

Козловский Антон Николаевич*магистр технических наук***Kazlouski A. M.***Master of Engineering Science*

АЛГОРИТМ ОБНАРУЖЕНИЯ ВЕРШИНЫ УГЛА НА ИЗОБРАЖЕНИИ НА ОСНОВЕ АППРОКСИМАЦИИ КОНТУРА БИНАРНОГО ИЗОБРАЖЕНИЯ

VERTEX DETECTION ALGORITHM IN IMAGE PROCESSING BASED O N BINARY IMAGE CONTOUR APPROXIMATION

Аннотация. Задача обнаружения характерной черты изображения является одной из центральных проблем цифровой обработки изображений. В статье рассматривается улучшение разработанного автором алгоритма обнаружения вершины угла объекта изображения на основе аппроксимации контура бинарного изображения векторами. В частности, представлены новые правила аппроксимации контура бинарного изображения направленными отрезками, для более точного и быстро действенного обнаружения вершины угла. Приводится краткий обзор основных алгоритмов обнаружения характерной черты изображения. Дается сравнительная оценка этих алгоритмов.

Ключевые слова: сегментация изображения, контурный анализ, объект изображения, характерная черта изображения, вершина угла.

Summary. The salient image feature detection task is one of the central problem of image processing. This paper presents the improvement of the developed by the author the vertex detection algorithm in image processing based on approximation of binary image contour by vector. For example, new rules on approximation with given accuracy of binary image contour by directed segments were developed to more accurately and quickly effective detection of the vertex. Brief emphasis on main image salient feature detection algorithms provided. The comparative evaluation of these algorithms carried out.

Keywords: image segmentation, contour analysis, image object, salient image feature, the vertex.

Введение

В последние десятилетия значительный прогресс в развитии аппаратного и программного обеспечения, а также методик технического зрения сделал 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. Одной из важнейших задач цифровой обработки изображений является разработка алгоритма обнаружения характерной черты изображения, так как ее выделение является ключевым этапом решения различных задач. Поэтому разработка алгоритма обнаружения характерной черты изображения актуальна в научном и практическом плане.

Алгоритмы обнаружения характерной черты изображения используются при решении задачи обнаружения и распознавания объекта на изображении, совмещения изображений и многих других.

В качестве характерной черты изображения выступают отсчет, кривая или объект. В литературе из-

вестно большое количество алгоритмов обнаружения характерной черты изображения на основе анализа окрестности его отсчета [1–21]. Их общим недостатком является отсутствие инвариантности относительно проективного отображения.

Целью статьи является улучшение разработанного автором алгоритма обнаружения вершины угла на изображении на основе аппроксимации контура бинарного изображения направленными отрезками [22]. Например, разработаны новые правила аппроксимации с заданной точностью контура бинарного изображения векторами, направленные на повышение результативности и быстродействия обнаружения. Приводится краткий обзор основных алгоритмов обнаружения вершины угла на изображении.

Характерный отсчет изображения

В качестве характерной черты изображения выступает его отсчет. Для дефиниции характерного отсчета изображения воспользуемся понятием окрестности.

Пусть функция $f: \mathbf{R}^2 \rightarrow \mathbf{R}$, определенная на множестве $\Omega \subset \mathbf{R}^2$ — это изображение I . Под объектом

(областью) O на изображении I будем понимать область определения: $\Omega \subset \mathbf{R}^2$.

Окрестность отсчета (i, j) изображения I — это множество отсчетов, содержащее отсчет (i, j) и близкие (в каком-либо смысле) к нему отсчеты.

Под характерным отсчетом s изображения I будем понимать отсчет (i, j) с окрестностью $Nbhd(i, j)$, которую можно выделить (каким-либо образом) из остального множества окрестностей $\{Nbhd(n_i)\}$ изображения I .

В литературе известно большое количество алгоритмов обнаружения характерного отсчета изображения [1–22]. Общим недостатком алгоритмов [1–8] является отсутствие геометрического смысла. При этом алгоритмы SIFT [1], PCA-SIFT [2], SURF [3], GLOH [4], DAISY [5] и BRIEF [6] чувствительны к изменению угла, под которым получено изображение. Дескриптора контекст формы [7, 8] инвариантен относительно отображения сдвига, масштаба и поворота, а также устойчив к изменению формы объекта. Дескриптор масштабного пространства кривизны (curvature scale space, МПК) инвариантен к аффинным искажениям и устойчив к помехам [9].

Часто в роли характерной черты изображения выступает вершина угла его объекта [10–22]. Проективное отображение сохраняет вершину угла, кроме случаев превращения угла преобразованием в угол $0, \pi$ и 2π (рад). Однако большинство известных алгоритмов обнаружения вершины угла на изображении [10–21] не обладают инвариантностью относительно проективного отображения.

Из математики известно, что угол может быть образован, например, двумя векторами или дугами, а также кривыми на плоскости.

О п р е д е л е н и е 1. Угол есть фигура (рис. 1), образованная двумя лучами OA и OB (стороны угла), исходящими из одной точки O (вершина угла) [23].

