ НА ИЗОБРАЖЕНИИ

Общая схема алгоритма приведена в виде псевдокода в Приложении (алгоритм 1). Алгоритму на вход подается изображение с нечетко выделенными границами: на рис. 2, *a* таким изображением является результат анализа формы структурного тензора в окрестности каждой точки изображения, а для *б* — результат применения морфологической операции “tophat” (вычитания из исходного изображения результата его размывания). Выходом является набор ломаных, задающих на изображении положение протяженных границ. Поиск ломаных осуществляется в три этапа: ищутся отрезки при помощи БПХ; полученные отрезки группируются в неупорядоченные множества с учетом ограничения ломаных на угол излома; в каждой из групп определяется порядок обхода, группа аппроксимируется ломаной и проходит проверку на протяженность.

Далее мы подробно рассмотрим каждый из этапов алгоритма. Однако прежде, чем перейти к их описанию, изложим некоторые детали использования аппарата БПХ.

БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ХАФА

Быстрое преобразование Хафа вычисляет суммы значений пикселей вдоль прямых, которые аппроксимируются специальным дискретным паттерном — диадическим (Ershov et al., 2015). Самоподобная структура диадического паттерна и рекурсивная схема суммиций позволяют уменьшить вычислительную сложность подсчета Хаф-образа до $O(n^2 \log(n))$, в то время как асимптотическая сложность стандартного ПХ для плотных изображений — $O(n^3)$, где n — сторона изображения. Более того, в 2018 г. в работе (Khanipov, 2018) было доказано, что данная вычислительная сложность является наименьшей достижимой.

При вычислении БПХ прямые на изображении задаются точками пересечения с противоположными краями изображения $(s, 0)$ и $(s + t, n)$, где s — целочисленный сдвиг паттерна, а t — целочисленный наклон. Такой прямой соответствует точка (s, t) в Хаф-образе, значение в которой равняется сумме значений пикселей вдоль соответствующего паттерна.

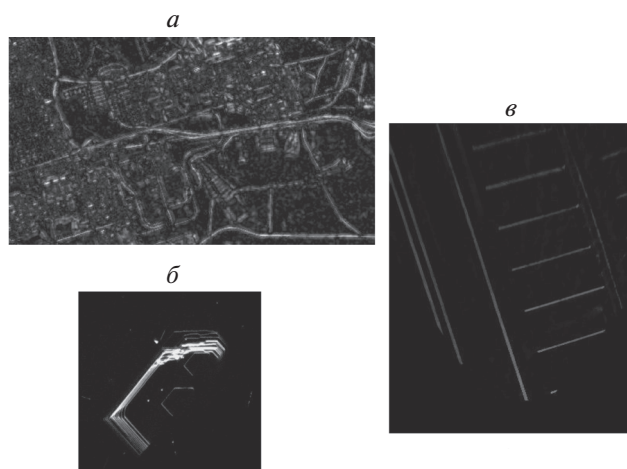


Рис. 2. Примеры изображений, подаваемых на вход детектору протяженных границ.

a — результат анализа формы структурного тензора в окрестности каждой точки изображения; *б* — результат контрастирования исходного изображения из (Zolotov et al., 2019); *в* — результат применения морфологической операции “tophat”.

Допустим, что на изображении есть прямая, параметризованная через s_0 и t_0 , значения пикселей вдоль которой велики, а по любой другой параллельной ей прямой — малы. Следовательно, сумма вдоль нее тоже будет велика, а суммы вдоль других параллельных прямых — малы. Тогда в строке t_0 Хаф-образа на профиле интенсивности будет наблюдаться резкий пик с вершиной в точке s_0 (рис. 3, *б*).

ПОИСК ОТРЕЗКОВ НА ИЗОБРАЖЕНИИ

Для поиска множества отрезков, покрывающих протяженные границы, изображение обрабатывается скользящим квадратным окном со стороной w . В каждом новом положении окна ищется одна превалирующая прямая при помощи БПХ. Сегмент, задаваемый граничными точками $(s, 0)$ и $(s + t, w)$ рассматриваемого окна, и будет искомым прямолинейным сегментом в пространстве исходного изображения.

Ввиду особенностей вычислений алгоритм Брейди вычисляет БПХ для прямых, угол наклона которых ограничен диапазоном в 45° (далее — квадрант). Ввиду того, что мы не накладываем ограничение на ориентацию отрезка в плоскости изображения, полученного Хаф-образа нам будет, очевидно, недостаточно. Для получения Хаф-образа для полного диапазона углов вычисляются четыре Хаф-образа, соответствующие следующим квадрантам: для горизонтальных линий с наклоном вправо в квадранте $[-45^\circ, 0^\circ]$ и влево в квадранте $[0^\circ, 45^\circ]$, для вертикальных ли-

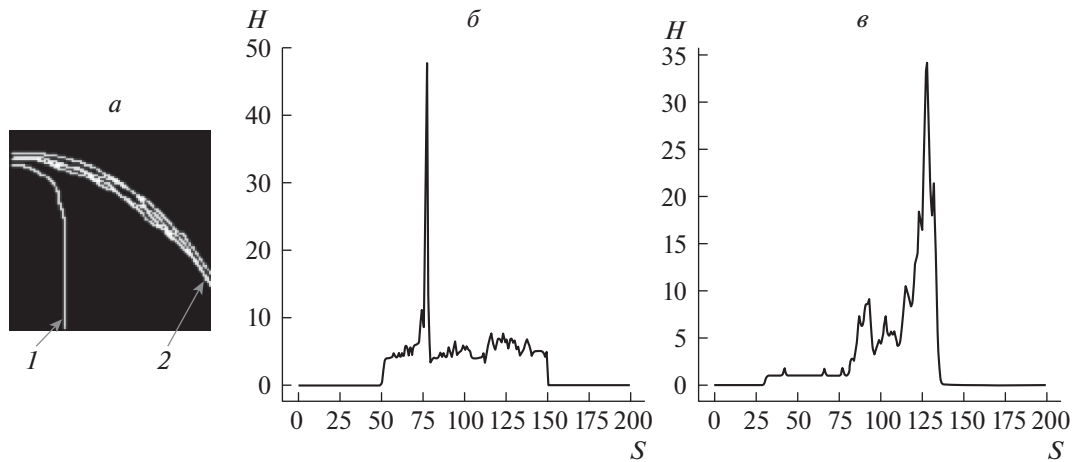


Рис. 3. *a* – исходное изображение с двумя линиями; *б* – результат суммирования по паттернам с наклоном, соответствующим направлению линии 1, дисперсия значений в строке равна 18,7, максимальное значение – 48; *в* – результат суммирования по паттернам с наклоном, соответствующим направлению линии 2, дисперсия – 32,04, максимальное значение – 34.

ний с наклоном вправо в квадранте $[45^\circ, 90^\circ]$ и влево в квадранте $[90^\circ, 135^\circ]$, которые затем объединяются в один (Aliev et al., 2019).

Как было сказано выше, суммирование вдоль прямой дает пик интенсивности в соответствующей строке Хаф-образа. Поэтому мы будем искать координаты пика в два последовательных этапа: сначала определимся со строкой, “пиковость” в которой максимальна, затем выберем максимальное значение в найденной строке.

