

19. Sirota, A. A., Mitrofanova, E. Yu., Milovanova, A. I. Analiz algoritmov poiska obyektov na izobrazheniyakh s ispolzovaniyem razlichnykh modifikatsiy svertochnykh neyronnykh setey [Analysis of algorithms for searching objects in images using various modifications of convolutional neural networks]. *Vestnik Voronezhskogo gosudarstvennogo universiteta. Seriya: Sistemnyy analiz i informatsionnyye tekhnologii* [Bulletin of the Voronezh State University. Series: System Analysis and Information Technologies], 2019, no. 3, pp. 123–137.

20. Ranzato, Marc'Aurelio, Christopher, Poultney, Sumit, Chopra and Yann, LeCun. Efficient Learning of Sparse Representations with an Energy-Based Model. *Advances in Neural Information Processing Systems (NIPS 2006)*, MIT Press, 2006.

21. Kravtsov, E. V., Nevrov, D. Yu., Selitrennikov, D. A. Metodika raspoznavaniya obyektov pri pomoshchi svertochnoy neyronnoy seti YOLOV3 [Object recognition technique using convolutional neural network YOLOV3]. *Tekhnika i bezopasnost obyektov ugovovno-ispolnitelnoy sistemy : sbornik materialov Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Technique and safety of objects of the penal system : collection of materials of the International Scientific and Practical Conference], in 2 vol. Voronezh, 2020, pp. 301–304.

22. Dzhoshi, P. *Iskusstvennyy intellekt s primerami na Python* [Artificial intelligence with examples in Python]. Moscow, Dialektika Publ. ; Vilyams Publ., 2019. 448 p.

УДК 004.93

АЛГОРИТМ БЫСТРОГО НАХОЖДЕНИЯ КОНТУРА НА ОСНОВЕ МЕТОДА СЛЕДОВАНИЯ ЗА ПИКСЕЛЯМИ

Статья поступила в редакцию 30.03.2022, в окончательном варианте – 28.04.2022.

Конаныхина Татьяна Николаевна, Юго-Западный государственный университет, 305014, Российская Федерация, г. Курск, ул. 50 лет Октября, 94,

кандидат технических наук, доцент, ORCID: 0000-0002-2755-0163, e-mail: govtn@mail.ru

Конаныхин Александр Юрьевич, Юго-Западный государственный университет, 305014, Российская Федерация, г. Курск, ул. 50 лет Октября, 94,

аспирант, ORCID: 0000-0003-1644-6675, e-mail: alexanderkonan@yandex.ru

Авилов Иван Александрович, Юго-Западный государственный университет, 305014, Российская Федерация, г. Курск, ул. 50 лет Октября, 94,

студент, ORCID: 0000-0001-8884-7705, e-mail: vano94600@gmail.com

Предложен алгоритм быстрого нахождения контура, в основе которого находится метод следования за пикселями изображения. Отслеживание и извлечение контурных пикселей широко используется в различных областях человеческой деятельности, анализ литературы говорит о постоянном увеличении применения этих методов в связи с ростом носимой электроники и постоянным совершенствованием самих алгоритмов трассировки. Традиционные способы отслеживания пикселей имеют определенные ограничения, некоторые алгоритмы, выполняют ненужные операции перемещения над белыми пикселями, что увеличивает время обработки изображения и нахождения контура. Не все алгоритмы могут определять контур в случае угловых пикселей; следовательно, они не могут быть описательной характеристикой объекта и определять связь между объектами. Предложенный алгоритм определяет пиксели внутреннего-внешнего угла, переднего-внутреннего угла и передней прямой линии. При экспериментальной проверке работоспособности предложенного алгоритма и сравнении качества его работы с различными алгоритмами (простой граничный повторитель, модифицированный простой граничный повторитель, трассировка соседей Мура, алгоритм радиальной развертки) предложенный алгоритм определил 99,4 % от общего числа пикселей контура и имеет максимальное общее количество трассируемых пикселей в сравнении с другими анализируемыми алгоритмами трассировки.

Ключевые слова: нахождение контура, алгоритм трассировки, распознавание изображений, метод отслеживания пикселей

ALGORITHM FOR FAST FINDING THE CONTOUR BASED ON THE PIXEL FOLLOWING METHOD

The article was received by the editorial board on 30.03.2022, in the final version – 28.04.2022.

Konanykhina Tatyana N., Southwest State University, 94 50 let Oktyabrya St., Kursk, 305014, Russian Federation,

Cand. Sci. (Engineering), Associate Professor, ORCID 0000-0002-2755-0163, e-mail: govtn@mail.ru

Konanykhin Alexander Yu., Southwest State University, 94 50 let Oktyabrya St., Kursk, 305014, Russian Federation,

postgraduate student, ORCID 0000-0002-2755-0163, e-mail: govtn@mail.ru

Avilov Ivan A., Southwest State University, 94 50 let Oktyabrya St., Kursk, 305014, Russian Federation,

student, ORCID 0000-0001-8884-7705, e-mail: vano94600@gmail.com

An algorithm for fast contour finding is proposed, which is based on the method of following image pixels. Edge pixel tracking and extraction are widely used for smart devices in object detection, logo recognition, object separation from their

background, object size calculation, shape classification, object feature point finding by edge length and shape, etc. Traditional pixel tracking methods have certain limitations, some algorithms perform unnecessary moving operations on white pixels, which increases the time of image processing and finding the contour. Not all algorithms can determine the contour in the case of corner pixels; therefore, they cannot be a descriptive characteristic of an object and define a relationship between objects. The proposed algorithm determines the pixels of the inner-outer corner, the front-inner corner, and the front straight line. During the experiments, the accuracy of pixel detection on the test image was compared by the proposed algorithm and other algorithms (simple edge repeater, modified GWP, Moore neighbor tracing, radial sweep algorithm), the proposed algorithm determined 99.4 % of the total number of contours pixels and has the maximum total number of traced pixels compared to other tracing algorithms.

