

УДК 004.93

Д. М. Талапина, О. А. Новосельская

Белорусский государственный технологический университет

**МОДИФИКАЦИЯ АЛГОРИТМА ХАФА ДЛЯ РАСПОЗНАВАНИЯ
И КЛАССИФИКАЦИИ ШТРИХОВЫХ ИЗОБРАЖЕНИЙ**

В статье рассмотрен вопрос распознавания сложных для анализа неоднородных линий. Для выбора традиционного алгоритма с целью извлечения признаков проведен анализ существующих алгоритмов. Поскольку штриховое изображение будет состоять из геометрических фигур в виде неоднородных штрихов разной толщины, за основу взят алгоритм Хафа, являющийся традиционным алгоритмом компьютерного зрения. Известно, что алгоритм Хафа имеет ряд недостатков. В частности, существуют проблемы распознавания прерывистых линий, а также линий ограниченной длины. В статье показан процесс разработки алгоритма распознавания штриховых изображений, нацеленного на анализ, интерпретацию и классификацию штриховых изображений и графических примитивов, представленных в виде неоднородных линий. Рассмотрены особенности представления штриховых изображений, обобщены требования, предъявляемые к алгоритму распознавания. Разработанный алгоритм основывается на модифицированном алгоритме Хафа, адаптированном для распознавания штриховых, сплошных и штрихпунктирных линий. В статье описаны основные этапы алгоритма, включая предварительную обработку изображений, определение и классификацию графических примитивов, а также методы постобработки. Исходя из проведенного анализа в качестве алгоритмов предварительной обработки целесообразно использовать комбинацию фильтров со сверткой (фильтр Гаусса и фильтр Кани). Принято решение о необходимости использования морфологической операции дилатации для расширения светлых пространств на изображении. Показаны особенности модифицированного алгоритма, предложенного авторами. Осуществлен расчет параметров линий с использованием Евклидова расстояния. Проведена оценка эффективности разработанного алгоритма. Ключевой особенностью предложенного метода является возможность вывода параметров штриховых изображений в отдельный файл.

Ключевые слова: распознавание изображений; штриховое изображение; графические примитивы; алгоритм Хафа; предварительная обработка изображений; постобработка изображений, Евклидово расстояние.

Для цитирования: Талапина Д. М., Новосельская О. А. Модификация алгоритма Хафа для распознавания и классификации штриховых изображений // Труды БГТУ. Сер. 3, Физико-математические науки и информатика. 2024. № 2 (284). С. 89–95.

DOI: 10.52065/2520-6141-2024-284-12.

D. M. Talapina, O. A. Novoselskaya

Belarusian State Technological University

**MODIFICATION OF THE HOUGH ALGORITHM FOR RECOGNITION
AND CLASSIFICATION OF LINE IMAGES**

The article considers the question of recognition the complex for analysis non-uniform lines. To select a traditional algorithm for feature extraction, an analysis of existing algorithms is performed. Since the line image will consist of geometric figures in the form of non-uniform strokes of different thickness, the Hough algorithm, which is a traditional algorithm of computer vision, is taken as a basis. It is known that the Hough algorithm has a number of disadvantages. In particular, there are problems of recognition of dashed lines, as well as lines of limited length. The article shows the process of developing an algorithm for recognition of line images aimed at analysis, interpretation and classification of line images and graphic primitives presented in the form of non-uniform lines. The features of representation of line images are considered, the requirements for the recognition algorithm are summarized. The developed algorithm is based on a modified Hough algorithm adapted for recognition of dashed, solid and dash-dotted lines. The article describes the main stages of the algorithm, including image preprocessing, definition and classification of graphic primitives, and postprocessing methods. Based on the analysis, it is advisable to use a combination of convolution filters (Gaussian filter and Canny filter) as preprocessing algorithms. It was decided to use a morphological dilation operation to expand light spaces in the image. The features of the modified algorithm proposed by the authors are shown. Calculation of line parameters using Euclidean distance is carried out. Effectiveness of the developed algorithm is assessed. The key feature of the proposed method is the ability to output the parameters of dashed images in a separate file.