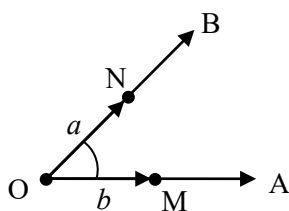


Рис. 1. Угол BOA

Из классической теории экстремалей как решения дифференциальных уравнений известно, что функции $y(x)$ или $y_i(x)$, минимизирующие или максимизирующие определенный интеграл

$$I = \int_{x_0}^{x_F} F[y(x), y'(x), x] dx$$

или

$$I = \int_{x_0}^{x_F} F[y_1(x), y_2(x), \dots, y_n(x); y'_1(x), y'_2(x), \dots, y'_n(x); x] dx,$$

могут иметь угловые точки для значений x , таких что $\partial^2 F / \partial y'^2 = 0$ или матрица $[\partial^2 F / \partial y'_i \partial y'_k]$ — полуопределенная для некоторых y, y' или $y_1, y_2, \dots, y_n; y'_1, y'_2, \dots, y'_n$ или же когда F имеет разрывы [24].

Угловые точки, в частности, могут встречаться [24]:

1) для некоторых x на интервале значений x («свободная» угловая точка $[x_1, y(x_1)]$ или $[x_1; y_1(x_1), y_2(x_1), \dots, y_n(x_1)]$, рис. 2, а);

2) на кривой, поверхности и гиперповерхности

$$S(x, y) = 0 \text{ или } S(x; y_1, y_2, \dots, y_n) = 0,$$

пересекаемых экстремалами («преломление» экстремалей рис. 2, б);

3) на границе некоторой области, из которой экстремали исключены посредством ограничений-неравенств

$$S(x, y) \leq 0 \text{ или } S(x; y_1, y_2, \dots, y_n) \leq 0$$

(рис. 2, в и г).

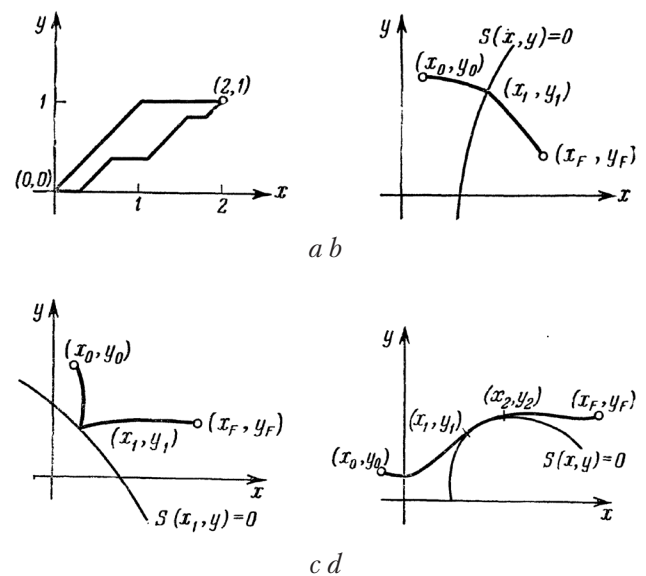


Рис. 2. Экстремали с угловыми точками [24]: а) «свободная» угловая точка; б) преломление экстремали; в) отражение экстремали; г) граничный (односторонний) экстремум

Проблема обнаружения вершины угла на изображении $I(x, y) \in \mathbf{R}$ обусловлена сложностью численного приближения производной функции интенсивности изображения I . Поскольку она известна только в дискретных отсчетах, то невозможно вычислить производную до тех пор, пока не будет определена интенсивность дифференцируемой функции, проходящей через эти отсчеты. При этом приближения ее производных можно определить лишь с ограниченной точностью. На практике используются различные маски для численного приближения производной функции

интенсивности изображения I , в частности, дифференциальный оператор Робертса [25], Превитта [26], Собеля [26] и др.

Пусть «свободная» угловая точка — это вершина угла v . Рассмотрим широко распространенные алгоритмы обнаружения вершины угла на изображении более подробно.

Алгоритмы обнаружения вершины угла Моравека [10] и Харриса [11, 12] инвариантны по отношению к повороту и не чувствительны к изменениям значения интенсивности изображения, но недостаточно точно определяют вершину угла v .

Основным недостатком алгоритмов обнаружения вершины угла на основе дескриптора МПК [14, 15], является чувствительность дескриптора МПК к локальной вариации контура. При этом имеющиеся способы сглаживания не являются достаточными для решения проблемы. Говоря другими словами, Гауссиан с большим значением δ уменьшает шум, но воздействует на локализацию, а Гауссиан с малым значением δ чувствителен к шуму. Наибольшее распространение получили алгоритмы обнаружения вершины угла на основе дескриптора МПК [14, 15].

Известны алгоритмы обнаружения вершины угла на изображении, направленные на решение конкретных задач [20, 21].

Как уже упоминалось выше, почти все известные алгоритмы обнаружения характерного отсчета изображения [1–21] обладают инвариантностью относительно отображений аффинной группы. Аффинное отображение является частным случаем проективного отображения. Поэтому разработка алгоритма обнаружения вершины угла на изображении инвариантного относительно проективного отображения актуальна.

Постановка задачи

Задача 1. Пусть нам дано изображение $I(x, y) \in \mathbb{R}$. Необходимо обнаружить и локализовать вершину угла v ($V\{v\}$) объекта O исходного изображения I .

Анализ поставленной задачи [1–22, 26] показывает, что вершина угла v определяется с помощью некоторого алгоритма обработки исходного изображения, представленного в бинарном виде.

Под произвольным контуром Γ изображения I будем понимать связное множество его отсчетов: $\Gamma = \{n_i\}$, $i = 0, \dots, k - 1$, $k > 0$, $k \in \mathbb{N}$.

Подробно теория контурного анализа рассмотрена в работах [27, 28].

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 . Это связано, в частности, со сложностью анализа как окрестности характерного отсчета $Nbhd(s_i)$ изображения I , так и произвольного контура бинарного изображения. 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 $R \in \mathbb{N}$.