Keywords: contour finding, tracing algorithm, image recognition, pixel tracking method

Graphical annotation (Графическая аннотация)



Изображение трассированное с помощью алгоритма МППГ
(не обнаружены пиксели внутреннего угла)
Image traced using the MPPG algorithm
(no inner corner pixels detected)



Траектории поиска пикселей предложенным алгоритмом, способным отследить пиксели на прямой, пиксели внутренне-внешнего, внутреннего и внешнего угла
Pixel search trajectories by the proposed algorithm, capable of tracking pixels on a straight line, pixels of the inner-outer, inner and outer corner



Изображение трассированное с помощью предложенного алгоритма
Image traced using the proposed algorithm

Введение. Обнаружение контуров является важной научно-технической задачей, которая в настоящий момент находит широкое применение в различных областях человеческой деятельности [1–4].

В настоящий момент с увеличением количества носимой электроники и увеличением вычислительной мощности персональных мобильных устройств (мобильных телефонов, смарт-часов) все более актуальной является задача создания алгоритмов быстрого нахождения контуров на изображениях, которые позволяют уменьшать объем хранимой и обрабатываемой информации в виду хоть и возросших, но ограниченных аппаратных ресурсов мобильной техники, но при этом позволяющих увеличить качество распознавания объекта (символа, логотипа, лиц и т.д.). Так как данная область исследований является актуальной и имеет широкое применение, поэтому было создано достаточно большое количество алгоритмов трассировки контуров объектов, но предложенные алгоритмы имеют ряд ограничений и недостатков. Например, большинство алгоритмов не пропускают контурные пиксели, то есть не обладают достаточной точностью распознавания, некоторые алгоритмы трассировки проходят несколько раз по одним и тем же пикселям, что приводит к увеличению времени обработки изображения при отслеживании пикселя, большая часть алгоритмов либо совсем не предусматривает сжатие данных или имеет неточности при восстановлении контура из сохраненной сжатой информации.

Обзор существующих решений. Чаще всего для трассировки необходимо сначала сделать изображение бинарным, так как большинство алгоритмов основаны на том, что контур – это черные пиксели, а фон – белые. Следовательно, если черный пиксель относится к контуру, то рядом с ним обязательно находится хотя бы один пиксель белого цвета [1, 2].

Алгоритмы нахождения контура делятся на алгоритмы, отслеживающие пиксели, вершины (угловые пиксели), и алгоритмы, отслеживающие контурные пиксели на основании данных прогона всех пикселей изображения. Метод отслеживания пикселей в настоящий момент является самым распространенным [1, 2]. Методы, относящиеся к данной группе методов, отслеживают пиксели согласно определенному правилу, например, проверяются все пиксели по часовой стрелке от уже найденного пикселя контура, далее координаты пикселей сохраняются в памяти в соответствии с порядком их обнаружения. К таким методам относятся простой граничный повторитель (ППП), модифицированный ППП (МППП), алгоритм радиальной развертки (АРР), трассировка соседей Мура (ТСМ) [1–3]. Данные методы обнаруживают пиксели

контура на основе цепного кода и работают на основе достаточно простых правил трассировки, сохраняют только координаты идентифицированных контурных пикселей, то есть для запоминания результатов трассировки требуют меньше памяти, объем которой составляет размер кадра и менее.

К методам, требующим для сохранения результатов трассировки меньший объем памяти, относятся методы группы отслеживания вершин, потому что они отслеживают только угловые пиксели контура, а также то, как эти угловые пиксели соединены между собой. Данные методы также используют цепной код и позволяют уменьшить количество точек, которые необходимо запомнить для сохранения найденного контура.

Две первые группы методов находят контурные пиксели, не сканируя при этом все пиксели изображения, последовательность отслеживания генерируется автоматически по определенному правилу. Но данные алгоритмы не подходят для больших изображений, потому что они могут содержать большое количество контуров, что приведет к выполнению большого количества операций [2, 5, 6]. Это в свою очередь приведет к тому, что выполнение программы, работающей по таким алгоритмам, потребует большой объем памяти, чем если бы эта программа работала на основании алгоритма, основанного на группе методов, приведенных ниже.