Критерием для проверки строки на “пиковость” было решено выбрать дисперсию значений в строке. Это связано с тем, что в одном окне могут встретиться две линии с кардинально разным углом наклона. Например, такое характерно для снимков ДЗЗ, когда на изображении дорожной сети в сторону от главной трассы уходят боковые ответвления. В этом случае к преобладающей прямой должна приближаться главная трасса. Как правило, такая дорога отличается от соседних шириной и большим количеством образующих ее линий с близким углом наклона. На профиле интенсивности в соответствующей строке Хаф-образа это будет выражено в большем количестве пиков и их толщине. В этом случае дисперсия значений в строке, соответствующим направлению “главной” линии, будет больше, чем вдоль других линий (рис. 3).

Стоит заметить, что это не единственный возможный критерий. Так, например, в работе (Bezmaternykh et al., 2020) в качестве критерия берется сумма квадратов градиентов в строке.

Без потери общности далее будем говорить о поиске отрезков только в квадранте $[45^\circ, 90^\circ]$. Поиск в полном Хаф-образе аналогичен и отличается лишь обратным преобразованием точки

(s, t) в координаты сегмента на исходном изображении.

Общая схема поиска отрезков для одного квадранта приведена в Приложении (алгоритм 2).

Чтобы избавиться от размытых пиков и сохранить более резкие, можно дополнительно сгладить полученный Хаф-образ гауссовым фильтром вдоль оси t и вычесть из исходного Хаф-образа полученный, оставив только неотрицательные значения.

ГРУППИРОВКА ОТРЕЗКОВ

Целью группировки отрезков является их соотношение с той или иной границей. В основе группировки – попарный перебор отрезков. Пара отрезков (u, v) считаются принадлежащими одной границе, если выполняются два следующих условия:

1. Расстояние между отрезками меньше l_{max} ;
2. Угол $\angle(u, v)$ между отрезками лежит в диапазоне $[\phi_{min}, \phi_{max}]$.

В данной работе **расстояние между отрезками** определяется как расстояние между ближайшими на отрезках (u, v) точками, а **угол между отрезками** – как угол между двумя лучами, выходящими из точки пересечения отрезков к их наиболее удаленным концам (рис. 4, *a*). Для непересекающихся отрезков такой угол доопределяется параллельным переносом отрезков до совпадения двух ближайших концов в одной вершине (*б*).

Пусть каждая такая пара отрезков, которая удовлетворяет условиям 1–2, образует ребро $e \in E = \{(u, v) | u, v \in V\}$ в неориентированном гра-

фе $G = (V, E)$. Тогда группировку отрезков можно свести к поиску компонент связности в образованном графе G (Приложение, алгоритм 3).

АППРОКСИМАЦИЯ ГРУПП ОТРЕЗКОВ ЛОМАНЫМИ

Все границы на изображении будем разделять на разомкнутые прямолинейные и криволинейные границы и окружности. Последнее характерно для изображений дорожной разметки на круговом движении (рис. 1, в). Аппроксимация группы отрезков делится соответственно на аппроксимацию одиночным прямолинейным сегментом (ломаной с одним звеном) и аппроксимацию ломаной с более чем одним звеном (окружности тоже аппроксимируются ломаной, начало и конец которой совпадают). Для каждого результата аппроксимации вычисляется протяженность ломаной или сегмента как сумма расстояний между вершинами. Если вычисленная протяженность превосходит заданную, то результат подается на выход алгоритма (Приложение, алгоритм 4).

Для случая приближения группы отрезков одиночным прямолинейным сегментом ищется пара наиболее удаленных точек среди всех концов отрезков в группе. Эта пара точек задает координаты концов сегмента. Далее для всех остальных концов отрезков в группе проверяется их близость до образованного сегмента через расстояние от конкретной точки до построенного сегмента. Если доля близких точек велика, то линия разметки считается прямолинейной, а полученный сегмент – ее аппроксимацией. Общая схема описанных вычислений и их порядок приведены в Приложении, алгоритм 5.

Если же приблизить группу прямолинейным сегментом не удастся, то запускается аппроксимация группы отрезков ломаной (Приложение, алгоритм 6).

Для аппроксимации группы отрезков C ломаной необходимо сначала задать порядок обхода этих отрезков в соответствующей компоненте связности C графа G . Таким образом, при обходе отрезков для смежных отрезков будут выполняться условия 1–2, введенные в разделе “Группировка отрезков”. Однако этого все еще недостаточно для выстраивания ломаной, звенья которой также удовлетворяют условию 2.

Поясним на конкретном примере, приведенном на рис. 5, а. Допустим, на угол между парой отрезков наложены ограничения вида $\phi_{min} = 90^\circ$, $\phi_{max} = 180^\circ$, $l_{max} = 0$. Как мы видим из примера, выполнение ограничений для пар отрезков 1–2 и 1–3 не гарантирует соблюдения введенных ограничений при построении пути на отрезках 2–1–3 (такой путь обозначен пунктирной черной полилинией). Для контроля обхода таких отрезков бу-

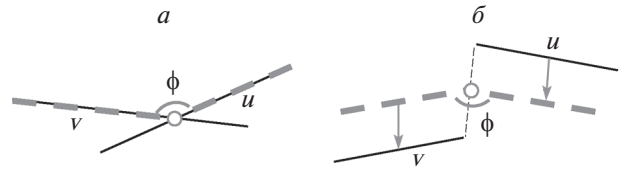


Рис. 4. Определение угла ϕ между отрезками u и v для случая: a – пересекающихся и b – непересекающихся отрезков.

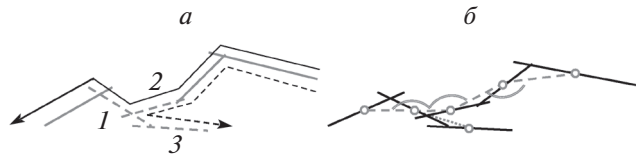


Рис. 5. Построение обхода отрезков в группе с учетом ограничений $\phi_{min} = 90^\circ$, $\phi_{max} = 180^\circ$, $l_{max} = 0$: a – пример корректного (черного сплошного) и некорректного (черного пунктирного) путей обхода: пары отрезков 1–3 и 1–2 удовлетворяют заданным ограничениям, построенный на них путь – нет; b – вычисление углов, образованных серединами трех соседних отрезков. Все углы, кроме построенных на отрезках 2–1–3, также удовлетворяют заданным ограничениям.

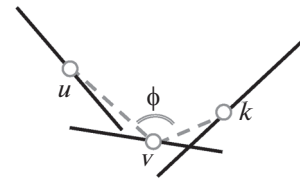


Рис. 6. Определение переходного угла ϕ между отрезками u, v, k .

дем дополнительно считать углы, образованные серединами троек отрезков (см. рис. 5(б)), которые мы будем называть переходными.

Переходным углом между отрезками u, v и k будем называть угол $\angle(u, v, k)$, образованный двумя лучами, выходящими из середины отрезка v к серединам отрезков u и k (рис. 6). Переходный угол также должен удовлетворять условию 2.