Алгоритм обнаружения вершины угла на изображении на основе аппроксимации контура бинарного изображения

В основе разработанного автором алгоритма обнаружения вершины угла v лежит аппроксимация с заданной точностью контура бинарного изображения направленными отрезками \overline{ON} и \overline{OM} , исходящими из общей точки O (рис. 1). Выделение вершины угла v выполняется за один проход по исходному изображению I , представленному в бинарном виде, в порядке построчной развертки.

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

Шаг 1. Выделить контур как границу объекта O изображения I ;

Шаг 2. Произвести сканирование исходного изображения I представленного в бинарном виде в порядке построчной развертки — слева направо в строке и сверху вниз по строкам;

Шаг 3. Определить претендент r на вершину угла v ;

Шаг 4. Сформировать код контура претендента r на вершину угла v ;

Шаг 5. Идентифицировать претендента r как вершину угла v .

Рассмотрим эти шаги подробнее.

Одним из возможных путей выделения контура на изображении I является применение алгоритма обнаружения края Кэнни [29] к исходному изображению I . Ширина контура равняется одному отсчету изображения. Подробно задача выделения контура как границы объекта изображения рассмотрена в работах [27, 29].

Произвольный отсчет (i, j) бинарного изображения может быть отнесен к одному из двух типов претендентов r на вершину угла v : претендент на вершину угла v первого типа r_1 или претендент на вершину угла v второго типа r_2 — при выполнении нескольких условий (рис. 3). В их основе лежит дифференциальный оператор Робертса [25].

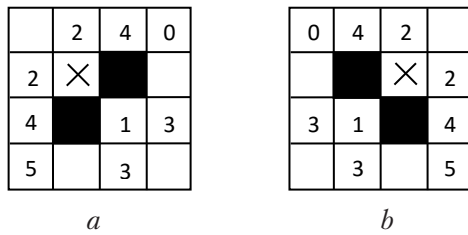


Рис. 3. Схемы определения произвольного отсчета бинарного изображения претендентом r на вершину угла v : а) первый тип r_1 ; б) второй тип r_2

На рис. 3 произвольный отсчет (i, j) бинарного изображения обозначен крестиком. Отсчеты $(i+1, j)$ и $(i, j+1)$ или $(i-1, j)$ и $(i, j+1)$ с образующие вершину угла v (точка O , рис. 1) обозначены черными квадратами. Равенство значений отсчетов c единице является необходимым условием для определения произвольного отсчета (i, j) бинарного изображения претендентом r на вершину угла v . Достаточным условием является выполнение дополнительных требований, в частности, значения отсчетов, обозначенных нулем и пятеркой на рис. 3, не равняются единице одновременно.

Отсчеты c претендента r на вершину угла v могут образовывать произвольный угол, за исключением развернутого и полного углов. Пусть отсчет $c_0 = (i, j+1)$, 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 на вершину угла первого или второго типа (Ag. 2) состоит из следующих шагов:

Шаг 1. 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 $((i+1, j)$ и $(i, j+1))$ претендента r_1 на вершину угла v . Во втором случае, если значение отсчета $(i-1, j)$ равняется единице, то обнаружены отсчеты c $((i-1, j)$ и $(i, j+1))$ претендента r_2 на вершину угла v ;

Шаг 2. Выполняется в случае, если на шаге 1 были обнаружены отсчеты c претендента r_1 . При этом 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, а также отсутствуют подобные отсчеты c , расположенные относительно рассматриваемых отсчетов c $((i+1, j)$ и $(i, j+1))$ 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. Тогда рассматриваемый

отсчет (i, j) является претендентом на вершину угла v первого типа r_1 . Алгоритм завершается;

Шаг 3. Выполняется в случае, если на шаге 1 были обнаружены отсчеты c претендента r_2 . При этом 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, а также отсутствуют подобные отсчеты c , расположенные относительно рассматриваемых отсчетов c $((i-1, j)$ и $(i, j+1))$ 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. Тогда рассматриваемый отсчет (i, j) является претендентом на вершину угла v второго типа r_2 . Алгоритм завершается.

Временная сложность алгоритма Ag. 2 не превышает 19 операций сравнения.

Шаги 2 и 3 алгоритма Ag. 2 обусловлены эффектом ложного контура [27, 30], в результате которого контуры на изображении могут соединяться, образуя сегменты контуров, и эти сегменты иногда соединяются подобно границам. Например, контуры могут образовать ломаную линию, состоящую из чередующихся вертикальных и горизонтальных отрезков, или окрестность в виде матрицы размером $M \times P$ отсчетов, где M и $P \in \mathbb{N}$.

Контур Γ претендента r на вершину угла v подразделяется на две части: контур Γ_a и контур Γ_b .

Контур Γ_a (направленный отрезок \overline{ON} , рис. 1) — это контур Γ начальным отсчетом a_0 которого является отсчет c_1 .

Контур Γ_b (направленный отрезок \overline{OM} , рис. 1) — это контур Γ начальным отсчетом a_0 которого является отсчет c_0 .

Длины контуров Γ_a и Γ_b полагаются равными и не превышают заданное число отсчетов n . Их кодирование осуществляется стандартными ЭВ [27] согласно предложенному способу (рис. 4) на основе 8-связного цепного кода Фримана [27, 30].