Методы отслеживания контурных пикселей на основе данных прогона сложнее других методов, но в случае с большими изображениями они позволяют увеличить скорость обработки, поскольку все пиксели сканируются один раз, что не приводит к необходимости совершать дополнительные операции. Так как у объекта могут быть как внешний, так и несколько внутренних контуров, то можно выделить пять типов данных, которые могут быть получены после прогона: левый край внешнего контура, правый край внешнего контура; левый край внешнего контура, левый край внутреннего контура; правый край внутреннего контура, левый край внутреннего контура; правый край внутреннего контура, правый край внешнего контура и левый край внешнего контура. Для отслеживания контура метод отслеживания контурных пикселей строит отношение следования за прогоном между крайними точками двух соседних строк сканирования [1, 2, 3]. К данным методам можно отнести метод кода RD и метод OpenCV [1, 4, 7, 8].

Рассмотрим общие принципы работы трассировщиков всех вышеприведенных типов. Пусть A является бинарным изображением размером $X \times Y$. Большинство алгоритмов отслеживания контуров используют трассер $T(P, d)$ с абсолютной информацией о направлении $d \in \{N, NE, NW, W, SW, S, SE, E, NE\}$. Трассировщик начинает работу после нахождения и сохранения первой точки контура и определения направления трассировки. Следующая точка контура определяется трассировщиком по определенному правилу, после чего трассер перемещается в обнаруженную точку и определяется дальнейшее направление следования. Следующая точка изображения может оказаться как точкой контура, так и точкой фона. Чтобы понять это, необходимо определить интенсивность пикселя и новое абсолютное направление dr , используя информацию об относительном направлении $r \in \{\text{спереди, спереди – слева, слева, сзади – слева, сзади, сзади – справа, справа, спереди – справа}\}$ [1, 2, 9, 10]. При возвращении трассировщика в начальную точку процедура обнаружения контуров заканчивается, то есть это является правилом остановки.

Можно выделить четыре типа пикселей контура (рис. 1): пиксель, принадлежащий прямой линии, внутренний угол, внешний угол, пиксель внутренне-внешнего угла.

	0	1	2	3	4	5	6	7	8	9
0										
1			1	1						
2			4	4				1	1	
3			1	3				3	1	
4					3	4	3			
5				1	2		4			
6			1	2			4			
7				1	2		4			
8					1	4	1			
9										

Рисунок 1 – Типы пикселей контура: пиксель внешнего угла (1), пиксель внутреннего угла (2), пиксель внутреннего-внешнего угла (3), пиксель, принадлежащий ребру (прямой линии) (4)

Рассмотрим недостатки и ограничения самых распространенных современных алгоритмов трассировки, которые подходят для обработки изображений мобильными вычислительными системами в реальном времени.

Простейшим алгоритмом отслеживания контуров является простой граничный повторитель. В различной литературе можно встретить названия алгоритма черепахи Пейперта или отслеживания квадратов [1, 2, 11, 12]. Данный алгоритм приведен на рисунке 2 и обладает значительными недостатками, он не может идентифицировать пиксели контура, которые расположены во внутренне-внешнем углу,

если трассировщик находится впереди справа от искомого пикселя. Для устранения данного ограничения применяется модифицированный ППП [2, 13], но попытки отследить упущенные алгоритмом ППП пиксели приводят к тому, что трассировщик перемещается как по пикселям контура, так и по пикселям фона, что ведет к увеличению времени трассировки, а также этот метод не позволяет отследить пиксели внутреннего угла, если они находятся перед трассировщиком.

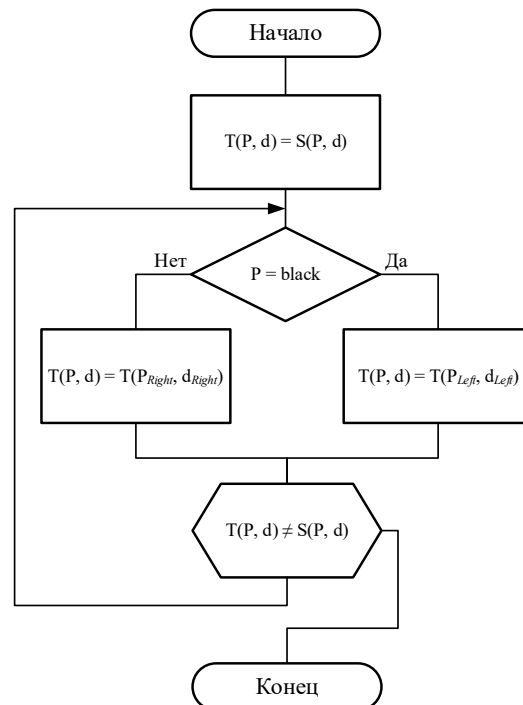


Рисунок 2 – Алгоритм простого граничного повторителя

ТСМ начинает трассировку с заднего пикселя, то есть трассер сначала движется назад, а затем начинает проверку соседних пикселей по часовой стрелке, используя восемь связанных кодов, проверяются задний левый, левый, передний левый, передний, правый передний, правый и правый задний пиксели [5, 6].

APP похож на ТСМ, но его трассировщик не имеет информации о направлении, он поддерживает две точки: предыдущий пиксель и текущий пиксель для определения начального направления трассировки [1, 13].

Оба этих метода не позволяют отследить внутренние углы. Поэтому актуальной является задача создания улучшенного алгоритма, преодолевающего эти ограничения [14].

