

**Козловский Антон Николаевич**

*магистр технических наук*

**Kazlouski Anton**

*Master of Engineering Science*

**АЛГОРИТМ ОБНАРУЖЕНИЯ ВЕРШИНЫ УГЛА НА ОСНОВЕ  
АППРОКСИМАЦИИ КОНТУРА БИНАРНОГО ИЗОБРАЖЕНИЯ  
VERTEX DETECTION ALGORITHM BASED ON BINARY IMAGE  
CONTOUR APPROXIMATION**

***Аннотация.** Задача обнаружения вершины угла является одной из центральных проблем цифровой обработки изображений. Рассматривается модификация разработанного автором алгоритма обнаружения вершины угла. Предложено использовать алгоритм прослеживания контура бинарного изображения на основе анализа окрестности Мура. Это позволило улучшить правила начала и завершения прослеживания, а также правила определения претендента вершиной угла. Показано, что предложенный алгоритм является более быстроедейственным по сравнению с алгоритмом разработанным автором ранее.*

***Ключевые слова:** проективная инварианта, вершина угла, окрестность Мура, прослеживание контура.*

***Summary.** The vertex detection task is one of the central problems of image processing. This paper presents the modified version of the developed by the author algorithm for vertex detection. It is suggested to use the Moore-neighbor tracing algorithm. This has improved starting and stopping criteria, as well as rules for determining the candidate as the vertex. It was shown that the*

*proposed algorithm takes less time in comparison with the algorithm developed by the author earlier.*

***Key words:** projective invariant, vertex, Moore neighborhood, contour tracing.*

## **1. Введение**

Последние достижения в развитии аппаратного и программного обеспечения сделали возможным практическое использование различных информационных систем, направленных на поддержку принятия решений. Алгоритмы цифровой обработки изображений находят все более широкое применение в научных и прикладных исследованиях в различных областях деятельности человека. Исследователи всего мира уделяют внимание их разработке. В области компьютерного зрения проблема обнаружения характерной черты изображения является одной из фундаментальных, так как является ключевым этапом решения различных задач. В ее качестве выступают геометрические понятия, такие как: вершина угла, кривая, поверхность и другие. Вершина угла представляется координатами одного отсчета изображения. Это позволяет устанавливать соответствие между разными множествами вершин углов с наименьшими вычислительными затратами. Проективное отображение сохраняет вершину угла. Поэтому разработка алгоритма обнаружения вершины угла инвариантного относительно проективного искажения актуальна в научном и практическом плане.

Алгоритм обнаружения вершины угла используются при решении задач: обнаружения и распознавания простого объекта, совмещения изображений и многих других.

**Целью статьи** является улучшение разработанного автором алгоритма обнаружения вершины угла на основе аппроксимации контура бинарного изображения направленными отрезками, для более

быстродейственного выделения. В начале работы показана актуальность решаемой задачи. Далее во втором разделе приводится краткий обзор основных алгоритмов обнаружения вершины угла на изображении. Постановка задачи приведена в разделе 3. В разделах 4-7 описаны разработанные автором алгоритмы: обнаружения вершины угла, определения отсчета бинарного изображения претендентом на вершину угла, кодирования контура претендента на вершину угла и определения претендента вершиной угла. Отметим, что в основе алгоритма кодирования контура претендента на вершину угла (раздел 5) лежит прослеживания конечного контура бинарного изображения на основе анализа окрестности Мура. Результаты тестирования показаны в разделе 8. В 9-ом разделе представлены выводы работы.

## **2. Обнаружение вершины угла на изображении**

Из математики известно, что проективное отображение сохраняет вершину угла, кроме случаев его превращения преобразованием в развернутый или полный углы. Однако известные в литературе алгоритмы обнаружения вершины угла [1–12] не обладают инвариантностью относительно проективного отображения. Отметим ранее разработанный автором алгоритм, выделяющий вершину угла инвариантно относительно проективного отображения [13]. Это достигается за счет выполняемой аппроксимации контура направленными отрезками. Математическая модель контура изображения подробно рассматриваются в работе [14].

## **3. Постановка задачи**

При решении многих ключевых задач цифровой обработки изображений важнейшую роль играет вершина угла. Ее обнаружение отличается сложностью. Проведенный анализ [1–15] показал, что вершина угла  $v$  определяется с помощью некоторого алгоритма обработки исходного изображения, представленного в бинарном виде. В частности,

вершина угла может быть выделена на изображении  $I$  в случае равенства значения первой или второй производной функции  $f$  нулю:  $f'(x) = 0$  или  $f''(x) = 0, x \in \Omega$ .

Под изображением  $I$  будем понимать непрерывное отображение  $f: \Omega \rightarrow \mathbf{R}, \Omega \subset \mathbf{R}^d$ , где  $d$  – это его пространственная размерность, а функция  $f$  имеет компактный носитель.

Вершина угла  $v$  – это точка  $O$ , откуда берут начало два и более луча или вектора; где сходятся два и более отрезка; где две и более прямых, лучей или отрезков пересекаются.

**Задача 1.** Пусть нам дано изображение  $I$ . Необходимо обнаружить и локализовать вершину угла  $v (V\{v_i\})$  исходного изображения.

Трудность обнаружения вершины угла  $v$  на изображении  $I$  обусловлена сложностью разработки алгоритма обнаружения точки  $O$ , с учетом требования равенства значения первой или второй производной функции  $f$  нулю. Поскольку функция  $f$  известна только в дискретных отсчетах, то невозможно вычислить ее производную до тех пор, пока она не будет определена. При этом приближение ее производной можно определить лишь с ограниченной точностью. На практике используются различные маски для численного приближения производной функции  $f$ , в частности, дифференциальный оператор: Превитта [15], Собеля [15], Робертса [16] и др.