Keywords: image recognition; line art; graphic primitives; Hough algorithm; image pre-processing; image post-processing, Euclidean distance.

For citation: Talapina D. M., Novoselskaya O. A. Modification of the Hough algorithm for recognition and classification of line images. *Proceedings of BSTU, issue 3, Physics and Mathematics. Informatics*, 2024, no. 2 (284), pp. 89–95 (In Russian). DOI: 10.52065/2520-6141-2024-284-12.

Введение. Разработка алгоритма распознавания штрихового изображения ставит своей целью создание эффективного инструмента, способного анализировать, интерпретировать и классифицировать графические примитивы в штриховых изображениях.

На сегодняшний день существует множество методов распознавания штриховых изображений, но не все они применимы для задачи распознавания изображений на физических носителях. Это связано с тем, что изображения могут варьироваться по своим геометрическим и яркостным характеристикам, а также подвергаться нетривиальным деформациям носителя, что затрудняет их классификацию [1].

Целью данной работы является разработка алгоритма распознавания, позволяющего проводить анализ штриховых изображений, состоящих из графических примитивов. Проводимый анализ должен решать следующие задачи: нахождение и классификация графических примитивов на изображении, сбор информации о характеристиках графических примитивов (ширина, длина, тип линии и т. д.), а также запись полученных данных в файл.

Под графическим примитивом понимается простейший геометрический объект, отображаемый на экране дисплея. Описание графического примитива обычно содержит метрическую и атрибутивную части. Метрическая часть позволяет сопоставить те величины, в которых задан графический примитив для отображения его на дисплее, и те величины, которые характеризуют его физическое или логическое представление. Атрибутивная часть передает геометрические параметры, характеризующие форму и расположение графического примитива [2].

Точка – наиболее простой графический примитив нулевой размерности. Точка определяется своими координатами на плоскости. Линия – это совокупность точек, через которые проходит геометрический отрезок с заданными конечными точками; характеризуется начальной и конечной точками, или начальной точкой и приращениями координат, либо длиной и углом наклона. В зависимости от стиля различают: сплошная линия – непрерывная линия, которая соединяет две точки без разрывов; штриховая линия – линия, состоящая из коротких отрезков или точек с промежутками между ними. В настоящей работе требуется разработать алгоритм распознавания как

сплошных, так и штриховых линий с определением их толщины.

Исходя из определения, толщина линии должна быть равна 1 пикселю, чтобы такой примитив считался непосредственно линией, в противном случае данный примитив относится к классу примитивов прямоугольник. Так как данное ограничение существенно снижает потенциал использования штриховых изображений в качестве элемента защиты данных, была поставлена задача распознавания, классификации и сбора информации о характеристиках линий толщиной более одного пикселя, т. е. распознавания прямоугольника с заливкой для сплошных линий толщиной более 1 пикселя, и распознавания совокупности прямоугольников с заливкой для штриховых линий толщиной более 1 пикселя.

Для решения задачи распознавания примитивов на изображении может быть применен алгоритм Хафа [3]. Алгоритм Хафа – это популярный метод обнаружения простых геометрических форм, таких как прямые, окружности, эллипсы и т. д., на изображениях [4]. Классический алгоритм Хафа не решает поставленные в данной работе задачи, поскольку имеет строгое ограничение толщины линии в 1 пиксель, не распознает штриховые линии как цельный объект, а также не определяет характеристики найденных примитивов (за исключением начальных и конечных координат), что затрудняет применение классического алгоритма для классификации графических примитивов.

В связи с этим различают достаточно большое количество вариаций алгоритма Хафа: для поиска окружностей [5–6], вероятностные, случайные, иерархические, рекуррентные преобразования [7–9] и др. В данной работе важно определить структуру штриховых объектов и его параметры, поэтому требуется корректировка известных подходов.