Алгоритм нахождения контура, преодолевающий указанные ограничения. Данный алгоритм (рис. 3) имеет шесть вариантов отслеживания пикселей контура на основе локальных шаблонов (рис. 4).

Процесс трассировки начинается после того, как трассером будет обнаружен первый черный пиксель, далее поиск следующего контурного пикселя происходит по нижеследующим правилам.

1. На первом этапе проверяется интенсивность соседнего левого пикселя. Если он белый, переходим к этапу 2, если он черный, то необходимо проверить, есть ли рядом с ним хотя бы один белый пиксель, если он отсутствует, то обнаруженный пиксель не является контурным, трассировщик остается на месте $T(P, d) = T(P, d_{Front})$, если белый пиксель обнаружен рядом с только что найденным черным пикселем, обнаруженный пиксель является соседним левым пикселем контура (рис. 4а), трассировщик перемещается на этот пиксель $T(P, d) = T(P_{Left}, d_{Left})$, абсолютное направление трассировки изменяется на left, после нахождения любого контурного пикселя необходимо перейти в пункт 6, затем цикл проверок начинается сначала, если выполнено условия $T(P, d) \neq S(P, d)$, то есть мы снова проверяем интенсивность соседнего левого пикселя, если он белый, то мы переходим ко 2 пункту и т.д.

2. Определяется интенсивность пикселей, находящихся слева сзади и сзади от ранее обнаруженного пикселя контура. Если левый задний пиксель черный, а задний пиксель белый, то мы обнаружили внутренне-внешний угол, который находится слева внизу (сзади) от ранее обнаруженного контурного пикселя (рис. 4б), трассер перемещается $T(P, d) = T(P_{Left-Rear}, d_{Rear})$. Абсолютное направление трассировки изменяется на Rear.

3. Если интенсивность какого-либо из двух пикселей иная, трассировщик проверяет передний левый пиксель, если он черный, переходим в пункт 4, если белый – в пункт 5.

4. Мы имеем дело либо с внутренним углом, который находится впереди от ранее трассируемого пикселя контура (рис. 4г), либо с внутренне-внешним углом, который находится спереди слева от ранее обнаруженного контурного пикселя (рис. 4в), если он белый, то необходимо перейти к пункту 4. Для того чтобы понять это, необходимо проверить интенсивность переднего пикселя, если он черный, то обнаружен случай, приведенный на рисунке 4г, трассер перемещается сначала на один пиксель вперед, помечая его пикселем контура $T(P, d) = T(P_{Front}, d)$, затем сразу на один пиксель влево $T(P, d) = T(P_{Left}, d)$, то есть обнаружено два контурных пикселя. Если передний пиксель оказался белым, то идентифицирован случай, приведенный на рисунке 4в, трассер перемещается по диагонали вперед-влево $T(P, d) = T(P_{Front-Left}, d)$.

5. Необходимо проверить интенсивность переднего пикселя. Если он черный, то мы обнаружили передний соседний пиксель (рис. 4д), абсолютное направление трассировщика изменяется на Right, трассер перемещается вперед, помечая пиксель как контурный $T(P, d) = T(P_{Front}, d_{Rear})$. Если передний пиксель белый, то мы обнаружили пиксель внешнего угла, трассер не перемещается $T(P, d) = T(P, d_{Rear})$, абсолютное направление трассировщика изменяется на Rear. Далее запускается подпрограмма поиска контурных пикселей справа от текущего пикселя.

6. После обнаружения каждого следующего контурного пикселя обязательно происходит проверка, не перешли ли мы в ходе трассировки на начальный пиксель контура (пиксель, с которого алгоритм начал свою работу). Если это произошло $T(P, d) = S(P, d)$, мы получили замкнутый контур. Далее начинается работу алгоритм поиска первого пикселя следующего контура. Если в ходе проверки трассер не переместился, а остался в начальной точке, то алгоритм продолжает свою работу.