Для возможности обхода группы в случае отсутствия переходных углов в нужном диапазоне будем достраивать подграф C' следующим образом: если $\angle(u, v, k) \notin [\phi_{min}, \phi_{max}]$ и $(u, k) \notin E$, но $\angle(u, k) \in [\phi_{min}, \phi_{max}]$, то пробуем построить ребро (u, k) в подграфе C' . Для этого должно выполняться следующее требование: $\exists u' \in V, (u', u) \in E$ и $\angle(u', u, k) \in [\phi_{min}, \phi_{max}]$ или $(k, u') \in E$ и

$\angle(u, k, u') \in [\phi_{min}, \phi_{max}]$, то добавляем ребро (u, k) в подграф C' .

Для поиска порядка обхода всех отрезков будем использовать алгоритм Флойда. Для этого присвоим вес каждому ребру (u, v) , равный расстоянию между самыми удаленными концами отрезков u, v . Выходом алгоритма будет матрица кратчайших путей D между всеми парами вершин и матрица P , хранящая информацию о номерах фаз, на которых было получено кратчайшее расстояние между парами вершин. Далее из всех кратчайших путей выбирается путь, значение которого максимально. Если выбор максимального пути не однозначен (максимальные значения близки в пределах заданной погрешности), то выбирается максимальный путь с минимальным количеством пройденных вершин. Далее с использованием информации о фазах P восстанавливается обход вершин-отрезков.

Ввиду особенностей вычислений, описанный выше алгоритм ищет только разомкнутые прямолинейные и криволинейные границы. Результатом аппроксимации группы отрезков, покрывающих окружность, будет ломаная, соответствующая только половине окружности. Это связано с тем, что полный обход окружности будет иметь в полученной матрице D минимальное значение (так как кратчайшее расстояние между началом и концом будет равно нулю), а обход половины — максимальное. Поэтому из концов выбранного пути ищется альтернативный путь, используя предварительно подсчитанную матрицу Флойда. Если такой путь существует, и он не имеет общих вершин с предварительно найденным путем (за исключением начала и конца), то эти пути объединяются в один.

Далее, когда порядок обхода отрезков найден, строится ломаная, вершины которой — точки пересечения или ближайшие концы соседних в пути отрезков, если отрезки не пересекаются.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ КАЧЕСТВА РАБОТЫ ПРЕДЛАГАЕМОГО АЛГОРИТМА

Качество работы детектора протяженных границ оценивалось преимущественно на тестовом наборе данных, содержащем изображения дороги. Изображения были получены с фронтальной камеры БПТС при проезде по закрытой территории полигона “Калибр”. Дополнительно были проведены эксперименты на отдельных снимках ДЗЗ и топограмме дислокаций кристалла Si(111) (Zolotov et al., 2019) для проверки работоспособности алгоритма на разных моделях входных данных. На изображениях дороги протяженными границами задавались неширокие полосы дорож-

ной разметки, на снимках ДЗЗ — дороги, на топограмме — дислокации.

Примеры входных данных приведены на рис. 1, демонстрация пошаговой работы детектора для всех трех типов данных приведена на рис. 7: a — на изображении дороги; b — на снимке ДЗЗ; c — на топограмме дислокаций в кристалле Si(111). Для выделения дорожной разметки использовалась морфологическая операция “*tophat*” с квадратным окном, сторона которого бралась соразмерно с шириной полосы дорожной разметки, а для выделения дорог на данных ДЗЗ анализировалась форма структурного тензора в окрестности каждой точки изображения. Топограмма предварительной фильтрации не подвергалась.

Для получения численных результатов использовался только набор данных, собранный на полигоне “Калибр”. Истинные положения полос дорожной разметки были вручную отмечены в виде ломаных, а для сравнения ответов детектора и истинных ответов использовалась метрика из работы (Абрамов и др., 2019).

МЕТРИКА КАЧЕСТВА

Для численной оценки результата работы детектора были использованы точность (*precision*) и полнота (*recall*), для вычисления которых использовались количества ложноположительных (*FP*), ложноотрицательных (*FN*) и уникальных истинно положительных (*UTP*) ответов детектора. Общее количество истинно положительных (*TP*) срабатываний также подсчитывалось, однако из-за возможного множественного срабатывания детектора на отдельной полосе дорожной разметки для честного подсчета полноты и точности решено было использовать именно уникальные истинно положительные срабатывания.

Рассмотрим ломаную линию L , состоящую из N звеньев с координатами (x_i, y_i) , $i \in [0, N]$. Введем понятие региона точности (*Region of Precision, ROP*):

$$ROP_L(\omega) = \{(x, y) | (x, y) \in \Omega_i, i \in [0, N - 1]\},$$

где Ω_i — прямоугольник на изображении, заданный точками

$$\begin{aligned} &(x_i - \omega \cos \alpha_i, y_i + \omega \sin \alpha_i) \\ &(x_i + \omega \cos \alpha_i, y_i - \omega \sin \alpha_i) \\ &(x_{i+1} + \omega \cos \alpha_i, y_{i+1} - \omega \sin \alpha_i) \\ &(x_{i+1} - \omega \cos \alpha_i, y_{i+1} + \omega \sin \alpha_i), \end{aligned}$$

α_i — наклон i -го звена цепи L , образованного точками (x_i, y_i) , (x_{i+1}, y_{i+1}) , ω — параметр алгоритма, отвечающий за размер окрестности вокруг звена ($\omega = 14$ пикселей) (рис. 8).