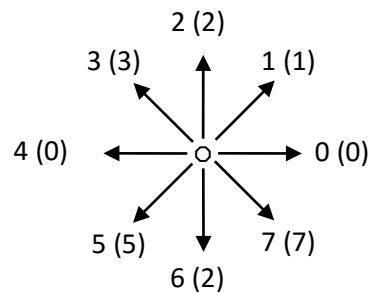


Рис. 4. Нумерация и кодирование направлений 8-связного цепного кода Фримана

На рис. 4 код для каждого из направлений 8-связного цепного кода Фримана указан в скобках.

Идентификация претендента r как вершины угла v выполняется, на пятом шаге работы алгоритма Ag. 1, посредством разработанных правил на основе аппроксимации с заданной точностью кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ направленными отрезками \overline{ON} и \overline{OM} .

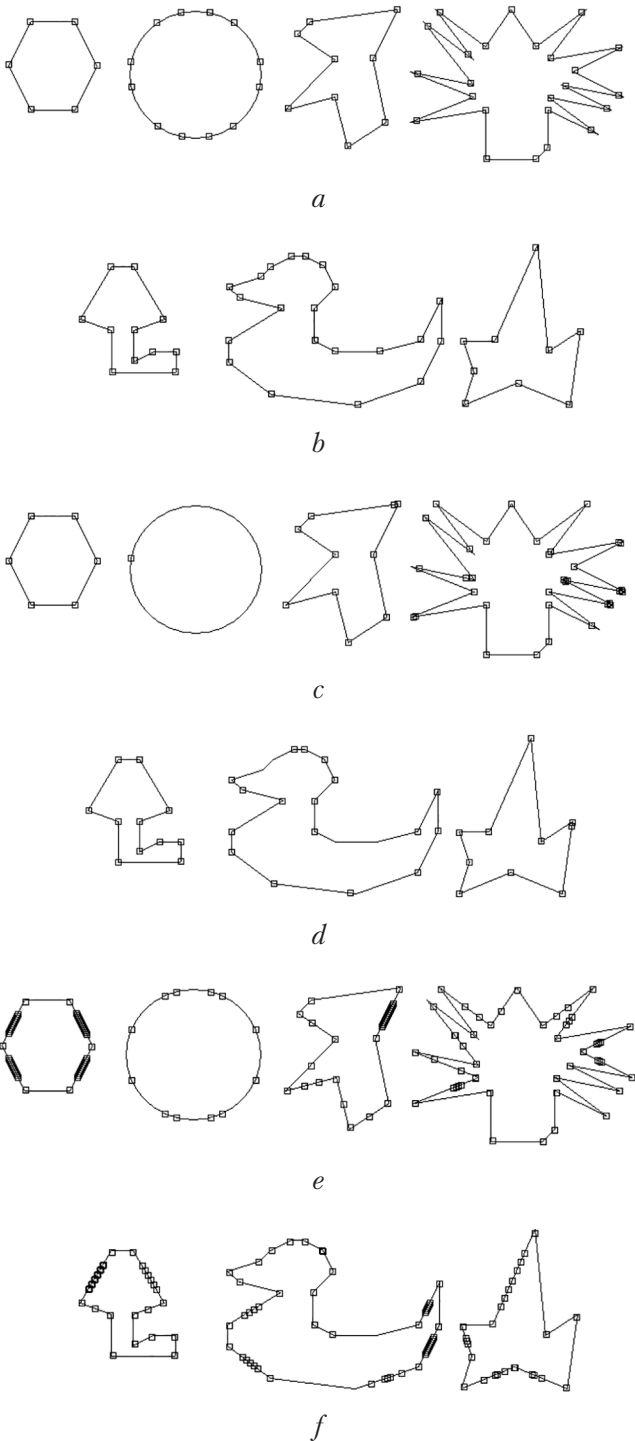


Рис. 5. Примеры работы алгоритмов обнаружения вершины угла на тестовых изображениях I_1 и I_2 , размером 717×209 и 594×224 отсчетов: *a), b)* алгоритм Ag. 1, $n = 8$; *c), d)* алгоритм обнаружения вершины угла на основе дескриптора МПК [14]; *e), f)* алгоритм Харриса [11]

В основе работы алгоритма Ag. 1 лежат следующие параметры:

- n — длина контуров Γ_a и Γ_b ;
- wk — количество допустимых ложных направлений в коде контура Γ_a или Γ_b (шум);
- mx и my — ширина и высота окрестности, в которой может находиться только одна вершина угла. Ее центром является ранее обнаруженная вершина угла.

Пусть размер входа алгоритма Ag. 1 определяет контур длиной m отсчетов, где $m = 2 * n$. Тогда его вычислительная сложность $T_{Ag,1} = \Theta(m)$.

Результат работы алгоритма Ag. 1, алгоритма на основе дескриптора МПК [14] и алгоритма Харриса [11] изображены на рис. 5.

Из рис. 5 видно, что алгоритм Ag. 1 лучше обнаруживает вершину угла v с наименьшим количеством ошибок второго рода.

Алгоритм формирования кода контура претендента на вершину угла

Пусть нам дан претендент r на вершину угла v первого или второго типа. Тогда формирование кода его контура $\Delta\Gamma$ (контур Γ_a и Γ_b , см. рис. 6) выполняется стандартными ЭВ на основе 8-связного цепного кода Фримана.