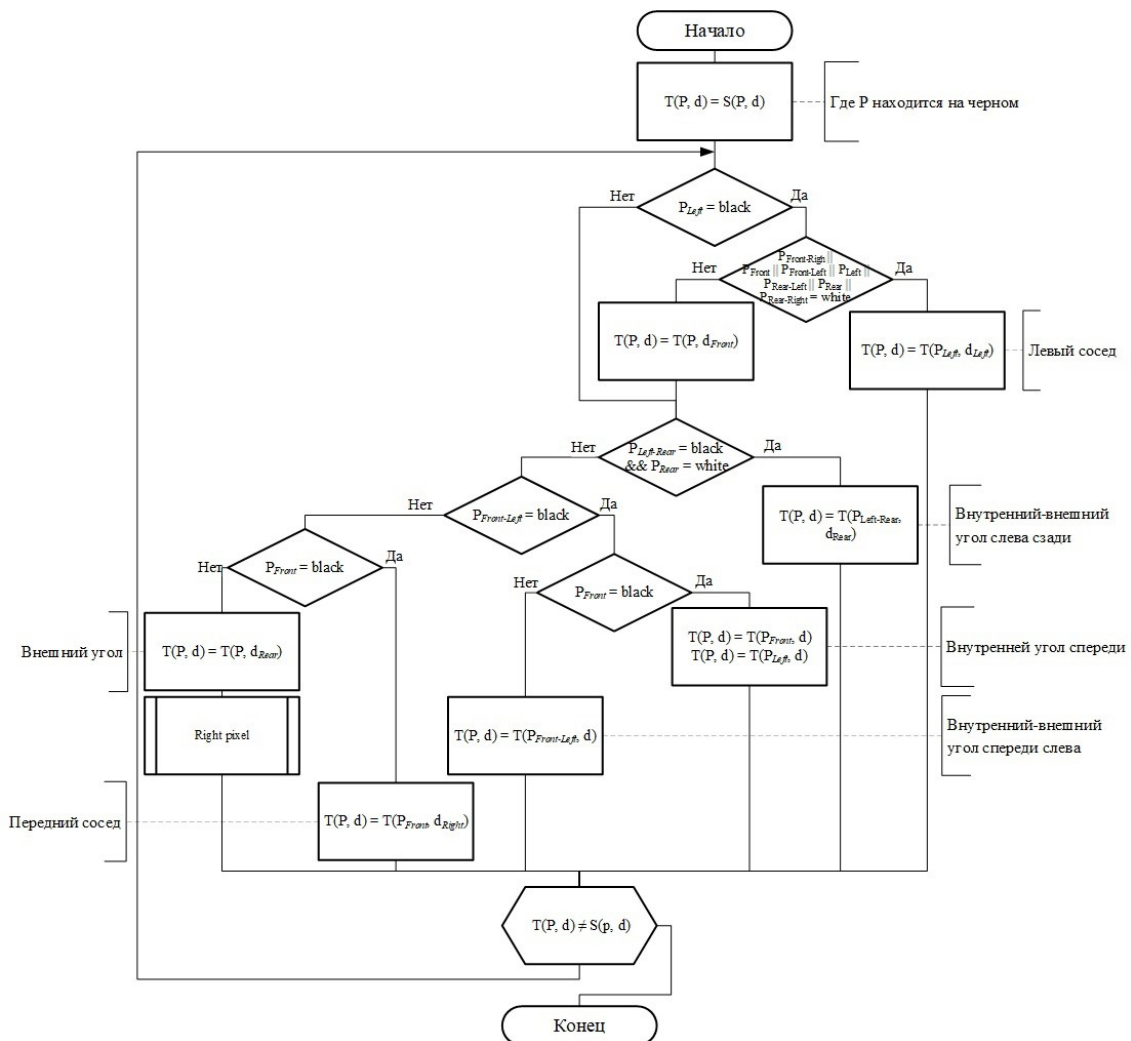


Рисунок 3 – Предложенный улучшенный алгоритм трассировки контура

Алгоритм ППП не способен выполнить трассировку случая 4б и 4г, а МППП не может обнаружить пиксель контура в случае, показанном на 4г, предложенный алгоритм успешно выполняет трассировку во всех случаях. Также предложенный алгоритм не просто позволяет находить контурные пиксели, а способен идентифицировать найденный пиксель как пиксель прямой линии или угловой пиксель (внутренний-внешний угол; внутренний угол; внешний угол). В дальнейшем эту информацию предполагается использовать для уменьшения объема памяти, используемой для хранения данных о контуре, за счет того, что достаточно хранить только репрезентативные точки контура, из которых впоследствии можно восстановить весь контур.

На рисунке 4 пунктирной линией показано направление поиска пикселя контура трассировщиком, т.е. получение информации о том, является ли пиксель черным или белым, но при этом не происходит перемещение трассировщика. Трассировщик перемещается только при обнаружении следующего контурного пикселя, не перемещаясь по белым пикселям, что уменьшает время трассировки, также объем хранимой после трассировки информации. Точкой показан текущий пиксель, который уже определен как элемент контура, стрелкой показан целевой пиксель контура, т.е. пиксель, который мы пытаемся обнаружить.

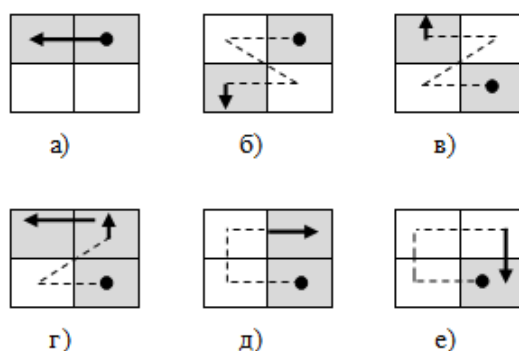


Рисунок 4 – Траектории поиска пикселей контура предложенным алгоритмом, обнаружение: (а) соседнего левого пикселя; (б) внутреннего-внешнего угла слева сзади; (в) внутреннего-внешнего угла спереди слева; г) внутреннего угла спереди; д) переднего соседа; е) внешнего угла

Алгоритм отслеживания пикселей требует критериев запуска и остановки, чтобы избежать неполного или бесконечного цикла отслеживания.

Трассировка начинается, когда трассировщик переходит с белого пикселя в черный, это означает, что для начала работы алгоритма трассировщик должен находиться на черном пикселе, перейдя в него из белого. Предлагается использовать критерий остановки Джейкоба [1, 2], который завершает трассировку, когда трассировщик повторно входит в начальный пиксель с абсолютным направлением, совпадающим с начальным направлением, т.е. трассировка заканчивается в начальном пикселе.