При обнаружении вершины  $v$  угла на изображении  $I$  возникают ошибки двух родов. Ошибка первого рода заключается в отклонении основной гипотезы, в то время как она справедлива, т. е. вершина угла  $v$  объекта изображения  $I$  не была обнаружена.

Ошибка второго рода заключается в принятии основной гипотезы, в то время как она не верна, т. е. произошло ошибочное обнаружение вершины угла  $v$  объекта изображения  $I$ . Эти ошибки оцениваются на основе коэффициентов  $k_1$  и  $k_2$ :  $k_1 = N/R$  и  $k_2 = L/R$ , где  $N$  – количество

необнаруженных вершин углов;  $L$  – количество определенных ложных вершин углов;  $R$  – общее число вершин углов изображения  $I$ ,  $N$ ,  $L$  и  $R \in \mathbb{N}$ .

#### **4. Алгоритм обнаружения вершины угла на основе аппроксимации контура бинарного изображения**

Рассматривается модификация разработанного автором алгоритма обнаружения вершины угла  $v$ . Выполняемая аппроксимация контура бинарного изображения направленными отрезками позволяет достигать инвариантность относительно проективного искажения. Его отличительной особенностью является применение алгоритма прослеживания контура на основе анализа окрестности Мура [17-18].

Окрестность отсчета  $(i, j)$  изображения  $I$  или окно обработки  $w$  – это множество его отсчетов, содержащее сам отсчет  $(i, j)$ , и близкие (в каком-либо смысле) к нему отсчеты исходного изображения.

Алгоритм обнаружения вершины угла на основе аппроксимации контура бинарного изображения направленными отрезками (Аг. 1) включает следующие шаги:

*Шаг 1.* Выделить контур как границу объекта изображения  $I$ ;

*Шаг 2.* Определить претендент  $r$  на вершину угла  $v$ ;

*Шаг 3.* Сформировать код контура претендента  $r$  на вершину угла  $v$ ;

*Шаг 4.* Если код контур корректен, то определить является ли претендент  $r$  вершиной угла  $v$ . Иначе алгоритм завершается.

Рассмотрим эти шаги подробнее.

Одним из возможных путей выделения контура на изображении  $I$  является применение алгоритма обнаружения края Кэнни [19] к исходному изображению. Ширина контура равняется одному отсчету изображения. Подробно задача выделения контура как границы объекта изображения рассмотрена в работах [14; 19-20].

Разработанные автором алгоритмы определения отсчета бинарного изображения претендентом  $r$  на вершину угла и кодирования контура претендента  $r$  на вершину угла, лежащие в основе второго и третьего шага, подробно рассматриваются ниже.

Определение претендента  $r$  вершиной угла  $v$  выполняется, на четвертом шаге работы алгоритма Ag. 1, посредством разработанных автором правил. В их основе лежит аппроксимации полученного на предыдущем шаге кода контура направленными отрезками.

Алгоритма Ag. 1 опирается в своей работе на следующие параметры:

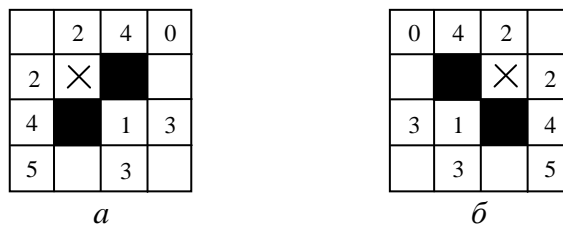
- $n$  – длина контуров  $\Gamma_a$  и  $\Gamma_b$ ;
- $wk$  – количество допустимых отличных от рассматриваемых направлений в коде конечного контура  $\Gamma_a$  или  $\Gamma_b$ ;
- $mx$  и  $my$  – ширина и высота окрестности вершины угла, в которой может находиться только она одна.

Временная сложность алгоритма Ag. 1 равна  $O(n)$ .

## **5. Алгоритм определения отсчета бинарного изображения претендентом на вершину угла**

В основе рассматриваемого алгоритма лежит анализ контура бинарного изображения, выполняемый в окне обработке  $w$  при сканировании исходного изображения  $I$  представленного в бинарном виде в порядке построчной развертки – слева направо в строке и сверху вниз по строкам.

В случае выполнения нескольких условий отсчет  $(i, j)$  бинарного изображения относится к одному из двух типов претендентов  $r$  на вершину угла  $v$ : претендент первого  $r_1$  или второго  $r_2$  типа (рис. 1). В их основе лежит дифференциальный оператор Робертса [16]. Здесь,  $i$  – это номер столбца,  $j$  – это номер строки.



**Рис. 1. Схемы определения произвольного отсчета бинарного изображения претендентом  $r$  на вершину угла  $\nu$ : а) первый тип  $r_1$ ; б) второй тип  $r_2$**

На рис. 1 рассматриваемый отсчет  $(i, j)$  бинарного изображения обозначен крестиком, а также отсчеты  $c$ :  $(i+1, j)$  и  $(i, j+1)$  или  $(i-1, j)$  и  $(i, j+1)$ , образующие вершину угла  $\nu$  отмечены черными квадратами. Необходимым условием определения отсчета  $(i, j)$  претендентом  $r$  на вершину угла  $\nu$  является равенство значений отсчетов  $c$  единице. Достаточное условие есть выполнение дополнительных критериев, в частности, значения отсчетов, обозначенные нулем и пятеркой на рис. 1, не равны единице одновременно. Выполняемый анализ обусловлен присущими изображению различного рода шумами, например, эффектом ложного контура [21].