Основная часть. В распознавании штрихового изображения можно выделить два этапа: идентификация контуров (границ) распознаваемого изображения и распознавание элементов самого изображения, состоящего из примитивов, описанных ранее.

Контур в контексте обработки изображений представляет собой кривую линию, обозначающую границу между объектами или областями с различными яркостными характеристиками.

Контур отображает переходы интенсивности пикселей и позволяет выделить форму объектов. Контурные представления ключевыми позициями, между которыми положение точек, принадлежащих контуру, вычисляются интерполированием.

Одним из основных подходов к обнаружению контуров является использование операторов градиентов, таких как оператор Собеля или оператор Превитта [10]. Они позволяют выделить разности яркостных характеристик на изображении и определить места наибольших изменений интенсивности. Такие изменения указывают на границы объектов. Для идентификации примитивов можно использовать метод, основанный на геометрических характеристиках объекта. Суть метода заключается в выделении набора ключевых точек (или областей) объекта с последующим выделением набора признаков. Каждый признак является либо расстоянием между ключевыми точками, либо отношением таких расстояний.

Для распознавания примитивов будет применяться алгоритм Хафа, модифицированный для решения поставленных задач [11]. Основная идея алгоритма заключается в преобразовании из координатного пространства (декартового пространства) изображения в пространство параметров (пространство Хафа), в котором каждая геометрическая форма представляется точкой или кривой.

Для корректной работы алгоритма необходима предварительная обработка входного изображения.

Первым шагом предобработки является преобразование исходного изображения в градации серого. Данный шаг позволяет работать с яркостными характеристиками как цветных, так и черно-белых изображений.

Вторым шагом применяется пороговый фильтр для сглаживания изображения и уменьшения шума. На этом этапе могут использоваться различные варианты фильтрации: свертка с использованием оператора градиента, свертка с маской (фильтры Собеля, Превитта). Выбор фильтра зависит от качественных характеристик изображения, наличия шума и других параметров [12].

Следующим этапом алгоритма является инициализация пространства Хафа: для каждой точки границы на изображении рассматриваются все возможные прямые линии, проходящие через эту точку. В случае прямых линий используется параметрическое представление линии.

Для понимания перевода параметров из декартового пространства в пространство Хафа обратимся к известной формуле прямой линии с угловым коэффициентом.

$$y = kx + b \quad (1)$$

В декартовом пространстве координат переменная k используется для обозначения углового коэффициента, а переменная b есть показатель смещения данной линии по оси ординат от начала координат. Для любой прямой, проходящей через точку (x_0, y_0) в пространстве изображений, выполняется условие

$$y - y_0 = k(x - x_0) + b. \quad (2)$$

Через точку (x_0, y_0) в пространстве Хафа проходит прямая линия. Данная линия может быть представлена в следующем виде:

$$b = kx_0 + y_0. \quad (3)$$

Точку (k_0, b_0) из пространства Хафа используем для построения линии в декартовой системе координат:

$$b_0 = k_0 x + y. \quad (4)$$

Для описания связи между декартовым пространством и пространством Хафа используем следующее математическое выражение:

$$(x_0, y_0) \rightarrow (k_0, b_0) = \left(-\frac{x_0}{y_0}, \frac{y_0}{k_0} \right), \quad (5)$$

где (x_0, y_0) – точка в декартовом пространстве, а (k_0, b_0) – соответствующая точка в пространстве Хафа.