Анализ точности трассировки предложенного алгоритма в сравнении с имеющимися решениями. Точность трассировки контуров показывает, насколько качественно алгоритм находит все контурные пиксели на исследуемом изображении. Для этого мы проверяем работу алгоритмов на тестовых изображениях, отмечаем найденные алгоритмом пиксели, затем подсчитываем все пиксели контура. Если пиксель трассируется несколько раз, то учитывается в статистике он только один раз.

В ходе исследования качество работы предложенного алгоритма проверялось на изображениях, содержащих символическую информацию (текст на русском, английском, комбинации английского и китайского языков).

В таблице представлены результаты сравнения предложенного алгоритма с традиционными. Столбец «Общее количество» в таблице подразумевает общее количество пикселей контура на изображении, включая внутренний угол, внешний угол, внутренний-внешний угол и пиксели прямой линии. В этом исследовании ТСМ и АРР трассировали наименьшее количество контурных пикселей, они не могли трассировать пиксели во внутренних углах. ППП также отслеживал меньше пикселей по сравнению с предложенным алгоритмом, поскольку не всегда верно определялись внутренние и внутренне-внешние углы.

Кроме того, МППП имеет неточности в определении внутренних углов, которые аналогичны несоответствиям ППП, и МППП отслеживает меньше пикселей по сравнению с предложенным алгоритмом. Предложенный алгоритм определил 99,4 % от общего числа пикселей контура и имеет максимальное общее количество трассируемых пикселей контура. Предложенный алгоритм дал наилучшие результаты в отношении точности трассировки.

Пример изображения, на котором проводилось определение точности работы предложенного алгоритма, приведен на рисунке 5.

Приведенный алгоритм предлагается применять для обнаружения контуров при быстродействующей обработке символической информации в условиях малого размера символа. Методы улучшения выделенной области изображения, предваряющие трассировку приведены в работе [7].

Таблица 1 – Сравнение качества обнаружения пикселей контуров различными алгоритмами

Номер тестового изображения	Общее количество, шт	Процент обнаруженных пикселей контура, %				
		TCM	APP	ПГП	МПГП	Предложенный алгоритм
1	8122	80,7	80,7	90,7	90,8	100
2	4092	76,4	76,4	88,5	88,5	100
3	13245	82,9	82,9	91,9	91,5	100
4	29281	80,9	80,9	90,7	90,7	100
5	16837	80,9	80,9	90,5	90,5	100
6	8797	83,4	83,4	92,5	92,5	97,9
7	17271	84,1	84,1	93,1	93,1	99,9
8	15703	78,6	78,6	89,5	89,5	98,5
9	48618	79,7	79,7	92,1	92,1	98,3
Итого	161966	80,8	80,8	91,1	91,0	99,4

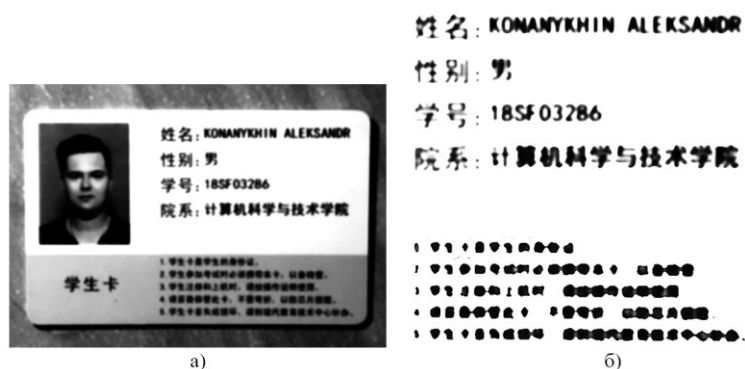


Рисунок 5 – Пример анализируемого изображения: а) изображение до предварительной обработки; б) исследуемое изображение после предварительной обработки, на котором проводилась проверка качества работы предложенного алгоритма

Результаты нахождения контурных пикселей на приведенном изображении показаны в таблице, изображению был присвоен номер 9. Для данного изображения процент обнаруженных контурных пикселей самый низкий ввиду достаточно большого зашумления изображения, нахождению на изображении большего количества пикселей внутренне-внешнего угла по сравнению с другими анализируемыми изображениями, так как на данном изображении присутствуют иероглифы.

Примеры пикселей, которые не удалось отследить предложенным алгоритмом, приведены на рисунке 6.