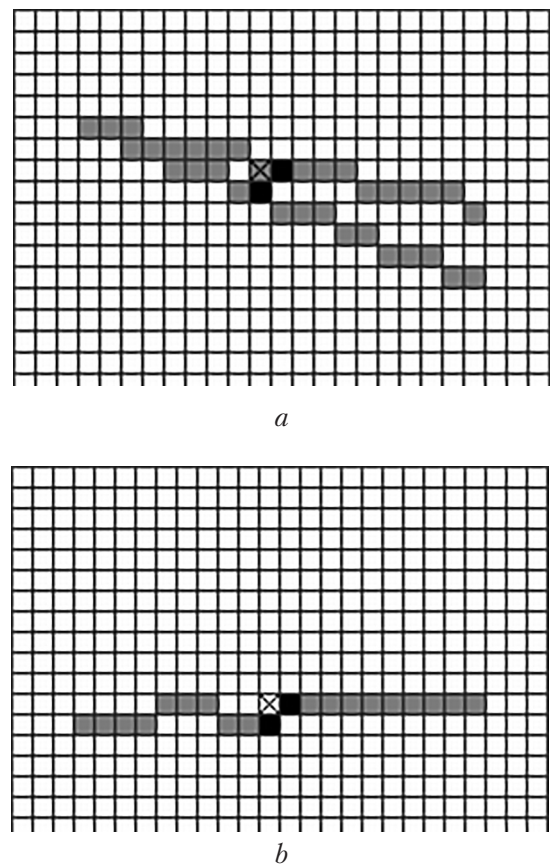


Рис. 6. Угол ВОА на бинарном изображении: *a)* изображение 1; *b)* изображение 2

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

Шаг 1. Задать начальное и конечное направления обнаружения отсчета a_0 контуров Γ_a и Γ_b ;

Шаг 2. Выполнить кодирование контуров Γ_a и Γ_b ;

Шаг 3. Проверить размер кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$. В случае если он $< n$ для любого из них, тогда выполнить алгоритм начиная с шага 1 еще раз и завершить его. Иначе алгоритм завершается.

Рассмотрим эти шаги подробнее.

Прослеживание и кодирование линии контуров Γ_a и Γ_b претендента r на вершину угла v выполняется на основе следующих правил.

Правила начала прослеживания линии контура бинарного изображения:

- претендент r на вершину угла v , а также отсчет, обозначенный на схеме единицей (рис. 3), не должны принимать участие в формировании кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$;
- обнаружение отсчета a_0 контуров Γ_a и Γ_b выполняется согласно одному из двух типов осмотров направлений: осмотр по часовой стрелке {7; 6; 5; 4; 3; 2; 1; 0} и осмотр против часовой стрелки {5; 6; 7; 0; 1; 2; 3; 4};
- для претендента r_1 на вершину угла v обнаружение отсчета a_0 контура Γ_a выполняется на основе осмотра против часовой стрелки, начиная с направления номер 7 {0; 1; 2; 3}, а обнаружение отсчета a_0 контура Γ_b выполняется на основе осмотра по часовой стрелке, начиная с направления номер 7 {6; 5; 4; 3};
- для претендента r_2 на вершину угла v обнаружение отсчета a_0 контура Γ_a выполняется на основе осмотра по часовой стрелке, начиная с направления номер 5 {4; 3; 2; 1}, а обнаружение отсчета a_0 контура Γ_b выполняется на основе осмотра против часовой стрелки, начиная с направления номер 5 {6; 7; 0; 1};
- для претендента r_1 на вершину угла v в случае равенства значений отсчетов изображения $(i+1, j+2)$ и $(i+2, j+1)$ единице (обозначены тройками на рис. 3), а, обнаружение отсчета a_0 контура Γ_a начинается с направления номер 1, а обнаружение отсчета a_0 контура Γ_b с направления номер 5;
- для претендента r_1 на вершину угла v в случае равенства значений отсчетов изображения $(i-1, j)$ и $(i, j-1)$ единице (обозначены двойками на рис. 3), а, обнаружение отсчета a_0 контура Γ_a заканчивается на направлении номер 1, а обнаружение отсчета a_0 контура Γ_b заканчивается на направлении номер 5;
- для претендента r_2 на вершину угла v в случае равенства значений отсчетов изображения $(i-1, j+2)$ и $(i-2, j+1)$ единице (обозначены тройками на рис. 3), b , обнаружение отсчета a_0 контура Γ_a начинается с направления номер 3, а обнаружение отсчета a_0 контура Γ_b начинается с направления номер 7;

- для претендента r_2 на вершину угла v в случае равенства значений отсчетов изображения $(i+1, j)$ и $(i, j-1)$ единице (обозначены двойками на рис. 3), b , обнаружение отсчета a_0 контура Γ_a заканчивается на направлении номер 3, а обнаружение отсчета a_0 контура Γ_b заканчивается на направлении номер 7;
- в случае повторного выполнения шага 1 обнаружение отсчета a_0 контуров Γ_a и Γ_b начинается с направления, номер которого следует за номером первого элемента кода контура $\Delta\Gamma_a$ или $\Delta\Gamma_b$ согласно типу осмотра направлений. При этом номер конечного направления прослеживания контура не может быть превышен, если он был задан на предыдущем шаге работы алгоритма.

Правила прослеживания линии контура бинарного изображения. Отличительной особенностью рассматриваемого алгоритма анализа контура бинарного изображения, является независимость его работы от алгоритма прослеживания линии контура. Отметим, что при прослеживании линии контура также осуществляется подсчет количества того или иного направления в коде соответствующего контура и подсчет количества одинаковых направлений, следующих друг за другом. Нумерация и кодирование направлений 8-связного цепного кода Фримана выполняется согласно схемы, приведенной на рис. 4.

Правила завершения прослеживания линии контура бинарного изображения. Кодирование линии контура прекращается в случае выполнения одного из следующих условий:

- встретился один из отсчетов c ;
- рассмотрены все возможные направления осмотра;
- прослежено n отсчетов контура.