Отсчеты  $c$  претендента  $r$  на вершину угла  $\nu$  могут образовывать произвольный угол, за исключением развернутого и полного углов. Пусть

$$\text{отсчет } c_0 = (i, j+1), \text{ а отсчет } c_1 = \begin{cases} (i+1, j), r_1 \\ (i-1, j), r_2 \end{cases}$$

Алгоритм определения произвольного отсчета бинарного изображения претендентом первого или второго типа на вершину угла (Ag. 2) состоит из следующих шагов:

*Шаг 1.* В случае если значение отсчета  $(i, j+1)$ , расположенного по отношению к текущему отсчету  $(i, j)$  сканирования, равняется единице. Тогда, в первом случае, если значение отсчета  $(i+1, j)$  равно единице, то обнаружены отсчеты  $c$  претендента первого типа  $r_1$ . Во втором случае, если значение отсчета  $(i-1, j)$  равняется единице, то обнаружены отсчеты  $c$  претендента второго типа  $r_2$ . В противных случаях алгоритм завершается;

*Шаг 2.* Выполняется в случае, если на шаге 1 были обнаружены отсчеты  $c$  претендента первого типа  $r_1$ . Тогда, если значение одного из отсчетов  $(i+2, j-1)$  или  $(i-1, j+2)$  равняется нулю и значения отсчетов  $(i-1, j)$  и  $(i, j-1)$  или  $(i+1, j+2)$  и  $(i+2, j+1)$  не равняются единице одновременно. При этом отсутствуют отсчеты подобные текущим отсчетам  $c$  ( $(i+1, j)$  и  $(i, j+1)$ ), расположенные по отношению к ним: слева ( $(i, j)$  и  $(i-1, j+1)$ ) и справа ( $(i+2, j)$  и  $(i+1, j+1)$ ) или снизу ( $(i, j+2)$  и  $(i+1, j+1)$ ) и сверху ( $(i, j)$  и  $(i+1, j-1)$ ). Кроме того, значение одного из отсчетов  $(i+1, j-1)$ ,  $(i-1, j+1)$  или  $(i-1, j+2)$  не равно единице. В этом случае, рассматриваемый отсчет  $(i, j)$  является претендентом первого типа  $r_1$  на вершину угла  $v$ . Алгоритм завершается;

*Шаг 3.* Выполняется в случае, если на шаге 1 были обнаружены отсчеты  $c$  претендента второго типа  $r_2$ . Тогда, если значение одного из отсчетов  $(i-2, j-1)$  или  $(i+1, j+2)$  равняется нулю и значения отсчетов  $(i+1, j)$  и  $(i, j-1)$  или  $(i-1, j+2)$  и  $(i-2, j+1)$  не равняются единице одновременно. При этом отсутствуют отсчеты подобные текущим отсчетам  $c$  ( $(i-1, j)$  и  $(i, j+1)$ ), расположенные по отношению к ним: слева ( $(i-2, j)$  и  $(i-1, j-1)$ ) и справа ( $(i, j)$  и  $(i+1, j+1)$ ) или снизу ( $(i, j+2)$  и  $(i-1, j+1)$ ) и сверху ( $(i, j)$  и  $(i-1, j-1)$ ). Кроме того, значение одного из отсчетов  $(i-1, j-1)$ ,  $(i+1, j+1)$  или  $(i+1, j+2)$  не равно единице. В этом случае, рассматриваемый отсчет  $(i, j)$  является претендентом второго типа  $r_2$  на вершину угла  $v$ . Алгоритм завершается.

Временная сложность алгоритма Ag. 2 равна  $O(1)$ .

## **6. Алгоритм кодирования контура претендента на вершину угла**

Контур  $\Gamma$  претендента  $r$  на вершину угла  $v$  подразделяется на две части: конечный контур  $\Gamma_a$  и конечный контур  $\Gamma_b$ . Начальным отсчетом  $a_0$  конечного контура  $\Gamma_a$  является отсчет  $c_1$ ,  $a_0 = c_1$ , а начальным отсчетом  $a_0$  конечного контура  $\Gamma_b$  является отсчет  $c_0$ ,  $a_0 = c_0$  (рис. 2).



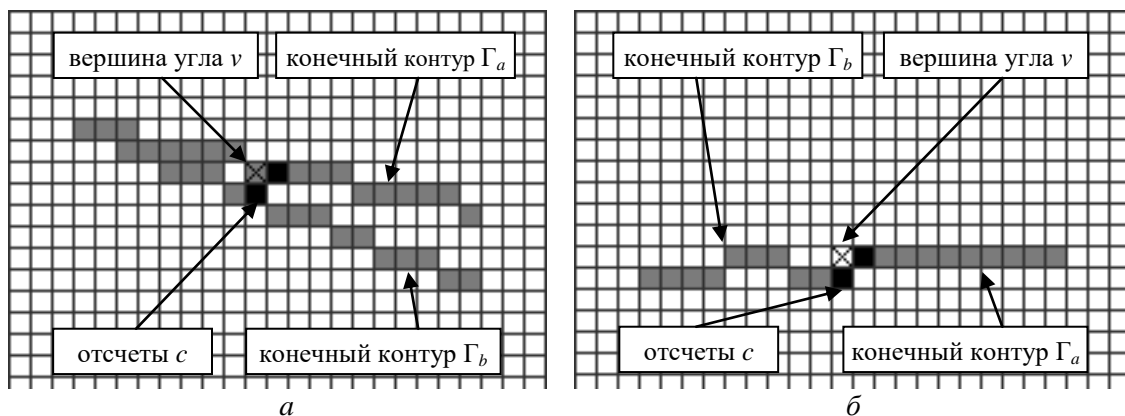


Рис. 2. Контур бинарного изображения: а) контур 1; б) контур 2

Длины конечных контуров  $\Gamma_a$  и  $\Gamma_b$  полагаются равными и не превышают заданное число элементов  $n$ . Их кодирование осуществляется стандартными ЭВ [20] согласно предложенному способу (рис. 3) на основе 8-связного цепного кода Фримана [20-21].