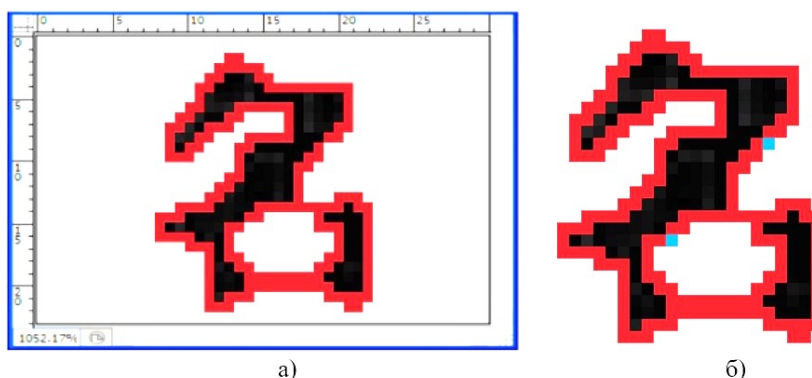


Рисунок 6 – Пример неотслеженных пикселей контура: а) контурные пиксели, обнаруженные предложенным алгоритмом (красным цветом показаны идентифицированные пиксели контура); б) синим цветом показан пиксель, который не удалось трассировать

Наилучшие результаты были получены на изображениях с разрешением 200 точек на дюйм (dpi) и выше, не содержащих иероглифы.

Наибольшее количество неточностей при трассировке предложенным алгоритмом возникает, если изображение напоминает шахматную доску, то есть имеются части изображения, представляющие собой единичные пиксели черного и белого цветов, расположенные в шахматном порядке, потому что такие изображения содержат большое количество внутренних и внешних углов. Такие ситуации при распознавании символов чаще возникают при малом размере символов, а также при обнаружении контуров у иероглифов.

Заключение. Вышеупомянутые традиционные способы отслеживания пикселей имеют определенные ограничения. Во-первых, некоторые алгоритмы, такие как ПГП и МПГП, выполняют ненужные операции перемещения над белыми пикселями, что увеличивает время обработки изображения и нахождения

контура. Во-вторых, не все алгоритмы могут определять контур в случае угловых пикселей; следовательно, они не могут дать описательную характеристику объекта и определить связь между объектами. Например, алгоритм МПП не всегда способен определить пиксель внутреннего-внешнего угла. Более того, МПП, как и ПП, также не может определить пиксели внешнего и внутреннего угла.

Приведенный алгоритм смог безошибочно идентифицировать пиксели внутреннего-внешнего угла, переднего-внутреннего угла и прямой линии, но не во всех случаях смог идентифицировать левый-внутренний угол и некоторые пиксели внешнего угла. Несмотря на это, предложенный алгоритм показал максимальную точность трассировки среди предложенных алгоритмов. Данный алгоритм идентифицирует также тип контурного пикселя, что в дальнейшем может использоваться для сжатия информации о хранимом контуре с возможностью последующего его восстановления по репрезентативным точкам.

Библиографический список

1. Jonghoon, Seo. Fast Contour-Tracing Algorithm Based on a Pixel-Following Method for Image Sensors / Jonghoon Seo, Seungho Chae, Jinwook Shim, Dongchul Kim, Cheolho Cheong, Tack-Don Han // *Sensors*. – 2016. – № 16 (3). – P. 353.
2. Соловьев, Ф. С. Распознавание образов и обнаружение контуров объекта на изображении / Ф. С. Соловьев, И. Е. Тарасов, А. Б. Петров // *Инновации в науке*. – 2019. – № 4 (92). – С. 4–9.
3. Reddy, P. R. Evaluation of stopping criterion in contour tracing algorithms / P. R. Reddy, V. Amarnadh, M. Bhaskar // *Int. J. Comput. Sci. Inf. Technol.* – 2012. – № 3. – P. 3888–3894.
4. Konanykhin, A. Y. Algorithm for finding contours for the purpose of processing symbolic information in conditions of small symbol size and chromatic aberrations / A. Y. Konanykhin, T. N. Konanykhina // *Journal of Physics: Conference Series*. – 2021. – № 1843 (1). – P. 012018.
5. Бурмака, А. А. Системная модель распознавания и идентификации состояния предстательной железы по ее изображениям УЗИ и ТРУЗИ / А. А. Бурмака, Т. Н. Говорухина, К. В. Разумова // *Научные технологии*. – 2014. – Т. 15, № 12. – С. 13–20.
6. Konanykhin, A. Yu. Processing of MRI data for oncological diseases prediction / A. Yu. Konanykhin, T. N. Konanykhina // *Медико-экологические информационные технологии – 2021 : сборник научных статей по материалам XXIV Международной научно-технической конференции*. – Курск, 2021. – С. 76–83.
7. Конаныхин, А. Ю. Методы улучшения выделенной области изображения при быстросействующей обработке символьной информации / А. Ю. Конаныхин, Т. Н. Конаныхина, В. С. Панищев // *Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение*. – 2021. – Т. 11, № 4. – С. 106–120.
8. Dolly, Indra. Feature Extraction of Bisindo Alphabets, Using Chain Code Contour / Dolly Indra, Sarifuddin Madenda, Eri Prasetyo Wibowo // *International Journal of Engineering and Technology (IJET)*. – Vol. 9, № 4. – P. 3233–3242.
9. Галеев, Д. Т. Использование искусственных нейронных сетей для распознавания образов уличных сцен / Д. Т. Галеев, В. С. Панищев // *Информационные технологии и математическое моделирование систем 2020 : труды Международной научно-технической конференции*. – Одинцово, 2020. – С. 132–134.
10. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман. – Москва : Бином, 2006. – С. 762.
11. Потапов, А. Системы компьютерного зрения: современные задачи и методы / А. Потапов // *Control Engineering*. – 2014. – № 1. – С. 20–26.
12. Панищев, В. С. Распознавание символьной информации для автоматизации производственных процессов / В. С. Панищев, М. И. Труфанов, О. Г. Добросердов, О. О. Хомяков // *Известия Юго-Западного государственного университета*. – 2021. – Т. 25, № 1. – С. 122–137.
13. Shelhamer, E. Fully Convolutional Networks for Semantic Segmentation / E. Shelhamer, J. Long, T. Darrell // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2017. – P. 640–651.
14. Ляшева, С. А. Метод последовательного уточнения границ на изображениях дорожной обстановки / С. А. Ляшева, М. П. Шлеймович // *Прикаспийский журнал: управление и высокие технологии*. – 2020. – № 4. – С. 21–31.