Временная сложность алгоритма Ag. 3 не превышает 4 операций сравнения плюс временная сложность алгоритма прослеживания линии контура не превосходящей $4*n$ отсчетов изображения.

Алгоритм идентификации претендента как вершины угла

Идентификация претендента r как вершины угла v выполняется посредством разработанных автором правил, подразделяющихся на две категории: правила идентификации претендента r как ложной вершины угла и правила идентификации претендента r как вершины угла v . В основе разработанных правил лежит аппроксимация кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ направленными отрезками \overline{ON} и \overline{OM} .

Алгоритм идентификации претендента как вершины угла (Ag. 4) состоит из следующих шагов:

Шаг 1. Проверить размер кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$. В случае если он $< n$ для любого контура завершить алгоритм;

Шаг 2. Выполнить правила идентификации претендента как ложной вершины угла. В случае выполнения любого из правил полагаем, что претендент r не является вершиной угла v и алгоритм завершается;

Шаг 3. Выполнить правила идентификации претендента как вершины угла. В случае выполнения любого из правил полагаем, что претендент r является вершиной угла v . Алгоритм завершается.

Рассмотрим эти шаги подробнее.

Пусть направления 1, 3, 5 и 7 (см. рис. 4) формируют множество $C = \{1, 3, 5, 7\}$.

Правила идентификации претендента как ложной вершины угла:

- претендент r не является вершиной угла v , в случае если значения первых элементов кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ равны 0, а вторые элементы кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом 1–5 или 7–3 соответственно;
- претендент r не является вершиной угла v , в случае если значения первых элементов кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ равны 2, а вторые элементы кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом 1–5 или 3–7 соответственно;
- претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой линией. Коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом: $\{0\} - \{0\}$, $\{2\} - \{2\}$, $\{1\} - \{5\}$, $\{3\} - \{7\}$, $\{5\} - \{1\}$ или $\{7\} - \{3\}$. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{1; 1; 1; 1; 1; 1; 1; 1\}$, а код контура $\Delta\Gamma_b = \{5; 5; 5; 5; 5; 5; 5; 5\}$;
- претендент r не является вершиной угла v , в случае если код контура $\Delta\Gamma_a$ или $\Delta\Gamma_b$ задан направлениями с кодом $\{0, 2\}$ 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 код контура $\Delta\Gamma_a = \{0; 2; 0; 2; 0; 2; 0; 2\}$;
- претендент r не является вершиной угла v , в случае если коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом $\{0, 2\}$. 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. Например, для претендента r_2 код контура $\Delta\Gamma_a = \{0; 0; 0; 0; 2; 0; 3; 0\}$, а код контура $\Delta\Gamma_b = \{0; 0; 0; 0; 7; 0; 0; 0\}$;
- претендент r не является вершиной угла v , в случае если коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направления-

ми с кодом $\{0, 2\}$. 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. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{2; 2; 2; 0; 2; 0; 2; 0\}$, а код контура $\Delta\Gamma_b = \{2; 2; 2; 2; 2; 0; 2; 2\}$;

- претендент r не является вершиной угла v , в случае если код контура $\Delta\Gamma_a$ или $\Delta\Gamma_b$ задан направлениями с кодом $\{0, 2\}$ 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. Например, для претендента r_2 код контура $\Delta\Gamma_a = \{0; 0; 2; 0; 0; 2; 0; 0\}$, а код контура $\Delta\Gamma_b = \{0; 0; 7; 0; 7; 0; 2; 0\}$;
- претендент r не является вершиной угла v , в случае если в коде контура $\Delta\Gamma_a$ или $\Delta\Gamma_b$ значение количества одинаковых направлений, следующих друг за другом с кодом 0 и 2 более единицы или значение количества одинаковых направлений, следующих друг за другом с кодом 0 или 2 более единицы и значение количества кода любого одинакового направления из множества C , следующего друг за другом более нуля. Например, для претендента r_1 код контура $\Delta\Gamma_b = \{2; 5; 5; 2; 2; 2; 2; 2\}$;
- претендент r не является вершиной угла v , в случае если код контура $\Delta\Gamma_a$ или $\Delta\Gamma_b$ заданы направлениями с кодом $\{0, 2\}$ 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. Например, для претендента r_2 код контура $\Delta\Gamma_a = \{0; 3; 0; 3; 0; 0; 3; 0\}$, а код контура $\Delta\Gamma_b = \{0; 2; 0; 0; 2; 0; 0; 2\}$;
- претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой линией. Коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом: $\{0, 1\} - \{0, 5\}$, $\{0, 5\} - \{0, 1\}$, $\{0, 3\} - \{0, 7\}$, $\{0, 7\} - \{0, 3\}$, $\{1, 2\} - \{2, 5\}$, $\{2, 5\} - \{1, 2\}$, $\{2, 3\} - \{2, 7\}$ или $\{2, 7\} - \{2, 3\}$ и наличие кода направления из множества C обязательно. При этом модуль разницы значений количества кода направления из множества C в коде контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ меньше либо равен единице. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{2; 1; 1; 2; 1; 1; 2; 1\}$, а код контура $\Delta\Gamma_b = \{5; 2; 5; 5; 2; 5; 5; 2\}$;
- претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой