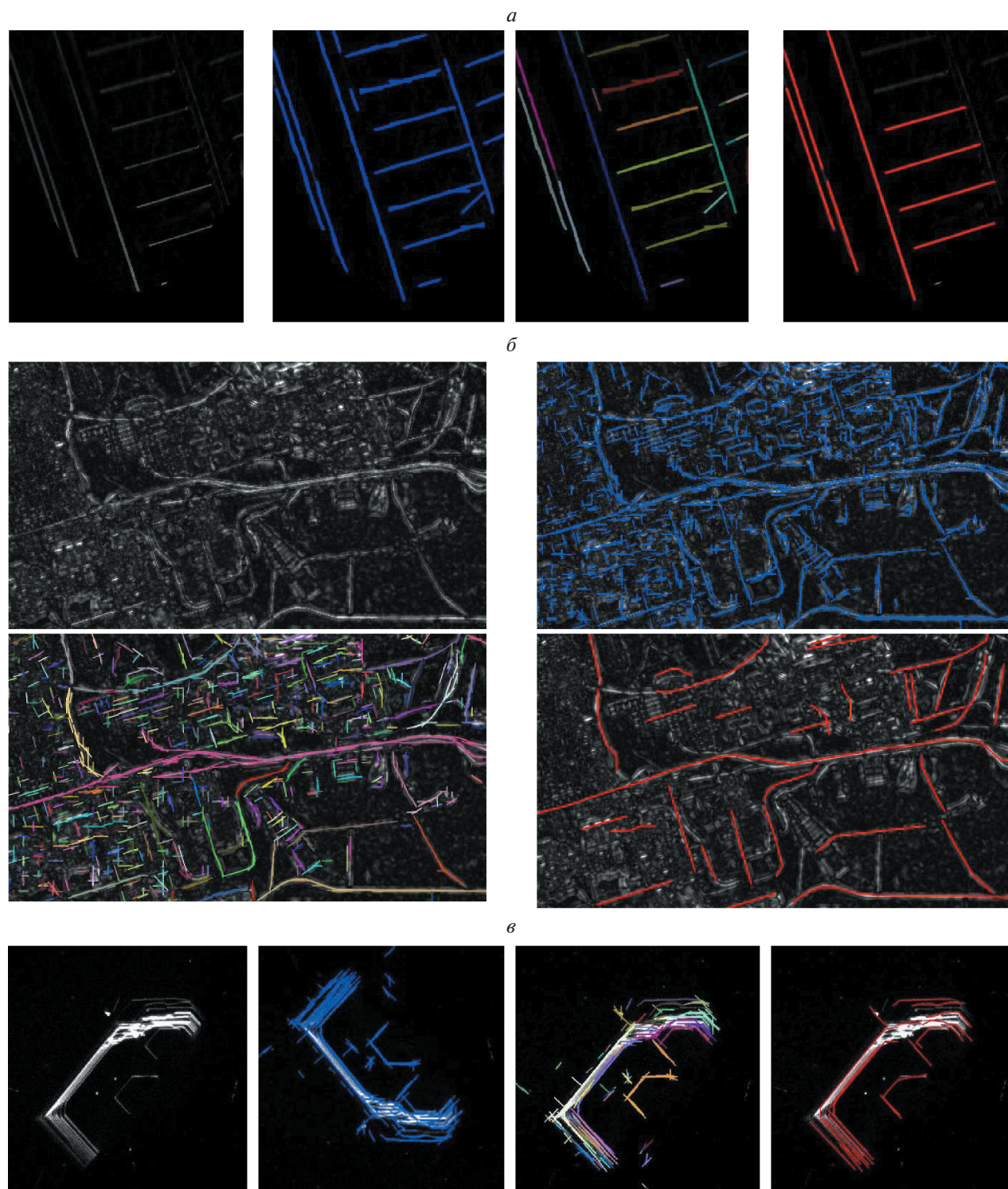


Рис. 7. Примеры пошаговой работы детектора протяженных границ:

a – на изображении дороги; *б* – на снимке ДЗЗ; *в* – на топограмме дислокаций в кристалле Si(111). Слева направо – входное изображение; результат поиска отрезков; результат группировки отрезков; результат аппроксимации групп отрезков ломаными.

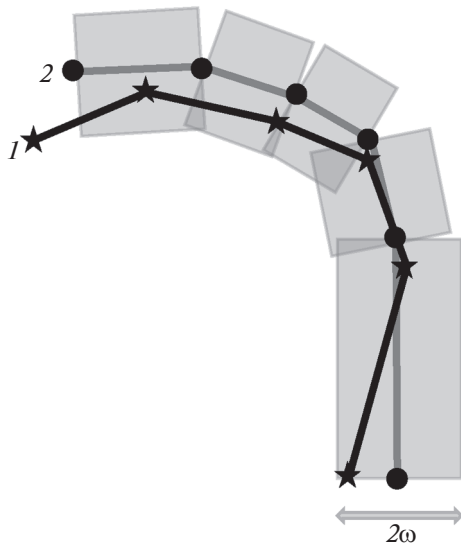


Рис. 8. Сопоставление найденной или истинной (1) ломаной с истинной или найденной (2) соответственно. Каждый прямоугольник шириной 2ω , одна из осей которого совпадает со звеном ломаной, отвечает за регион точности вокруг звена ломаной.

Тогда ответ детектора классифицируется как *TP*, *FP* или *FN*, если выполняется одно из условий 1–3:

1. $g_{e_i} = ROP_{L_{r_j}}(\omega) \cap L_{e_i}$ – насколько хорошо i -я найденная ломаная сопоставилась с j -й истинной. Если $g_{e_i} = 0 \forall j$, то ответ детектора – *FP*.
2. $g_{r_j} = ROP_{L_{e_i}}(\omega) \cap L_{r_j}$ – насколько хорошо j -я истинная ломаная r_j сопоставилась с i -й найден-

ной ломаной e_i . Если $g_{r_j} = 0 \forall i$, то ответ детектора – *FN*.

3. Иначе $g = \min(g_{r_j}, g_{e_i}) > 0$ и ответ детектора – *TP*.

Количество уникальных истинно положительных срабатываний считалось следующим образом:

$$UTP = \{g_{r_j} | g_{r_j} > 0\}.$$

ЧИСЛЕННЫЕ РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ АЛГОРИТМА

Набор изображений, снятых на полигоне “Калибр”, состоял из 1597 изображений дороги. Из них 1132 изображения содержали прямолинейную дорожную разметку, 437 – криволинейную, 28 кадров не содержали дорожной разметки вообще. Общее количество линий дорожной разметки – 5745.

Было найдено 10861 линий дорожной разметки, из них истинно положительных – 5308 (уникальных – 4182), ложноположительных – 5553. Полнота детектирования на тестовом наборе составила 73%, точность – 43%.

На рис. 9 приведены дополнительные примеры результата работы детектора. Как видно из приведенных данных, алгоритм успешно справился с поиском дорожной разметки различной кривизны, протяженности и ориентации, однако сработал на других светлых протяженных объектах в кадре (на светлом краю мусорного бака (а); белой скамейке (б); на протяженном блике (в)), что является скорее следствием свойств сцены и недостаточно точной разметкой, нежели недо-

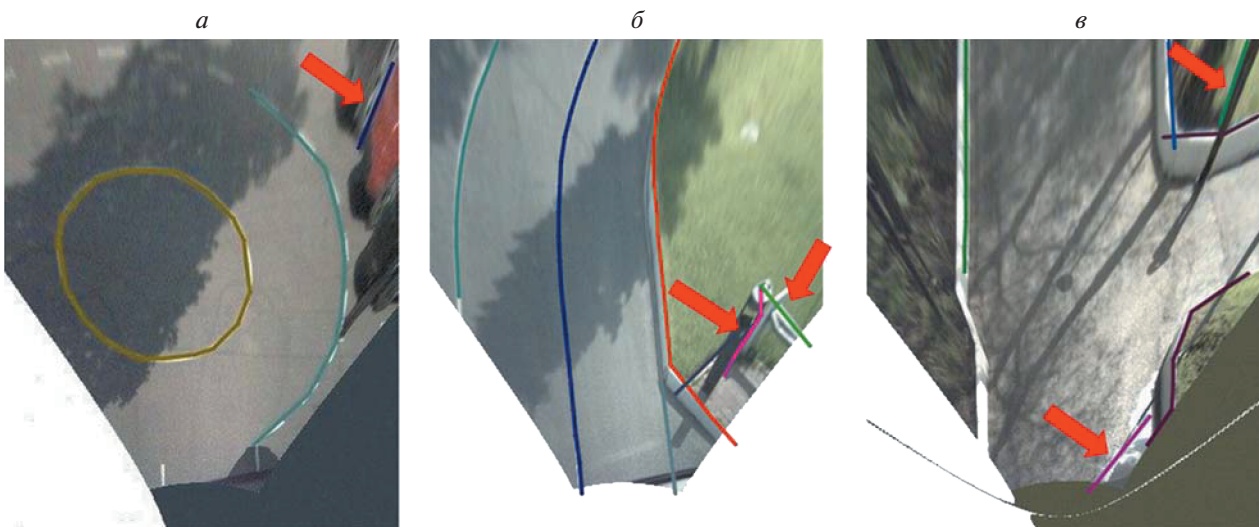


Рис. 9. Примеры работы детектора на отдельных изображениях дороги. Красная стрелка указывает на отдельное ложноположительное срабатывание.