Обратное отображение из пространства Хафа в декартово пространство задается уравнением

$$(k_0, b_0) \rightarrow (x_0, y_0) = \left(-\frac{b_0}{k_0}, \frac{b_0}{k_0^2 + 1} \right). \quad (6)$$

Пространство Хафа обычно дискретизируется на сетку ячеек, каждая из которых представляет собой диапазон значений k и b . Для каждой линии в пространстве Хафа инкрементируется соответствующая [13]. Таким образом, выбираются ячейки с наибольшим количеством голосов (инкрементов) и соответствующие им параметры используются для построения линии на исходном изображении (в декартовом пространстве). В результате задача обнаружения заданной прямой линии на исходном изображении преобразуется в задачу о пике в пространстве параметров.

Так как угловой коэффициент k можно представить как тангенс угла наклона прямой к оси, то для любой точки изображения с координатами (x, y) в декартовом пространстве найдется точка (ρ, θ) в параметрическом пространстве, где $\theta \in [0, \pi]$:

$$\rho(\theta) = x \cos \theta + y \sin \theta. \quad (7)$$

Из этого следует, что если 3 точки (x_1, y_1) , (x_2, y_2) , (x_3, y_3) лежат на одной прямой, то при переводе их в параметрическое пространство их

три кривые пересекутся в одной точке (точке интереса), что можно наблюдать на рис. 1 [14].

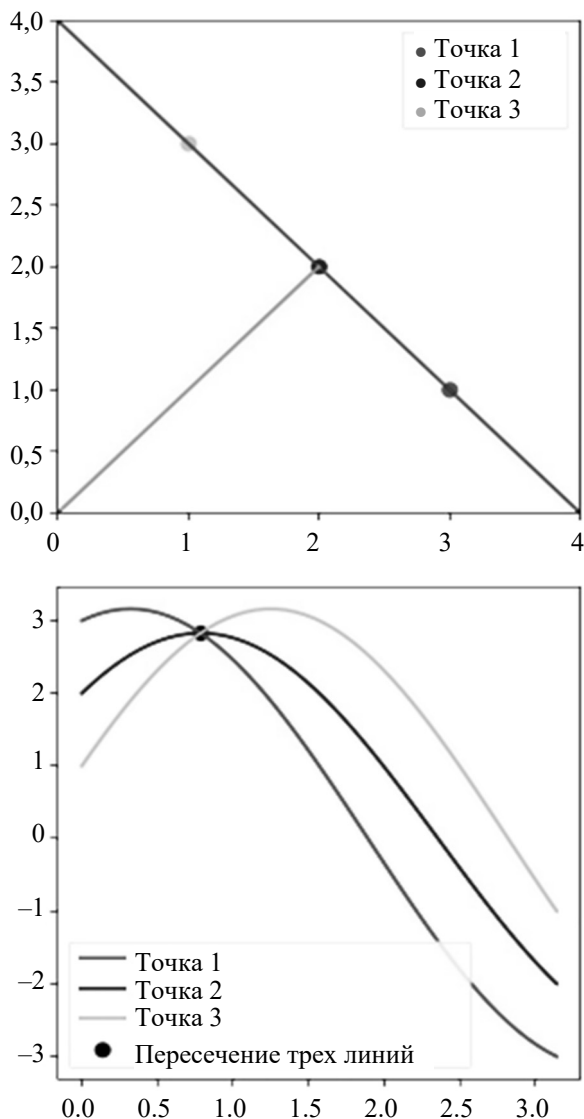


Рис. 1. Перевод точек из декартового пространства в пространство параметров

Классическое преобразование Хафа вычисляет все дискретные точки на изображении, поэтому точки за пределами прямой линии также вычисляются и сохраняются. Данная особенность может вызвать появление ложных пиков, соответственно, обнаруженная прямая линия будет ошибочной [15]. Также классический алгоритм Хафа не определяет штриховые линии как единый объект. Изменение алгоритма для решения поставленных задач может содержать следующие шаги и особенности адаптации базового алгоритма.

Особенность 1. Усиленный шаг предварительной обработки: перед применением алгоритма Хафа требуется более сложная предварительная обработка, чтобы выделить штриховые линии, особенно, если штрихи нерегулярны или имеют переменные интервалы.