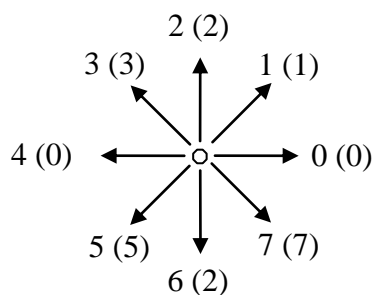


Рис. 3. Нумерация и кодирование направлений 8-связного цепного кода Фримана

На рис. 3 код для каждого из направлений 8-связного цепного кода Фримана указан в скобках.

В литературе известно большое количество алгоритмов прослеживания контура бинарного изображения [17-18; 20–22], в частности, на основе анализа окрестности Мура [17-18], Розенфельда [20], Павлидиса [22]. Каждый, из них имеет свои достоинства и недостатки. Целью его применения, является прослеживание конечных контуров  $\Gamma_a$  и  $\Gamma_b$ , где длина каждого из которых не превосходит  $n$  элементов (в частности, 8). Поэтому в первом приближении нам подойдет любой из отмеченных алгоритмов. Однако среди них выделим алгоритм на основе анализа окрестности Мура [17; 18], позволяющий выполнять обнаружение

следующего элемента контура на основе осмотра отсчетов окрестности его текущего элемента, как по часовой стрелке, так и против нее. При этом возможно определить начальное и конечное направления осмотра. Все это хорошо дополняет алгоритм Ag. 2.

Алгоритм кодирования контура претендента на вершину угла (Ag. 3) включает следующие шаги:

*Шаг 1.* Задать отсчет  $a_0$  конечных контуров  $\Gamma_a$  и  $\Gamma_b$ , а также начальное и конечное направления осмотра обнаружения следующего элемента контура  $a_1$ ;

*Шаг 2.* Выполнить прослеживание и кодирование конечного контура  $\Gamma_a$ ;

*Шаг 3.* Если значение размера кода контура  $\Delta\Gamma_a$  меньше  $n$ . Тогда в случае если рассмотрены все возможные начальные направления осмотра, то полученный код удаляем и алгоритм завершается, в противном случае задаем значение нового начального направления (следующие по порядку осмотра) и продолжаем с шага 2. Иначе выполняем шаг 4.

*Шаг 4.* Выполнить прослеживание и кодирование конечного контура  $\Gamma_b$ ;

*Шаг 5.* Если значение размера кода контура  $\Delta\Gamma_b$  меньше  $n$ . Тогда в случае если рассмотрены все возможные начальные направления осмотра, то полученный код удаляем и алгоритм завершается, в противном случае задаем значение нового начального направления (следующие по порядку осмотра) и выполняем шаг 4. Иначе алгоритм завершается.

Рассмотрим эти шаги подробнее.

Прослеживание контура выполняется на основе алгоритма анализа окрестности Мура [17]. Кодирование контура выполняется на основе описанного выше способа (см. рис. 3). Правила начала и завершения прослеживания контура бинарного изображения подробно представлены ниже.

*Правила начала прослеживания контура бинарного изображения:*

- для претендента  $r$  на вершину угла  $v$  первым элементом  $a_0$  конечного контура  $\Gamma_a$ , является отсчет  $c_1$ , а в случае конечного контура  $\Gamma_b$  отсчет  $c_0$ ;
- для конечных контуров  $\Gamma_a$  и  $\Gamma_b$  начальным направлением осмотра обнаружения следующего элемента контура  $a_1$  является отсчет, обозначенный на рис. 1 единицей. В случае претендента  $r_1$  отсчет  $(i+1, j+1)$ , а для претендента  $r_2$  отсчет  $(i-1, j+1)$ . Конечным направлением осмотра является отсчет  $(i, j)$ ;
- для претендента  $r_1$  на вершину угла  $v$  обнаружение отсчета  $a_1$  конечного контура  $\Gamma_a$  выполняется, на основе осмотра против часовой стрелки, а обнаружение отсчета  $a_1$  конечного контура  $\Gamma_b$  выполняется, на основе осмотра по часовой стрелке;
- для претендента  $r_2$  на вершину угла  $v$  обнаружение отсчета  $a_1$  конечного контура  $\Gamma_a$  выполняется, на основе осмотра по часовой стрелке, а обнаружение отсчета  $a_1$  конечного контура  $\Gamma_b$  выполняется, на основе осмотра против часовой стрелки;
- для претендента  $r_1$  на вершину угла  $v$ , если значения отсчетов  $(i+2, j+1)$  и  $(i+1, j+2)$  равны единице (см. рис. 1, а, обозначены тройками). Тогда начальным направлением осмотра обнаружения элемента  $a_1$  конечного контура  $\Gamma_a$  является отсчет  $(i+2, j+1)$ , а в случае конечного контура  $\Gamma_b$  отсчет  $(i+1, j+2)$ ;
- для претендента  $r_1$  на вершину угла  $v$ , если значения отсчетов  $(i-1, j)$  и  $(i, j-1)$  равны единице (см. рис. 1, а, обозначены двойками). Тогда конечным направлением осмотра обнаружения элемента  $a_1$  конечного контура  $\Gamma_a$  является отсчет  $(i, j-1)$ , а в случае конечного контура  $\Gamma_b$  отсчет  $(i-1, j)$ ;
- для претендента  $r_2$  на вершину угла  $v$ , если значения отсчетов  $(i-2, j+1)$  и  $(i-1, j+2)$  равны единице (см. рис. 1, б, обозначены