- линией. Коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом: $\{0, 1, 2\} - \{0, 2, 5\}$, $\{0, 2, 5\} - \{0, 1, 2\}$, $\{0, 2, 3\} - \{0, 2, 7\}$ или $\{0, 2, 7\} - \{0, 2, 3\}$ и наличие 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. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{0; 0; 0; 2; 0; 0; 1; 0\}$, а код контура $\Delta\Gamma_b = \{0; 0; 2; 0; 0; 5; 0; 2\}$;
- претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой линией. Коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом: $\{0, 1, 2\} - \{0, 2, 5\}$, $\{0, 2, 5\} - \{0, 1, 2\}$, $\{0, 2, 3\} - \{0, 2, 7\}$ или $\{0, 2, 7\} - \{0, 2, 3\}$ и 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. Например, для претендента r_2 код контура $\Delta\Gamma_a = \{3; 3; 2; 3; 3; 3; 2; 0\}$, а код контура $\Delta\Gamma_b = \{2; 7; 7; 7; 2; 7; 7; 7\}$;
 - претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой линией. Коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом: $\{0, 1, 2\} - \{0, 2, 3\}$, $\{0, 2, 3\} - \{0, 1, 2\}$, $\{0, 2, 3\} - \{0, 2, 5\}$, $\{0, 2, 5\} - \{0, 2, 3\}$, $\{0, 2, 7\} - \{0, 2, 5\}$, $\{0, 2, 5\} - \{0, 2, 7\}$, $\{0, 2, 7\} - \{0, 1, 2\}$ или $\{0, 1, 2\} - \{0, 2, 7\}$ и 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 кода направления из множества C в каждом из контуров меньше либо равен единице и в одном из контуров значение количества кода направления из множества C равно единице. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{0; 0; 1; 2; 0; 2; 0; 2\}$, а код контура $\Delta\Gamma_b = \{2; 0; 2; 7; 0; 2; 0; 0\}$;
 - 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. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{0; 7; 0; 7; 0; 2; 2; 2\}$, а код контура $\Delta\Gamma_b = \{2; 2; 7; 0; 2; 7; 0; 0\}$;
 - претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой линией. Коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом: $\{0, 7\} - \{0, 2, 3\}$, $\{2, 7\} - \{0, 2, 3\}$, $\{0, 2, 3\} - \{0, 7\}$, $\{0, 2, 3\} - \{2, 7\}$, $\{0, 3\} - \{0, 2, 7\}$, $\{2, 3\} - \{0, 2, 7\}$, $\{0, 2, 7\} - \{0, 3\}$, $\{0, 2, 7\} - \{2, 3\}$, $\{0, 5\} - \{0, 1, 2\}$, $\{2, 5\} - \{0, 1, 2\}$, $\{0, 1, 2\} - \{0, 5\}$, $\{0, 1, 2\} - \{0, 1\}$, $\{0, 1\} - \{0, 2, 5\}$, $\{1, 2\} - \{0, 2, 5\}$, $\{0, 1\} - \{0, 2, 5\}$ или $\{1, 2\} - \{0, 2, 5\}$. При этом для контура код которого содержит коды наименьшего числа направлений (мощность его множества не превышает 2), соответствующие направления встречаются более одного раза. Для оставшегося контура код направления из множества C 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 единице и модуль разницы количества значения направления с одинаковым кодом (0 или 2), следующих друг за другом в коде каждого из контуров меньше либо равно единице. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. Например, для претендента r_2 код контура $\Delta\Gamma_a = \{0; 3; 0; 0; 2; 0; 0; 2\}$, а код контура $\Delta\Gamma_b = \{0; 0; 7; 0; 7; 0; 0; 7\}$;
 - претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой линией. Код контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ задан направлениями с кодом 0 или 2 и одного из направлений из множества C (например, $\{0, 3\}$ или $\{2, 3\}$) и каждое из направлений присутствует в коде более одного раза. При этом код оставшегося контура ($\Delta\Gamma_a$ или $\Delta\Gamma_b$) задан направлениями с кодом $\{0, 2\}$ и его код не состоит лишь из направлений с кодом 0 или 2, а также модуль разницы количества значения направления с одинаковым кодом (0 или 2) в коде каждого из контуров меньше либо равно единице и модуль разницы количества значения направления с одинаковым кодом (0 или 2), следующих друг за другом в коде каждого из контуров меньше либо равно единице. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. Например, для претендента r_1 код кон-

- тура $\Delta\Gamma_a = \{0; 0; 0; 2; 0; 0; 7\}$, а код контура $\Delta\Gamma_b = \{0; 0; 5; 0; 0; 5; 0; 0\}$;
- претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой линией. Код контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ задан направлениями с кодом 0 или 2 и одного из направления из множества C (например, $\{0, 5\}$ или $\{2, 5\}$) и каждое из направлений присутствует в коде более одного раза. При этом код оставшегося контура ($\Delta\Gamma_a$ или $\Delta\Gamma_b$) задан направлениями с кодом $\{0, 2\}$ и его код не состоит лишь из направлений с кодом 0 или 2, а также модуль разницы количества значения направления с одинаковым кодом (0 или 2) в коде каждого из контуров меньше либо равно единицы и значение количества значения направления с одинаковым кодом (0 или 2), следующих друг за другом в коде каждого из контуров больше нуля. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{0; 2; 0; 2; 0; 0; 0; 2\}$, а код контура $\Delta\Gamma_b = \{0; 0; 3; 0; 0; 0; 0; 3\}$;
 - претендент r не является вершиной угла v , в случае если контуры Γ_a и Γ_b аппроксимируются прямой линией. Коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом: $\{0, 1, 2\} - \{0, 2, 5\}$, $\{0, 2, 5\} - \{0, 1, 2\}$, $\{0, 2, 3\} - \{0, 2, 7\}$ или $\{0, 2, 7\} - \{0, 2, 3\}$ и значение количества кода каждого из направлений в описании контура меньше n , а также модуль разницы значений количества кода направления 0 или 2 в каждом из контуров меньше либо равен единице. При этом коды четных, нечетных или всех элементов кода контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ равны: 0, 2, 1–5, 3–7, 5–1 или 7–3. Например, для претендента r_2 код контура $\Delta\Gamma_a = \{7; 7; 0; 7; 7; 7; 0; 7\}$, а код контура $\Delta\Gamma_b = \{0; 3; 3; 3; 0; 3; 3; 3\}$.