Нами предложен подход итеративного применения фильтра Кернела совместно с использованием фильтра Гаусса и фильтра Канни. Многоступенчатое применение фильтров позволит убрать шум, избежать считывания зерна шума в качестве точки пика, яснее обозначить границы штриховых линий на границах сегментов.

Особенность 2. Модификация алгоритма для учета разрывов в штриховых линиях. Так как разрывы между сегментами приводят к тому, что голоса в пространстве Хафа распределяются менее плотно, алгоритм адаптируется для учета разрывов, например, путем ввода дополнительных параметров.

В качестве дополнительных параметров предложены значения минимальной длины линии (решает задачу определения штриха), а также значение максимального расстояния между линиями (решает задачу определения интервалов между штрихами).

Особенность 3. Важной задачей после определения примитивов является их классификация с последующим определением основных параметров (ширина, длина, тип линии и т. д.). Для этого рассмотрим следующие шаги постобработки. Необходимо внедрение шагов, которые идентифицируют и соединяют разрозненные сегменты линий воедино, рассчитывают длину штриха и интервала между ними.

Для определения длины штриха используем расчет Евклидова расстояния между контурами, образующими рамку вокруг каждого штриха. Для расчета длины интервала найдем расстояние между контурами соседних сегментов также с применением Евклидова расстояния.

Важно принять во внимание, что функция расчета значения длины интервала между штрихами, а также расчета длины штриха должна учитывать некоторые особенности штриховых линий. К такой особенности можно отнести окончание штриховой линии штрихом неполной длины. Данный факт может повлиять на расчет длины штриха, поэтому дополнительно необходимо реализовать функцию расчета медианной длины штриха и интервала между штрихами, что снизит вероятность расчетной погрешности.

Особенность 4. Для решения задачи определения линий толщиной более 1 пикселя необходимо ввести ограничение для исходного изображения. Так как сплошная линия толщиной более 1 пикселя не имеет перепадов яркости внутри линии, то она определяется значениями верхней границы и нижней границы линии (т. е. сторонами прямоугольника). Исходя из данной особенности для определения толщины сплошной линии можно прибегнуть к следующей последовательности шагов.

1. Сортируется массив всех линий по необходимой координате (выбор координаты для сортировки зависит от ориентации линий вертикально или горизонтально).

2. Рассчитываются расстояния между всеми соседствующими линиями. Здесь появляется необходимость ввода ограничения для алгоритма – расстояние между всеми соседствующими линиями должно быть больше толщины самой широкой линии на данном изображении. Без ввода данного ограничения не представляется возможным определить, является ли данная пара линий границами сплошной линии толщиной более 1 пикселя или же это две линии толщиной 1 пиксель, расположенные рядом.

3. Исходя из введенного ограничения определяется максимальное расстояние между линиями в массиве. Полученную величину используем для сравнения. Если расстояние между линиями по выбранной координате меньше эталона – линия сплошная, толщиной более одного пикселя. Если расстояние между линиями по выбранной координате больше эталона – две линии толщиной 1 пиксель.

В отличие от сплошной линии, в штриховых линиях присутствуют перепады яркости в местах разрыва. При работе алгоритма такие линии будут определяться как последовательность идентичных линий с шагом 1. Ось шага будет определяться в зависимости от ориентации линий.

Таким образом, толщина сплошной линии является разницей координат соседствующих линий, а толщина штриховой линии является суммой количества линий с шагом 1.

4. Для проверки определения корректности типа линий обратимся к рассчитанному в шаге 3 параметру длины штриха. Для сплошной линии длина штриха будет равна длине линии, для штриховой линии длина штриха будет меньше длины всей линии.