тройками). Тогда начальным направлением осмотра обнаружения элемента  $a_1$  конечного контура  $\Gamma_a$  является отсчет  $(i-2, j+1)$ , а в случае конечного контура  $\Gamma_b$  отсчет  $(i-1, j+2)$ ;

- для претендента  $r_2$  на вершину угла  $\nu$ , если значения отсчетов  $(i+1, j)$  и  $(i, j-1)$  равны единице (см. рис. 1, б, обозначены двойками). Тогда конечным направлением осмотра обнаружения элемента  $a_1$  конечного контура  $\Gamma_a$  является отсчет  $(i, j-1)$ , а в случае конечного контура  $\Gamma_b$  отсчет  $(i+1, j)$ .

*Правила завершения прослеживания контура бинарного изображения:*

- встретился один из отсчетов  $c$ ;
- рассмотрены все возможные направления осмотра;
- если обнаруженный элемент  $a_i$  контура встретился второй раз;
- прослежено  $n$  отсчетов контура.

Временная сложность алгоритма Ag. 3 равна  $O(n)$ .

## **7. Алгоритм определения претендента вершиной угла**

Сопоставление претендента  $r$  вершине угла  $\nu$  выполняется посредством разработанных автором правил, подразделяющихся на две категории: соответствия претендента ложной вершине угла и соответствия претендента вершине угла. В их основе лежит аппроксимация кода контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  направленными отрезками.

Алгоритм определения претендента вершиной угла (Ag. 4) состоит из следующих шагов:

*Шаг 1.* Выполнить правила соответствия претендента ложной вершине угла. В случае выполнения любого из них считаем, что рассматриваемый претендент  $r$  не является вершиной угла  $\nu$  и алгоритм завершается;

*Шаг 2.* Выполнить правила соответствия претендента вершине угла. В случае выполнения любого из них считаем, что рассматриваемый претендент  $r$  является вершиной угла  $v$  и алгоритм завершается.

Пусть значения кодов следующих направлений 8-связного цепного кода Фримана: 1, 3, 5 и 7 (см. рис. 3), формируют множество  $C = \{1, 3, 5, 7\}$ .

*Правила соответствия претендента ложной вершине угла:*

- претендент  $r$  не является вершиной угла  $v$ , в случае если первые два элемента кода контура  $\Delta\Gamma_a$  или  $\Delta\Gamma_b$  заданы направлением с кодом 0 (2), а последние два элемента кода заданы направлением с кодом 2 (0). При этом код оставшегося контура не состоит лишь из направления с кодом 0 или 2. Допускается присутствие любого другого направления в коде контура не более  $wk$  раз. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{0; 7; 0; 7; 0; 2; 2; 2\}$  и  $\Delta\Gamma_b = \{2; 2; 7; 0; 2; 7; 0; 0\}$ ;
- претендент  $r$  не является вершиной угла  $v$ , в случае если значение первого элемента кода контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  равно 0 или 2 и значение количества оставшегося направления (0 или 2), следующего друг за другом, более единицы. При этом значение второго элемента кода контура соответствует одному из направлений множества  $C$ . В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{2; 1; 0; 0; 0; 0; 0; 0\}$ ;
- претендент  $r$  не является вершиной угла  $v$ , в случае если контуры  $\Gamma_a$  и  $\Gamma_b$  аппроксимируются прямой линией. Коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами:  $\{0\} - \{0\}$ ,  $\{2\} - \{2\}$ ,  $\{1\} - \{5\}$ ,  $\{3\} - \{7\}$ ,  $\{5\} - \{1\}$  или  $\{7\} - \{3\}$ . Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{1; 1; 1; 1; 1; 1; 1; 1\}$  и  $\Delta\Gamma_b = \{5; 5; 5; 5; 5; 5; 5; 5\}$ ;