References

1. Jonghoon, Seo, Seungho, Chae, Jinwook, Shim, Dongchul, Kim, Cheolho, Cheong, Tack-Don, Han. Fast Contour-Tracing Algorithm Based on a Pixel-Following Method for Image Sensors, *Sensors*, 2016, no. 16 (3), p. 353.
2. Soloviev, F. S., Tarasov, I. E., Petrov, A. B. Raspoznavanie obrazov i obnaruzhenie konturov obekta na izobrazhenii [Pattern Recognition and Detection of Object Contours in an Image]. *Innovatsii v nauke* [Innovations in Science], 2019, no. 4 (92), pp. 4–9.
3. Reddy, P. R., Amarnadh, V., Bhaskar, M. Evaluation of stopping criterion in contour tracing algorithms. *Int. J. Comput. Sci. Inf. Technol.*, 2012, no. 3, pp. 3888–3894.
4. Konanykhin, A. Y., Konanykhina, T. N. Algorithm for finding contours for the purpose of processing symbolic information in conditions of small symbol size and chromatic aberrations. *Journal of Physics: Conference Series*, 2021, no. 1843 (1), p. 012018.
5. Burmaka, A. A., Govorukhina, T. N., Razumova, K. V. Sistemnaya model raspoznavaniya i identifikatsii sostoyaniya predstatelnoy zhelezy po ee izobrazheniyam UZI i TRUZI [A systemic model for recognizing and identifying the state of the prostate gland based on its ultrasound and TRUS images]. *Naukoemkie tekhnologii* [High Technologies], 2014, no. 12 (15), pp. 13–20.
6. Konanykhin, A. Yu., Konanykhina, T. N. Processing of MRI data for oncological diseases prediction. *Mediko-ekologicheskie informatsionnye tekhnologii – 2021 : sbornik nauchnykh statey po materialam XXIV Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii* [Medical and environmental information technologies – 2021 : a collection of scientific articles based on the materials of the XXIV International Scientific and Technical Conference]. Kursk, 2021, pp. 76–83.
7. Konanykhin A. Yu., Konanykhina, T. N., Panishchev, V. S. Metody uluchsheniya vydelennoy oblasti izobrazheniya pri bystrodeystvuyushchey obrabotke simvolnoy informatsii [Methods for improving the selected area of the image during

high-speed processing of symbolic information]. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika, informatika. Meditsinskoe priborostroenie* [Proceedings of the South-Western State University. Series: Management, Computer Engineering, Informatics. Medical instrumentation], 2021, no. 4 (11), pp. 106–120.

8. Dolly, Indra, Sarifuddin, Madenda, Eri, Prasetyo, Wibowo. Feature Extraction of Bisindo Alphabets, Using Chain Code Contour. *International Journal of Engineering and Technology (IJET)*, vol. 9, no. 4, pp. 3233–3242.

9. Galeev, D. T., Panishchev, V. S. Ispolzovanie iskusstvennykh neyronnykh setey dlya raspoznavaniya obrazov ulichnykh stsen [Using artificial neural networks for pattern recognition of street scenes]. *Informatsionnye tekhnologii i matematicheskoe modelirovanie sistem 2020 : trudy Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii* [Information technologies and mathematical modeling of systems 2020 : Proceedings of the International Scientific and Technical Conference]. Odintsovo, 2020, pp. 132–134.

10. Shapiro, L., Stockman, J. *Kompyuternoe zrenie* [Computer vision]. Moscow, Binom Publ., 2006, p. 762.

11. Potapov, A. Sistemy kompyuternogo zreniya: sovremennyye zadachi i metody [Computer vision systems: modern problems and methods]. *Control Engineering*, 2014, no. 1, pp. 20–26.

12. Panishchev, V. S., Trufanov, M. I., Dobroserdov, O. G., Khomyakov, O. O. Raspoznavanie simvolnoy informatsii dlya avtomatizatsii proizvodstvennykh protsessov [Recognition of symbolic information for automation of production processes]. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta* [Bulletin of the Southwestern State University], 2021, vol. 25, no. 1, pp. 122–137.

13. Shelhamer, E., Long, J., Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, pp. 640–651.

14. Lyasheva, S. A., Shlyemovich, M. P. Metod posledovatel'nogo utochneniya granits na izobrazheniyakh dorozhnoy obstanovki [Method of Sequential Refinement of Borders on Images of Road Conditions]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2020, no. 4, pp. 21–31.