Заключение. Псевдокод для разработанного улучшенного алгоритма:

Algorithm 1: Line image detection and classification

1. Function $\text{calc_line_length}(x_l, y_l, x_2, y_2)$
2. Return $\text{sqrt}((x_2 - x_l)^2 + (y_2 - y_l)^2)$
3. $\text{img} \leftarrow \text{read image in grayscale mode}$
4. Apply Gaussian blur with kernel
5. Detect edges using Canny algorithm
6. Detect lines using Hough transform
7. For each line in detected lines
8. Function $\text{calc_line_length}(x_l, y_l, x_2, y_2)$
9. Find contours in the segment mask
10. Calculate segment lengths
11. If no segments detected then

12. $\text{dominant_length} \leftarrow \text{line_length}$
13. Else
14. Calculate most common segment length
15. Sort line coordinates and find max gap between lines
16. Calculate line thickness based on max gap between lines
17. Classify if line is solid or dashed based on segment length
18. Write values in file for each found line (type, thickness, segment length)

Корректность и эффективность работы алгоритма рассмотрим на примере изображения, содержащего штриховые и сплошные линии. Исходное изображение для обработки представлено на рис. 2.

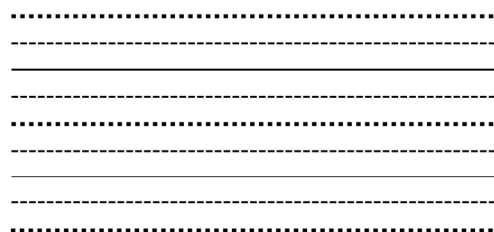


Рис. 2. Исходное изображение

Результат работы классического алгоритма Хафа представлен на рис. 3.

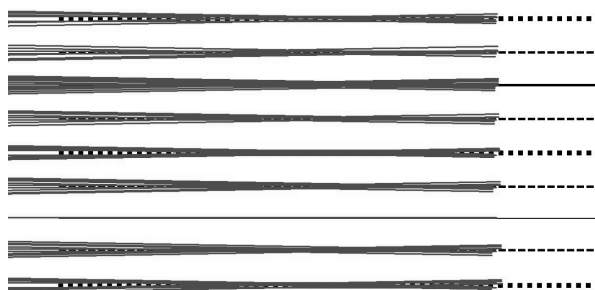


Рис. 3. Результат работы классического алгоритма Хафа

Количество линий, найденных классическим алгоритмом Хафа, на данном изображении – 38, фактическое количество 9. Как можно заметить, после обработки изображения классический алгоритм нашел линии, не относящиеся к представленным на исходном изображении. Также алгоритм не обработал линии до конца. Алгоритм не классифицирует линии, а только хранит каждую из найденных линий в виде совокупности их координат (x_1, y_1, x_2, y_2) . Исходя из совокупности описанных выше факторов не представляется возможным использовать классический алгоритм, корректно рассчитать количество линий на изображении, а также длину штриха или же толщину

линии, поскольку присутствует чрезмерная избыточность данных.

Результат работы улучшенного алгоритма на примере того же исходного изображения представлен на рис. 4.

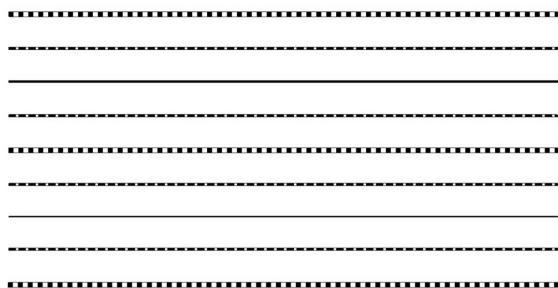


Рис. 4. Результат работы улучшенного алгоритма Хафа

Количество линий, найденных улучшенным алгоритмом на данном изображении (до

сортировки, расчета толщины линии), – 18, при фактическом количестве 9. После необходимых преобразований массива линий распознано 9 линий. Среди распознанных определено 2 сплошные и 7 штриховых линий с различной длиной сегмента (от 10 до 16 пикселей). Толщина линий определяется благодаря верно распознанным границам. Сплошная линия толщиной 1 пиксель распознана отдельным примитивом.