статком самого детектора. В дальнейшем, например, можно отметить на изображениях из тестового набора зоны, содержащие светлые протяженные объекты, которые не являются линиями разметки, и не учитывать ложноположительные срабатывания в этих зонах при оценке качества работы детектора.

В работе (Shipitko et al., 2018) приведены результаты экспериментов, которые демонстрируют уменьшение средней ошибки локализации БПТС в 8 раз при использовании предлагаемого алгоритма.

ДАЛЬНЕЙШИЕ ИССЛЕДОВАНИЯ

Несмотря на то что приведенные выше результаты продемонстрировали перспективность предложенного алгоритма, к применяемому в практических целях алгоритму ставятся не только требования к точности, но и к скорости выполнения. Последнее напрямую связано с мощностными ресурсами используемого вычислительного устройства, которые в свою очередь имеют сильную корреляцию с габаритами вычислителя и габаритами роботизированной платформы (Kobov et al., 2019).

В будущем планируется оптимизировать поиск отрезков за счет однократного вычисления БПХ на области перекрытия соседних окон или исключить перекрытие окон вовсе, если это не скажется негативно на процедуре группировки отрезков. Кроме этого, представляет научный и практический интерес уточнить границы найденного прямолинейного сегмента в пределах ограничивающего его окна или найти его продолжение в соседнем окне с использованием быстрого вычисления сумм по произвольным отрезкам на изображении в так называемой БПХ-пирамиде (Soshin et al., 2020).

Также ускорение предложенного алгоритма может быть в дальнейшем достигнуто за счет уменьшения вычислительной сложности поиска порядка обхода отрезков в группе, например, за счет замены алгоритма Флойда на алгоритм Дейкстры для разреженных графов ввиду меньшей вычислительной сложности последнего.

ЗАКЛЮЧЕНИЕ

В данной работе предложен алгоритм поиска протяженных границ на изображении в виде ломанных с ограничением на угол излома. Алгоритм принимает на вход изображение с выделенными границами и возвращает набор ломанных, указывающих на положение границ.

При небольшом количестве основных параметров (угол между звеньями ломаной, минимальная протяженность ломаной) алгоритм способен детектировать границы разнообразной формы: прямолинейные, дугообразные, окруженные, множественные параллельные прямые. При этом алгоритму не подается на вход никаких сведений о предполагаемом месте прохождения границы на изображении и количестве границ, что делает его инвариантным к повороту или кропу изображения.

Благодаря предложенной схеме обхода множества отрезков из всех путей выбирается именно тот, который удовлетворяет заданным ограничениям на угол излома границы даже в условиях большого количества других отрезков в окрестности, например, в случае пересечения дорог на снимках ДЗЗ.

Полученные качественные и количественные результаты экспериментов свидетельствуют о практической ценности предлагаемого алгоритма.

Исследование выполнено при частичной финансовой поддержке РФФИ в рамках научных проектов № 17-29-03161 и № 18-29-26035.

ПРИЛОЖЕНИЕ

Алгоритм 1. ПОИСК_ПРОТЯЖЕННЫХ_ГРАНИЦ

Вход: изображение I с нечетко выделенными границами

Выход: множество ломанных L

- 1 $V \leftarrow$ ПОИСК_ОТРЕЗКОВ (I)
 - 2 $C \leftarrow$ ГРУППИРОВКА_ОТРЕЗКОВ (V)
 - 3 $L \leftarrow$ АППРОКСИМАЦИЯ_ГРУПП_ЛОМАНЫМИ (C)
-

Алгоритм 2. ПОИСК_ОТРЕЗКОВ

Вход: изображение I , w , $step$, Θ_{var} , Θ_s

Выход: множество отрезков V

- 1 $k = 0$
- 2 **Пока** $y <$ количество строк I **выполнить:**

Алгоритм 2. ПОИСК_ОТРЕЗКОВ

```

3   Пока  $x <$  количество столбцов  $I$  выполнить:
4       Задать окно  $I_w$  на  $I$  с левой верхней вершиной  $(x, y)$  и правой нижней  $(x + w, y + w)$ 
5        $H_w = \text{БПХ}(I_w)$ 
6        $i = 0$ 
7       Для всех  $i <$  количество строк  $H_w$  выполнить:
8           Подсчитать дисперсию  $T[i]$  в строке  $H_w[i]$ 
9            $i = i + 1$ 
10       $T_{var} = \text{max}(T)$ 
11       $t = \text{argmax}(T)$ 
12       $S_{max} = \text{max}(H_w[t])$ 
13       $s = \text{argmax}(H_w[t])$ 
14      Если  $T_{var} > \Theta_{var}$  и  $S_{max} > \Theta_s$ 
15           $p_b = (s, 0)$ 
16           $p_e = (s + t, w)$ 
17          Добавить в  $V$  отрезок  $[p_b + (x, y), p_e + (x, y)]$ 
18       $x = x + w$ 
19   $y = y + w$ 
20  Вернуть  $V$ 

```

Алгоритм 3. ГРУППИРОВКА_ОТРЕЗКОВ

```

      Ввод: множество отрезков  $V, l_{max}, [\phi_{min}, \phi_{max}]$ 
      Выход: множество групп отрезков  $C$ 
1  Для всех пар отрезков  $V[i], V[j] \in V$  выполнить:
2       $l =$  расстояние между  $V[i], V[j]$ 
3       $\phi =$  угол между  $V[i], V[j]$ 
4      Если  $l < l_{max}$  и  $\phi \in [\phi_{min}, \phi_{max}]$  то:
5           $G[i, j] = 1$ 
6      Иначе:
7           $G[i, j] = 0$ 
8   $C = \text{ПОИСК\_КОМПОНЕНТ\_СВЯЗНОСТИ}(G)$ 
9  Вернуть  $C$ 

```

Алгоритм 4. АППРОКСИМАЦИЯ_ГРУПП_ЛОМАНЫМИ

```

      Ввод: множество групп отрезков  $C, L_{min}$ 
      Выход: множество ломаных  $L$ 
1  Для всех групп отрезков  $C \in C$  выполнить:
2       $L = \text{АППРОКСИМАЦИЯ\_ПРЯМОЛИНЕЙНЫМ\_СЕГМЕНТОМ}(C)$ 
3      Если  $L = \emptyset$ , то:
4           $L = \text{АППРОКСИМАЦИЯ\_ЛОМАНОЙ}(C)$ 
5       $i = 0, l = 0$ 
6      Пока  $i <$  количество вершин в  $L$ :
7           $l_{cur} =$  расстояние между  $L[i]$  и  $L[i + 1]$ 
8           $l = l + l_{cur}$ 