- претендент  $r$  не является вершиной угла  $v$ , в случае если код контура  $\Delta\Gamma_a$  или  $\Delta\Gamma_b$  задан направлениями с кодами  $\{0, 2\}$  и код каждого из них присутствует более одного раза, а также значения их количества равны. При этом значения количества одинаковых направлений, следующих друг за другом, с кодом 0 и 2 равны нулю. Например, для претендента  $r_1$ :  $\Delta\Gamma_a = \{0; 2; 0; 2; 0; 2; 0; 2\}$ ;
- претендент  $r$  не является вершиной угла  $v$ , в случае если коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами  $\{0, 2\}$ . При этом модуль разницы значений количества одного из направлений 0 или 2 в коде контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  меньше либо равен единице и модуль разницы значений количества оставшегося направления, следующего друг за другом, более единицы. Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_2$ :  $\Delta\Gamma_a = \{0; 0; 0; 0; 2; 0; 3; 0\}$  и  $\Delta\Gamma_b = \{0; 0; 0; 0; 7; 0; 0; 0\}$ ;
- претендент  $r$  не является вершиной угла  $v$ , в случае если коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами  $\{0, 2\}$ . При этом в коде любого из контуров значение количества направления с кодом 0 или 2 больше или равно  $n - wk$  и значение количества данного направления, следующего друг за другом, в коде другого контура больше единицы. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{2; 2; 2; 0; 2; 0; 2; 0\}$  и  $\Delta\Gamma_b = \{2; 2; 2; 2; 2; 0; 2; 2\}$ ;
- претендент  $r$  не является вершиной угла  $v$ , в случае если код контура  $\Delta\Gamma_a$  или  $\Delta\Gamma_b$  задан направлениями с кодами  $\{0, 2\}$  и модуль разницы значений количества одинаковых направлений в коде каждого из контуров меньше либо равен единице. При этом значение количества направления с кодом 0 (2) в коде каждого контура больше  $n/2$  и наличие кода направления 2 (0) в коде одного из контуров

обязательно. В частности, для претендента  $r_2$ :  $\Delta\Gamma_a = \{0; 0; 2; 0; 0; 2; 0; 0\}$  и  $\Delta\Gamma_b = \{0; 0; 7; 0; 7; 0; 2; 0\}$ ;

- претендент  $r$  не является вершиной угла  $v$ , в случае если код контура  $\Delta\Gamma_a$  или  $\Delta\Gamma_b$  задан направлениями с кодами  $\{0, 2\}$  и каждое из направлений присутствует в коде более одного раза и их количества не равны. При этом в случае, если в коде контура присутствует больше направлений, в частности, с кодом 0, то значение количества направлений с этим же кодом (0) в другом контуре больше или равно половине количества элементов кода контура. То же верно и для направления с кодом 2. В частности, для претендента  $r_2$ :  $\Delta\Gamma_a = \{0; 3; 0; 3; 0; 0; 3; 0\}$  и  $\Delta\Gamma_b = \{0; 2; 0; 0; 2; 0; 0; 2\}$ ;
- претендент  $r$  не является вершиной угла  $v$ , в случае если контуры  $\Gamma_a$  и  $\Gamma_b$  аппроксимируются прямой линией. Коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами:  $\{0, 1\} - \{0, 5\}$ ,  $\{0, 5\} - \{0, 1\}$ ,  $\{0, 3\} - \{0, 7\}$ ,  $\{0, 7\} - \{0, 3\}$ ,  $\{1, 2\} - \{2, 5\}$ ,  $\{2, 5\} - \{1, 2\}$ ,  $\{2, 3\} - \{2, 7\}$  или  $\{2, 7\} - \{2, 3\}$ , наличие значения кода направления из множества  $S$  обязательно. При этом модуль разницы значений количества направления из множества  $S$  в коде контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  меньше либо равен единице. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{2; 1; 1; 2; 1; 1; 2; 1\}$  и  $\Delta\Gamma_b = \{5; 2; 5; 5; 2; 5; 5; 2\}$ ;
- претендент  $r$  не является вершиной угла  $v$ , в случае если контуры  $\Gamma_a$  и  $\Gamma_b$  аппроксимируются прямой линией. Коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами:  $\{0, 1, 2\} - \{0, 2, 5\}$ ,  $\{0, 2, 5\} - \{0, 1, 2\}$ ,  $\{0, 2, 3\} - \{0, 2, 7\}$  или  $\{0, 2, 7\} - \{0, 2, 3\}$ , наличие значения кода каждого из направлений обязательно. При этом модуль разницы значений количества направления с кодом 0 или 2 в коде контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  меньше либо равен единице и модуль разницы значений количества направления с кодом 0 или 2, следующего друг за другом, в коде каждого из контуров меньше

либо равен единице. Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{0; 0; 0; 2; 0; 0; 1; 0\}$  и  $\Delta\Gamma_b = \{0; 0; 2; 0; 0; 5; 0; 2\}$ ;

- претендент  $r$  не является вершиной угла  $v$ , в случае если контуры  $\Gamma_a$  и  $\Gamma_b$  аппроксимируются прямой линией. Коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами:  $\{0, 1, 2\} - \{0, 2, 5\}$ ,  $\{0, 2, 5\} - \{0, 1, 2\}$ ,  $\{0, 2, 3\} - \{0, 2, 7\}$  или  $\{0, 2, 7\} - \{0, 2, 3\}$ , наличие кода направления из множества  $C$  обязательно, а также модуль разницы значений количества направлений из множества  $C$  в коде контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  меньше либо равен единице. При этом модуль разницы значений количества направления с кодом 0 или 2 в коде контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  меньше либо равен единице и модуль разницы значений количества направления с кодом 0 или 2, следующего друг за другом, в коде каждого из контуров меньше либо равен единице. Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_2$ :  $\Delta\Gamma_a = \{3; 3; 2; 3; 3; 3; 2; 0\}$  и  $\Delta\Gamma_b = \{2; 7; 7; 7; 2; 7; 7; 7\}$ ;