В данной работе исследованы и предложены шаги адаптации существующего классического алгоритма Хафа для эффективного обнаружения контура распознаваемого изображения и графических примитивов, составляющих изображение. Предложенный вариант алгоритма распознает не только сплошные, но и штриховые линии. Распознанные линии подвергаются последующей классификации с расчетом всех необходимых параметров и записью данных в файл.

Список литературы

1. Наздрюхин А. С., Храмцов И. Н., Тушев А. Н. Обработка изображений товарных чеков для выделения и распознавания текстовой информации // Вестник Дагестанского государственного технического университета. Технические науки. 2019. Т. 46. №. 4. С. 113–122.
2. Цепковская Т. А. Вопросы поиска и классификации графических примитивов внутри анализируемых изображений // Перспективное развитие науки, техники и технологий. 2020. С. 268–271.
3. Болгов А. Н., Фаворская М. Н. Применение преобразования Хафа и его модификаций для нахождения графических примитивов // Актуальные проблемы авиации и космонавтики. 2011. Т. 1. №. 7. С. 352.
4. Фаворская М. Н. Преобразование Хафа для задач распознавания // DSPA: Вопросы применения цифровой обработки сигналов. 2016. Т. 6. №. 4. С. 826–830.
5. Кудрина М. А. Использование преобразования Хафа для обнаружения прямых линий и окружностей на изображении // Известия Самарского научного центра РАН. 2014. № 4–2. С. 476–478.
6. Овчинников А., Фан Чан Данг Хо. Обработка полусферических изображений: определение центра изображения и выделение вертикальных линий // Компоненты и технологии. 2011. № 12. С. 133–136.
7. Панфилова Е., Кунина И. Использование оконного преобразования Хафа для поиска протяженных границ на изображении // Сенсорные системы. 2020. № 34. С. 340–353. DOI: 10.31857/S0235009220030075.
8. Matas J., Galambos C., Kittler J. Progressive Probabilistic Hough Transform // proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1999. Vol. 1.
9. Mammeri A., Bouckerche A., Lu G. Lane detection and tracking system based on the MSER algorithm, Hough transform and Kalman filter // Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems. 2014. P. 259–266.
10. Соловьев Ф. С., Тарасов И. Е., Петров А. Б. Распознавание образов и обнаружение контуров объекта на изображении // Инновации в науке. 2019. №. 4 (92). С. 4–9.
11. Астратов О. С., Филатов В. Н., Кузьмин С. А. Модифицированное преобразование Хафа в системе компьютерного зрения // Успехи современной радиоэлектроники. 2013. №. 2. С. 77–83.
12. Денисюк В. С. Применение и оптимизация преобразования Хафа для поиска объектов на изображении // Международный конгресс по информатике: Информационные системы и технологии: материалы Междунар. науч. конгресса, Минск, 31 окт. – 3 нояб. 2011 г.: в 2 ч. Минск, 2011. Ч. 2. С. 162–165.
13. Doermann, D. Document Analysis Systems: Theory and Practice. New York: World Scientific Publishing Co. Inc., 2016. 400 p.
14. Doermann, D. Handbook of Document Image Processing and Recognition. New York: Springer, 2014. 666 p.
15. Sonka M., Hlavac V., & Boyle R. Image processing, analysis, and machine vision. New York: Cengage Learning, 2014. 520 p.