```

Алгоритм 4. АППРОКСИМАЦИЯ_ГРУПП_ЛОМАНЫМИ

- 9 Если $l > L_{min}$, то:
- 10 Добавить L в L
- 11 Вернуть L

Алгоритм 5. АППРОКСИМАЦИЯ_ПРЯМОЛИНЕЙНЫМ_СЕКМЕНТОМ

Вход: группа отрезков C , R_{min} , \mathcal{D}_{max}

Выход: сегмент $[p_b, p_e]$

- 1 $d_{max} = 0$; $p_b = \emptyset$, $p_e = \emptyset$
- 2 Для всех $p[i], p[j] \in \{\text{концы отрезков из } C\}$, выполнить:
- 3 $d = \text{расстояние между } p[i] \text{ и } p[j]$
- 4 Если $d > d_{max}$, то:
- 5 $d = d_{max}$
- 6 $p_b = p[i]$
- 7 $p_e = p[j]$
- 8 $N = M = 0$
- 9 Для всех $p[i] \in \{\text{концы отрезков из } C\}$, выполнить:
- 10 $d = \text{расстояние от } p[i] \text{ до сегмента } (p_b, p_e)$
- 11 Если $d < \mathcal{D}_{max}$, то:
- 12 $N = N + 1$
- 13 $M = M + 1$
- 14 Если $\frac{N}{M} > R_{min}$, то:
- 15 Вернуть (p_b, p_e)
- 16 Иначе:
- 17 Вернуть \emptyset

Алгоритм 6. АППРОКСИМАЦИЯ_ЛОМАНОЙ

Вход: компонента связности C , V , ϕ_{min} , ϕ_{max}

Выход: ломаная L

- 1 Для всех $C[i, j] = 1$ и $V[i], V[j] \in C$ выполнить:
- 2 $C[i, j] = \text{расстояние между самыми удаленными концами отрезков } V[i], V[j]$
- 3 Для всех $V[i], V[j], V[k] \in C$ выполнить:
- 4 $\varphi = \angle(V[i], V[j], V[k])$
- 5 $\phi = \angle(V[i], V[k])$
- 6 Если $\varphi \notin [\phi_{min}, \phi_{max}]$ и $C[i, k] = 0$ и $\phi \in [\phi_{min}, \phi_{max}]$, то:
- 7 Для всех $V[l] \in C$, $l \neq i, j, k$ выполнить:
- 8 Если $(C[l, i] > 0$ и $\angle(V[l], V[i], V[k]) \in [\phi_{min}, \phi_{max}])$ или
- 9 $(C[k, l] \text{ и } \angle(V[i], V[k], V[l]) \in [\phi_{min}, \phi_{max}])$ то:
- 10 $C[i, k] = \text{расстояние между самыми удаленными концами отрезков } V[i], V[k]$
- 11 матрица кратчайших расстояний D , матрица номеров фаз $P = FLOID(C)$
- 12 $d_{max} = \max(D)$
- 13 $i, j = \text{argmax}(D)$
- 14 $L1 = \text{ВОССТАНОВИТЬ_ПУТЬ}(i, j, D, P, V)$
- 15 Для всех u, v , таких что $C[u, i] > 0$ и $C[v, j] > 0$ и $V[u], V[v] \in C$:

Алгоритм 6. АППРОКСИМАЦИЯ_ЛОМАНОЙ

- 16 $L2 = \text{ВОССТАНОВИТЬ_ПУТЬ}(u, v, D, P, V)$
 17 **Если** $L1 \cap L2 = \emptyset$, **то**:
 18 $L1 = L1 \cup L2$ и **прервать поиск другого пути**
 19 **Добавить** в L самый удаленный конец отрезка $L1[0]$ до $L1[1]$
 20 $i = 0$
 21 **Пока** $i <$ количество вершин в $L1$:
 22 **Добавить** в L пересечение $L1[i]$ и $L1[i + 1]$ или их ближайшие концы
 23 $i = i + 1$
 24 **Добавить** в L самый удаленный конец отрезка $L1[i]$ до $L1[i - 1]$
 25 **Вернуть** L

СПИСОК ЛИТЕРАТУРЫ

- Абрамов М.П., Шипитко О.С., Лукоянов А.С., Панфилова Е.И., Кунина И.А., Григорьев А.С. Система позиционирования внутри зданий мобильной робототехнической платформы на основе детекции краев. *Сенсорные системы*. 2019. Т. 33. № 1. С. 30–43. <https://doi.org/10.1134/S0235009219010025>
- Котов А.А., Коноваленко И.А., Николаев Д.П. Прслеживание объектов в видеопотоке, оптимизированное с помощью быстрого преобразования Хафа. *ИТuBC*. 2015. № 1. С. 56–68.
- Тропин Д.В., Шемякина Ю.А., Коноваленко И.А., Фараджев И.А. О локализации плоских объектов на изображениях со сложной структурой проективных искажений. *Информационные процессы*. 2019. Т. 19. № 2. С. 208–229.
- Aliev M., Ershov E.I., Nikolaev D.P. On the use of FHT, its modification for practical applications and the structure of Hough image. *ICMV 2018*. 2019. V. 11041. <https://doi.org/10.1117/12.2522803>
- Anil P.N., Natarajan S. A novel approach using active contour model for semi-automatic road extraction from high resolution satellite imagery. 2010 Second International Conference on Machine Learning and Computing. 2010. P. 263–266.
- Bezmaternykh P.V., Nikolaev D.P. A document skew detection method using fast Hough transform. *ICMV 2019*. 2020. V. 11433. P. 1–6. <https://doi.org/10.1117/12.2559069>
- Brady M.L., Yong W. Fast parallel discrete approximation algorithms for the Radon transform. Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures. 1992. P. 91–99.
- Bulatov K.B., Chukalina M.V., Nikolaev D.P. Fast x-ray sum calculation algorithm for computed tomography. *Bulletin SUSU MMCS*. 2020. V. 13. № 1. P. 95–106. <https://doi.org/10.14529/mmp200107>
- Canny J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*. 1986. № 6. P. 679–698.
- Ding Y., Xu Z., Zhang Y., Sun K. Fast lane detection based on bird's eye view and improved random sample consensus algorithm. *Multimedia Tools and Applications*. 2017. V. 76. № 21. P. 22979–22998.
- Ershov E., Terekhin A., Nikolaev D., Postnikov V., Karpenko S. Fast Hough transform analysis: pattern deviation from line segment. *ICMV 2015*. 2015. V. 9875. P. 1–5. <https://doi.org/10.1117/12.2228852>
- Fan S., Shui P., Zhang Z., Sun Y. SAR image edge detection via directional Bhattacharyya coefficient with its application on image segmentation. Eleventh International Conference on Machine Vision (ICMV 2018). 2019. V. 11041. P. 1104107.
- Hough P.V.C. Machine analysis of bubble chamber pictures. *Conf. Proc.* 1959. T. 590914. P. 554–558.
- Hough P.V.C. Method and means for recognizing complex patterns. Patent USA. No 3069654. 1962.
- Khanipov T.M. Computational complexity lower bounds of certain discrete Radon transform approximations. arXiv preprint arXiv:1801.01054. 2018.
- Kirthika A., Mookambiga A. Automated road network extraction using artificial neural network. 2011 International Conference on Recent Trends in Information Technology (ICRTIT). 2011. P. 1061–1065.
- Korobov N., Shipitko O., Konovalenko I., Grigoryev A., Chukalina M. SWaP-C based comparison of onboard computers for unmanned vehicles. Proceedings of 14th International Conference on Electromechanics and Robotics “Zavalishin’s Readings”. 2019. Singapore. Springer. V. 154. P. 573–583. https://doi.org/10.1007/978-981-13-9267-2_47
- Kunina I., Panfilova E., Gladkov A. Matching of SAR and optical images by independent referencing to vector map. Eleventh International Conference on Machine Vision (ICMV 2018). 2019. V. 11041. P. 1104102.
- Li Y., Chen L., Huang H., Li X., Xu W., Zheng L., Huang J. Nighttime lane markings recognition based on Canny detection and Hough transform. 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR). 2016. P. 411–415.
- Limonova E.E., Bezmaternykh P.V., Nikolaev D.P., Arlazarov V.V. Slant Rectification in Russian Passport OCR System Using Fast Hough Transform. *ICMV 2016*. SPIE, July 2017. 2017. V. 10341. ISSN 0277-786X. ISBN 978-15-10611-31-3. P. 1–5. <https://doi.org/10.1117/12.2268725>
- Limonova E.E., Tropin D.V., Savelev B.I., Mamay I.B., Nikolaev D.P. Mobile and Embedded Fast High Resolution Image Stitching for Long Length Rectangular Monochromatic Objects with Periodic Structure. IC-