- претендент  $r$  не является вершиной угла  $v$ , в случае если контуры  $\Gamma_a$  и  $\Gamma_b$  аппроксимируются прямой линией. Коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами:  $\{0, 1, 2\} - \{0, 2, 3\}$ ,  $\{0, 2, 3\} - \{0, 1, 2\}$ ,  $\{0, 2, 3\} - \{0, 2, 5\}$   $\{0, 2, 5\} - \{0, 2, 3\}$ ,  $\{0, 2, 7\} - \{0, 2, 5\}$ ,  $\{0, 2, 5\} - \{0, 2, 7\}$ ,  $\{0, 2, 7\} - \{0, 1, 2\}$  или  $\{0, 1, 2\} - \{0, 2, 7\}$ , наличие кода каждого из направлений обязательно. При этом модуль разницы значений количества направления с кодом 0 или 2 в коде контуров меньше либо равен единице, а также модуль разницы значений количества направления из множества  $C$  в каждом из контуров меньше либо равен единице и в одном из контуров



значение количества кода направления из множества  $S$  равно единице. Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{0; 0; 1; 2; 0; 2; 0; 2\}$  и  $\Delta\Gamma_b = \{2; 0; 2; 7; 0; 2; 0; 0\}$ ;

- претендент  $r$  не является вершиной угла  $v$ , в случае если контуры  $\Gamma_a$  и  $\Gamma_b$  аппроксимируются прямой линией. Коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами:  $\{0, 7\} - \{0, 2, 3\}$ ,  $\{2, 7\} - \{0, 2, 3\}$ ,  $\{0, 2, 3\} - \{0, 7\}$ ,  $\{0, 2, 3\} - \{2, 7\}$ ,  $\{0, 3\} - \{0, 2, 7\}$ ,  $\{2, 3\} - \{0, 2, 7\}$ ,  $\{0, 2, 7\} - \{0, 3\}$ ,  $\{0, 2, 7\} - \{2, 3\}$ ,  $\{0, 5\} - \{0, 1, 2\}$ ,  $\{2, 5\} - \{0, 1, 2\}$ ,  $\{0, 1, 2\} - \{0, 5\}$ ,  $\{0, 1, 2\} - \{0, 1\}$ ,  $\{0, 1\} - \{0, 2, 5\}$ ,  $\{1, 2\} - \{0, 2, 5\}$ ,  $\{0, 1\} - \{0, 2, 5\}$  или  $\{1, 2\} - \{0, 2, 5\}$ . При этом для контура код, которого содержит наименьшее число направлений (мощность его множества не превзойдет 2), соответствующие направления встречаются более одного раза. Для оставшегося контура код направления из множества  $S$  и направления с одинаковыми кодами (0 или 2) в коде каждого из контуров встречаются более одного раза или модуль разницы количества значения направления с одинаковым кодом (0 или 2) в коде каждого из контуров меньше либо равно единицы и модуль разницы количества значения направления с одинаковым кодом (0 или 2), следующих друг за другом, в коде каждого из контуров меньше либо равно единицы. Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_2$ :  $\Delta\Gamma_a = \{0; 3; 0; 0; 2; 0; 0; 2\}$  и  $\Delta\Gamma_b = \{0; 0; 7; 0; 7; 0; 0; 7\}$ ;
- претендент  $r$  не является вершиной угла  $v$ , в случае если в коде контура  $\Delta\Gamma_a$  или  $\Delta\Gamma_b$  присутствуют коды направлений:  $\{1, 3\}$ ,  $\{3, 5\}$ ,  $\{5, 7\}$  или  $\{7, 1\}$  и значение количества в коде одного из направлений более единицы. Допускается присутствие любого

другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{0; 7; 2; 5; 2; 5; 2; 5\}$ ;

- претендент  $r$  не является вершиной угла  $v$ , в случае если в коде контура  $\Delta\Gamma_a$  или  $\Delta\Gamma_b$  присутствуют коды направлений:  $\{1, 3\}$ ,  $\{3, 5\}$ ,  $\{5, 7\}$  или  $\{7, 1\}$ . При этом код этого контура состоит только из отмеченных направлений плюс кода 0 или 2. Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_1$ :  $\Delta\Gamma_b = \{2; 5; 2; 2; 2; 7; 2; 2\}$ .

*Правила соответствия претендента вершине угла:*

- претендент  $r$  является вершиной угла  $v$ , в случае если код контура  $\Delta\Gamma_a$  задан нулями, а код контура  $\Delta\Gamma_b$  задан двойками или наоборот. Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{0; 0; 0; 0; 0; 0; 0; 0\}$  и  $\Delta\Gamma_b = \{2; 2; 2; 5; 2; 2; 2; 2\}$ ;
- претендент  $r$  является вершиной угла  $v$ , в случае если коды контуров  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  заданы направлениями с кодами  $\{0, 2\}$  и кодом одного из направлений множества  $S$ . Допускается присутствие любого другого направления в коде контура  $\Delta\Gamma_a$  и  $\Delta\Gamma_b$  не более  $wk$  раз. В частности, для претендента  $r_1$ :  $\Delta\Gamma_a = \{0; 0; 2; 1; 0; 1; 0; 0\}$  и  $\Delta\Gamma_b = \{0; 3; 2; 3; 2; 3; 2; 2\}$ .

Временная сложность алгоритма Ag. 4 равна  $O(1)$ .

## **8. Результаты тестирования**

Тестирование алгоритмов Ag. 1 и алгоритма [13] выполнялось со следующими значениями параметров:  $n = 8$ ,  $wk = 1$  и  $mx = my = 3$ . Выделение контура изображения осуществлялось на основе алгоритма обнаружения края Кэнни [19], использующегося со значениями порогов:  $T_1$

$= 0$  и  $T_2 = 0,35$ . Ширина контура равняется одному отсчету изображения. Использовалась база из 100 различных реальных изображений.