References

1. Nazdryukhin A. S., Khramtsov I. N., Tushev A. N. Processing images of sales receipts for highlighting and recognizing text information *Vestnik Dagestanskogo gosudarstvennogo tekhnicheskogo universiteta. Tekhnicheskiye nauki* [Bulletin of the Dagestan State Technical University. Technical science], 2019, vol. 46, no. 4, pp. 113–122 (In Russian).
2. Tsepkovskaya T. A. Questions of search and classification of graphic primitives inside analyzed images. *Perspektivnoye razvitiye nauki, tekhniki i tekhnologii* [Perspective development of science, technology and technology], 2020, pp. 268–271 (In Russian).
3. Bolgov A. N., Favorskaya M. N. Application of the Hough transform and its modifications for finding graphic primitives. *Aktual'nyye problemy aviatsii i kosmonavтики* [Current problems of aviation and astronautics], 2011, vol. 1, no. 7, p. 352 (In Russian).
4. Favorskaya M. N. Hough transform for recognition problems. *DSPA: Voprosy primeneniya tsifrovoy obrabotki signalov* [DSPA: Issues of application of digital signal processing], 2016, vol. 6, no. 4, pp. 826–830 (In Russian).
5. Kudrina M. A. Using the Hough transform to detect straight lines and circles in an image. *Izvestiya Samarskogo nauchnogo tsentra RAN* [News of the Samara Scientific Center of the Russian Academy of Sciences], 2014, no. 4–2, pp. 476–478 (In Russian).
6. Ovchinnikov A., Fan Chan Dang Ho. Processing of hemispherical images: determining the center of the image and highlighting vertical lines. *Komponenty i tekhnologii* [Components and Technologies], 2011, no. 12, pp. 133–136 (In Russian).
7. Panfilova E., Kunina I. Using the windowed Hough transform to search for extended boundaries in an image. *Sensornyye sistemy* [Sensor systems], 2020, no. 34, pp. 340–353. DOI: 10.31857/S0235009220030075 (In Russian).
8. Matas J., Galambos C., Kittler J. Progressive Probabilistic Hough Transform. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1999, vol. 1.
9. Mammeri A., Bouckerche A., Lu G. Lane detection and tracking system based on the MSER algorithm, Hough transform and Kalman filter. *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. 2014, pp. 259–266.
10. Solov'yov F. S., Tarasov I. E., Petrov A. B. Pattern recognition and detection of object contours in an image. *Innovatsii v nauke* [Innovations in science], 2019, no. 4 (92), pp. 4–9 (In Russian).
11. Astratov O. S., Filatov V. N., Kuzmin S. A. Modified Hough transform in a computer vision system. *Uspekhi sovremennoy radioelektroniki* [Advances in modern radio electronics], 2013, no. 2, pp. 77–83 (In Russian).
12. Denisjuk V. S. Application and optimization of the Hough transform for searching for objects in an image. *Informatsionnyye sistemy i tekhnologii: Mezhdunarodnyy kongress po informatike* [Information Systems and Technologies: Proc. International Congress on Informatics]. Minsk, 2011, part 2, pp. 162–165 (In Russian).
13. Doermann D. Document Analysis Systems: Theory and Practice. New York, World Scientific Publishing Co. Inc. Publ., 2016, 400 p.
14. Doermann D. Handbook of Document Image Processing and Recognition. New York, Springer Publ., 2014, 666 p.
15. Sonka M., Hlavac V., & Boyle R. Image processing, analysis, and machine vision. New York, Cengage Learning Publ., 2014, 520 p.

Информация об авторах

Талапина Дарья Михайловна – магистрант кафедры информатики и веб-дизайна. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: taladarmi@mail.ru

Новосельская Ольга Александровна – кандидат технических наук, доцент кафедры информатики и веб-дизайна. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: nochka@tut.by

Information about the authors

Talapina Darya Mikhailovna – Master's degree student, the Department of Informatics and Web-design. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: taladarmi@mail.ru

Novoselskaya Olga Aleksandrovna – PhD (engineering), Assistant Professor, the Department of Informatics and Web Design. Belarusian State Technological University (220006, Minsk, Sverdlova str., 13a, Republic of Belarus). E-mail: nochka@tut.by

Поступила после доработки 24.05.2024