- MV 2017. SPIE, Apr. 2018. 2018. V. 10696. ISBN 978-15-10619-41-8. P. 1–8.
<https://doi.org/10.1117/12.2310093>
- Lopez A., Canero C., Serrat J., Saludes J., Lumbreras F., Graf T. Detection of Lane Markings based on Ridgeness and RANSAC. Proceedings 2005 IEEE Intelligent Transportation Systems. 2005. P. 254–259.
- Ma R., Wang W., Liu S. Extracting roads based on Retinex and improved Canny operator with shape criteria in vague and unevenly illuminated aerial images. Journal of applied remote sensing. 2012. V. 6. № 1. С. 063610.
- Mousavi S.M.H., Lyashenko V., Prasath S. Analysis of a robust edge detection system in different color spaces using color and depth images. Компьютерная оптика. 2019. V. 43. № 4.
- Mukhopadhyay P., Chaudhuri B.B. A survey of Hough Transform. Pattern Recognition. 2015. V. 48. № 3. P. 993–1010.
- Radon J. Über die Bestimmung von Funktionen durch ihre Integralwerte langs gewisser Mannigfaltigkeiten. Berichte Sachsische Academie der Wissenschaften, Leipzig, Math.-Phys. Kl. 1917. V. 69. P. 262–267.
- Shipitko O., Grigoryev A. Ground Vehicle Localization With Particle Filter Based On Simulated Road Marking Image. ECMS 2018. 2018.
<https://doi.org/10.7148/2018-0341>
- Soshin K.V., Nikolaev D.P., Gladilin S.A., Ershov E.I. Acceleration of Summation Over Segments Using the Fast Hough Transformation Pyramid. Bulletin of the South Ural State University. Ser. Mathematical Modelling, Programming & Computer Software. 2020. V. 13. № 1. P. 129–140.
<https://doi.org/10.14529/mmp200110>
- Sheshkus A., Ingacheva A., Arlazarov V., Nikolaev D. HoughNet: neural network architecture for vanishing points detection. ICDAR 2019. IEEE Feb. 2020. ISSN 1520-5363. ISBN 978-17-28130-15-6. P. 844–849.
<https://doi.org/10.1109/ICDAR.2019.00140>
- Wang Y., Teoh E.K., Shen D. Lane detection and tracking using B-Snake. Image and Vision computing. 2004. V. 22. № 4. P. 269–280.
- Xiao L., Li C., Zhao D., Chen T., Dai B. Road marking detection based on structured learning. 12th World Congress on Intelligent Control and Automation (WCICA). 2016. P. 2047–2051.
- Zolotov D.A., Asadchikov V.E., Buzmakov A.V., D'yachkova I.G., Krivonosov Y.S., Chukhovskii F.N., Suvorov E.V. X-ray Diffraction Tomography Using Laboratory Sources for Studying Single Dislocations in a Low Absorbing Silicon Single Crystal. Optoelectronics, Instrumentation and Data Processing. 2019. V. 55. № 2. P. 126–132.

Using window hough transform for detecting elongated boundaries in an image

E. I. Panfilova^{a,b,c,#} and I. A. Kunina^{a,b,d}

^a The Institute for Information Transmission Problems of Russian Academy of Sciences
 127051 Moscow, Bolshoi Karetny Lane, 19, Russia

^b Smart Engines Service LLC 117312 Moscow, Prospect 60-Letiya Oktyabrya, 9, Russia

^c V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences
 117997 Moscow, Profsoyusnaya street, 65, Russia

^d Moscow Institute of Physics and Technology 141701 Dolgoprudny, Institutsky Lane, 9, Russia

[#]E-mail: e.panfilova@smartengines.com

This work presents the algorithm for detecting elongated boundaries in an image. The case where boundary can be approximated by polyline with restricted maximum curvature angle is considered. The task of detecting such boundaries arises, for instance, during the detection of road markings, construction of a road map by analyzing Earth satellite images, and detection of crystal dislocations in a single X-ray topo-tomography projection. To find polyline segments an image is processed by a sliding window and for each window position, a straight line is detected by calculating Fast Hough transform (FHT). Further, detected segments are grouped by relative position in the image. Segments groups, covering desired boundaries, are approximated by polylines. The proposed algorithm was used in the problem of road marking detection to determine the ego-position of an unmanned vehicle on a given vector road map. The algorithm was tested on real data collected from the front-looking camera of the unmanned vehicle driving at the experimental area “Kalibr” (Moscow). The precision of road markings lines detector was evaluated as 43%, and the recall as 73%. The mean absolute positioning error in an experimental run with road marking detection was 0.2 m (Euclidean distance), which is 8 times less than localization error without road marking detection. The algorithm was also tested on remote sensing images and topo-tomograms.