Тесты показали, что представленный в данной работе алгоритм Ag. 1 сопоставим по точности обнаружения с ранее разработанным автором алгоритмом [13]. Для обоих алгоритмов  $k_1 = 0,07$  и  $k_2 = 0,09$ . Однако отличается меньшим временем обработки. Произошло увеличение быстродействия в среднем в 2,1 раза.

Алгоритм Ag. 1 находит свое практическое применение при решении задачи обнаружения и распознавания простого объекта на изображениях, а также в задачах дистанционного зондирования [23–25].

**Заключение.** Предложена модификация разработанного автором алгоритма обнаружения вершины угла на изображении, состоящая в применении алгоритма прослеживания контура на основе анализа окрестности Мура. Это позволило упростить критерии обнаружения вершины угла. В частности, правила начала и завершения прослеживания контура. Проведенное тестирование показало, что разработанный автором алгоритм обнаруживает вершину угла с меньшим временем обработки по сравнению с алгоритмом разработанным автором ранее.

**Источник финансирования исследования.** Данное исследование было самофинансировано на средства самого автора.

### **Литература**

1. Moravec H.P. Towards automatic visual obstacle avoidance / In Proc. of the 5<sup>th</sup> Intern. Joint Conf. of Artificial Intelligence. Cambridge, 1977. - P. 587–598.
2. Harris C., Stephens M. A combined corner and edge detector / In Proceedings of the Fourth Alvey Vision Conference. - 1988. - P. 147–152.

3. Shi F., Huang X., Duan Y. Robust Harris-Laplace Detector by Scale Multiplication. *Advances in Visual Computing*. Berlin: Springer, 2009. - P. 265–274.
4. Trajkovic M., Hedley M. Fast corner detection. *Image and vision computing*. – 1998. - Vol. 16. - P. 75 – 87.
5. He X.C., Yung N.H.C. Corner detector based on global and local curvature properties. *Optical Engineering*. 2008. Vol. 47, № 5. P. 057008-1 – 057008-12.
6. Awrangjeb M., Lu G. Robust Image Corner Detection based on the Chord-to-Point Distance Accumulation Technique // *IEEE Transaction on Multimedia*. 2008. Vol. 10, Issue 6. P. 1059–1072.
7. Rosten E., Porter R., Drummond T. Faster and better: a machine learning approach to corner detection // *IEEE Transaction on Pattern Analysis and Machine Intelligence*. 2008. Vol. 32, № 1. P. 105–119.
8. Quinlan J.R. Induction of decision trees // *Machine Learning*. 1986. Vol. 1. P. 81–106.
9. Kahaki S.M.M., Nordin M.J., Ashtari A.H. Contour-Based Corner Detection and Classification by Using Mean Projection Transform // *Sensors*. 2014. Vol. 14. P. 4126–4143.
10. Verkeenko M.S. Development of an algorithm for fast corner points detection // *Journal of Computer and Systems Sciences International*. 2014. Vol. 53, № 3. P. 392 – 401.
11. Lv G.L., Hou Z.J., Zhao H.Y. Research for the Square Corner Detection Algorithm Based on Electronic Measurement Engineering // *Advances in Mechanical and Electronic Engineering*. 2012. Vol. 177. P. 633–638.
12. Feltes M., [et al.]. Improved Contour-Based Corner Detection for Architectural Floor Plans // *Graphics Recognition. Current Trends and Challenges*. 2014. P. 191–203.

13. Козловский А.Н. Алгоритм обнаружения вершины угла на изображении на основе аппроксимации контура бинарного изображения // Международный научный журнал. 2016. № 9. С. 63–73. DOI:10.21267/IN.2016.9.3288.
14. Kazlouski A.M. Contour analysis-based mathematical model of a particular image object // International Journal of Research in Engineering, IT and Social Sciences. 2018. Vol. 8, Issue 9, P. 112–114.
15. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. М.: Техносфера, 2006. 616 с.
16. Робертс Л. Автоматическое восприятие трехмерных объектов. Интегральные роботы. М.: Мир, 1973. 162–208 с.
17. Pradhan R., Kumar S., Agarwal R., Pradhan M.P. and Ghose M.K. Contour line tracing algorithm for digital topographic maps // International Journal of Image Processing. 2010. Vol. 4, no. 2, P. 156–163.
18. Reddy P.R., Amarnadh V., Bhaskar M. Evaluation of stopping criterion in contour tracing algorithms // International Journal of Computer Science and Information Technologies. 2012. Vol. 3, P. 3888–3894.
19. Canny J. Computational Approach to Edge Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986. Vol. 8, № 6. P. 679–698.
20. Фурман Я.А. Введение в контурный анализ: приложения к обработке изображений и сигналов. М.: Физматлит, 2003. 588 с.
21. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2006. 1072 с.
22. Pavlidis T. Algorithms for Graphics and Image Processing. Springer-Verlag: Berlin, Germany, 2012.
23. Козловский А.Н. Алгоритмы обнаружения и распознавания простого объекта на изображениях // Эффективные исследования современности. Сборник научных работ X Международной научной

конференции Евразийского Научного Объединения. – Москва: ЕНО, октябрь 2015. Часть 1. С. 58–61.

24. Козловский А.Н. Алгоритм распознавания простого объекта изображения на основе стохастической геометрии // Международный научный журнал. 2016. № 11, С. 70-73. DOI:10.21267/IN.2016.11.3860.
25. Kazlouski A.M. Landmark-based image registration using a plain object model in remote sensing tasks // International scientific journal “Internauka”. 2018. No. 5 (45). P. 24-26.