Key words: boundary detection, lane detection, elongated dislocations, fast Hough transform, window image analysis

REFERENCES

- Abramov M.P., Shipitko O.S., Lukoyanov A.S., Panfilova E.I., Kunina I.A., Grigorev A.S. Sistema pozicionirovaniy vnutri zdaniy mobilnoi robototekhnicheskoi platformu na osnove detekcii kraev [Edge detection based mobile robot indoor localization.] Sensornye systemy [Sensory systems]. 2019. V. 33. № 1. P. 30–43 (in Russian).
<https://doi.org/10.1134/S0235009219010025>
- Kotov A.A., Konovalenko I.A., Nikolaev D.P. Proslzhivanie obektov v videopotoke, optimizirovannoe s pomoschyu bystrogo preobrazovaniya KHafa [Tracking of objects containing multiple concentric arcs in a video stream, optimized with Fast Hough transform] ITiVS. 2015. № 1. P. 56–68 (in Russian).
- Tropin D.V., Shemyakina Y.A., Konovalenko I.A., Faradzhev I.A. O lokalizacii ploskikh obektov na izobra-

- zeniych so sloznoi strukturoi proektivnuch iskazeni [Localization of planar objects on the images with complex structure of projective distortion]. *Informatsionnye protsessy*. 2019. V. 19. № 2. P. 208–229 (in Russian).
- Aliev M., Ershov E.I., Nikolaev D.P. On the use of FHT, its modification for practical applications and the structure of Hough image. *ICMV 2018*. 2019. V. 11041. <https://doi.org/10.1117/12.2522803>.
- Anil P.N., Natarajan S. A novel approach using active contour model for semi-automatic road extraction from high resolution satellite imagery. 2010 Second International Conference on Machine Learning and Computing. 2010. P. 263–266.
- Bezmaternykh P.V., Nikolaev D.P. A document skew detection method using fast Hough transform. *ICMV 2019*. 2020. V. 11433. P. 1–6. <https://doi.org/10.1117/12.2559069>
- Brady M.L., Yong W. Fast parallel discrete approximation algorithms for the Radon transform. *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures*. 1992. P. 91–99.
- Bulatov K.B., Chukalina M.V., Nikolaev D.P. Fast x-ray sum calculation algorithm for computed tomography. *Bulletin SUSU MMCS*. 2020. V. 13. № 1. P. 95–106. <https://doi.org/10.14529/mmp200107>
- Canny J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*. 1986. № 6. P. 679–698.
- Ding Y., Xu Z., Zhang Y., Sun K. Fast lane detection based on bird's eye view and improved random sample consensus algorithm. *Multimedia Tools and Applications*. 2017. V. 76. № 21. P. 22979–22998.
- Ershov E., Terekhin A., Nikolaev D., Postnikov V., Karp-en-ko S. Fast Hough transform analysis: pattern deviation from line segment. *ICMV 2015*. 2015. V. 9875. P. 1–5. <https://doi.org/10.1117/12.2228852>
- Fan S., Shui P., Zhang Z., Sun Y. SAR image edge detection via directional Bhattacharyya coefficient with its application on image segmentation. *Eleventh International Conference on Machine Vision (ICMV 2018)*. 2019. V. 11041. P. 1104107.
- Hough P.V.C. Machine analysis of bubble chamber pictures. *Conf. Proc.* 1959. T. 590914. P. 554–558.
- Hough P.V.C. Method and means for recognizing complex patterns. *Patent USA*. No 3069654. 1962.
- Khanipov T.M. Computational complexity lower bounds of certain discrete Radon transform approximations. *arXiv preprint arXiv:1801.01054*. 2018.
- Kirthika A., Mookambiga A. Automated road network extraction using artificial neural network. *International Conference on Recent Trends in Information Technology (ICRTIT)*. 2011. P. 1061–1065.
- Korobov N., Shipitko O., Konovalenko I., Grigoryev A. and Chukalina M. SWaP-C based comparison of on-board computers for unmanned vehicles. *Proceedings of 14th International Conference on Electromechanics and Robotics “Zavalishin’s Readings”*. 2019. Singapore. Springer. V. 154. P. 573–583. https://doi.org/10.1007/978-981-13-9267-2_47
- Kunina I., Panfilova E., Gladkov A. Matching of SAR and optical images by independent referencing to vector map. *Eleventh International Conference on Machine Vision (ICMV 2018)*. 2019. V. 11041. P. 1104102.
- Li Y., Chen L., Huang H., Li X., Xu W., Zheng L., Huang J. Nighttime lane markings recognition based on Canny detection and Hough transform. 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR). 2016. P. 411–415.
- Limonova E.E., Bezmaternykh P.V., Nikolaev D.P. and Arlazarov V.V. Slant Rectification in Russian Passport OCR System Using Fast Hough Transform. *ICMV 2016*. SPIE, July 2017. 2017. V. 10341. ISSN 0277-786X. ISBN 978-15-10611-31-3. P. 1–5. <https://doi.org/10.1117/12.2268725>
- Limonova E.E., Tropin D.V., Savelev B.I., Mamay I.B. and Nikolaev D.P. Mobile and Embedded Fast High Resolution Image Stitching for Long Length Rectangular Monochromatic Objects with Periodic Structure. *ICMV 2017*. SPIE, Apr. 2018. 2018. V. 10696. ISBN 978-15-10619-41-8. P. 1–8. <https://doi.org/10.1117/12.2310093>
- Lopez A., Canero C., Serrat J., Saludes J., Lumbreras F., Graf T. Detection of Lane Markings based on Ridge-ness and RANSAC. *Proceedings 2005 IEEE Intelligent Transportation Systems*. 2005. P. 254–259.
- Ma R., Wang W., Liu S. Extracting roads based on Retinex and improved Canny operator with shape criteria in vague and unevenly illuminated aerial images. *Journal of applied remote sensing*. 2012. V. 6. № 1. C. 063610.
- Mousavi S.M.H., Lyashenko V., Prasath S. Analysis of a robust edge detection system in different color spaces using color and depth images. *Компьютерная оптика*. 2019. V. 43. № 4.
- Mukhopadhyay P., Chaudhuri B.B. A survey of Hough Transform. *Pattern Recognition*. 2015. V. 48. № 3. P. 993–1010.
- Radon J. Uber die Bestimmung von Funktionen durch ihre Integralwerte langs gewisser Mannigfaltigkeiten. *Berichte Sachsische Academie der Wissenschaften, Leipzig, Math.-Phys. Kl.* 1917. V. 69. P. 262–267.
- Shipitko O., Grigoryev A. Ground vehicle localization with particle filter based on simulated road marking image. *ECMS 2018*. 2018. <https://doi.org/10.7148/2018-0341>
- Soshin K.V., Nikolaev D.P., Gladilin S.A., Ershov E.I. Acceleration of Summation Over Segments Using the Fast Hough Transformation Pyramid. *Bulletin of the South Ural State University. Ser. Mathematical Modelling, Programming & Computer Software*. 2020. V. 13. № 1. P. 129–140. <https://doi.org/10.14529/mmp200110>
- Sheshkus A., Ingacheva A., Arlazarov V., Nikolaev D. HoughNet: neural network architecture for vanishing points detection. *ICDAR 2019*. IEEE Feb. 2020. ISSN 1520-5363. ISBN 978-17-28130-15-6. P. 844–849. <https://doi.org/10.1109/ICDAR.2019.00140>
- Wang Y., Teoh E.K., Shen D. Lane detection and tracking using B-Snake. *Image and Vision computing*. 2004. V. 22. № 4. P. 269–280.
- Xiao L., Li C., Zhao D., Chen T., Dai B. Road marking detection based on structured learning. *12th World Congress on Intelligent Control and Automation (WCICA)*. 2016. P. 2047–2051.
- Zolotov D.A., Asadchikov V.E., Buzmakov A.V., D'yachkova I.G., Krivonosov Y.S., Chukhovskii F.N., Suvorov E.V. X-ray diffraction tomography using laboratory sources for studying single dislocations in a low absorbing silicon single crystal. *Optoelectronics, Instrumentation and Data Processing*. 2019. V. 55. № 2. P. 126–132.