Временная сложность правил идентификации претендента как ложной вершины угла не превзойдет $1004 + 32 * n$ операций сравнения.

Правила идентификации претендента как вершины угла:

- претендент r является вершиной угла v , в случае если код контура $\Delta\Gamma_a$ задан направлением с кодом 0, а код контура $\Delta\Gamma_b$ задан направлением с кодом 2 или наоборот. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{0; 0; 0; 0; 0; 0; 0; 0\}$, а код контура $\Delta\Gamma_b = \{2; 2; 2; 5; 2; 2; 2; 2\}$;
- претендент r является вершиной угла v , в случае если коды контуров $\Delta\Gamma_a$ и $\Delta\Gamma_b$ заданы направлениями с кодом $\{0, 2\}$ и одного из направлений: 1, 3, 5, 7. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. На-

- пример, для претендента r_1 код контура $\Delta\Gamma_a = \{0; 0; 2; 1; 0; 1; 0; 0\}$, а код контура $\Delta\Gamma_b = \{0; 3; 2; 3; 2; 3; 2; 2\}$;
- претендент r является вершиной угла v , в случае если код контура $\Delta\Gamma_a$ или $\Delta\Gamma_b$ задан направлением с кодом 0 или 2. При этом код оставшегося контура ($\Delta\Gamma_a$ или $\Delta\Gamma_b$) задан направлениями с кодом $\{0, 2\}$ и каждое направление встречается в коде контура более одного раза. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{0; 2; 0; 0; 2; 0; 2; 0\}$, а код контура $\Delta\Gamma_b = \{2; 2; 0; 2; 2; 2; 2; 2\}$;
- претендент r является вершиной угла v , в случае если код контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ задан направлениями с кодом $\{0, 2\}$ и каждое направление встречается более одного раза. При этом модуль разницы значений количества одинаковых направлений больше единицы. Допускается присутствие любого другого направления в коде контура $\Delta\Gamma_a$ и $\Delta\Gamma_b$ не более wk раз. Например, для претендента r_1 код контура $\Delta\Gamma_a = \{0; 0; 0; 2; 0; 0; 2; 0\}$, а код контура $\Delta\Gamma_b = \{0; 2; 0; 2; 0; 2; 0; 2\}$.

Временная сложность правил идентификации претендента как вершины угла не превзойдет 78 операций сравнения.

Временная сложность алгоритма Ag. 4 не превзойдет $1082 + 32 * n$ операций сравнения.

Результаты тестирования

Алгоритм Харриса [11] и алгоритм на основе дескриптора МПК [14] **используются для** сравнительного анализа точности обнаружения вершины угла v алгоритма Ag. 1.

Тестирование алгоритма Ag. 1 осуществлялось со следующими значениями параметров: $n = 8$, $wk = 1$ и $mx = my = 3$. При этом использовался алгоритм прослеживания линии контура бинарного изображения Я.А. Фурмана [27]. Выделение контура на исходном изображении I выполняется на основе алгоритма обнаружения края Кэнни [27], применяющегося со значениями порогов: $T_1 = 0$ и $T_2 = 0,35$. Ширина контура равняется одному отсчету изображения.

В таблице 1 приведены результаты оценки работы рассматриваемых алгоритмов обнаружения вершины угла на исходных изображениях. Тестирование выполнялось на базе ЭВМ IBM PC Pentium IV 3.2 ГГц с 4096 Мб ОЗУ. Время работы алгоритма на каждом из тестов считалось как среднее из 100 запусков на одном и том же наборе данных. Использовалась база реальных аэрокосмических изображений, включающая 140 изображений. Тестирование выполнялось в среде Matlab.

Таблиця 1

Оценки результата обнаружения вершины угла

Алгоритм обнаружения вершины угла	Среднее время обнаружения, с	Количество вершин углов	Ошибка первого рода, k_1	Ошибка второго рода, k_2
1	2	3	4	5
Харриса [11]	57,79	192052	0,29	0,65
МПК [14]	425,92	192052	0,15	0,21
Алгоритм Ag. 1	198,08	192052	0,07	0,09

Алгоритмы обнаружения вершины угла Харриса [11] и на основе дескриптора МПК [14] тестировались с экспериментально подобранными значениями параметра δ , при котором алгоритмы показали наилучший результат в обнаружении вершины угла $\delta = 2$ и $\delta = 3$. Проведенное тестирование показало, что с уменьшением значения δ алгоритм обнаружения на основе дескриптора МПК [14] лучше определяет тупые углы. Однако при этом значение количества найденных ложных вершин углов на острых углах возрастает. С увеличением значения δ уменьшается значение количества определяемых вершин углов.

Как уже упоминалось выше проективное отображение сохраняет вершину угла. Тестирование показало, что алгоритма Ag. 1 выделяет вершину угла инвариантно относительно проективного отображения. Это достигается за счет выполняемой аппроксимации линии контура Г.

Алгоритм Ag. 1 находит свое практическое применение при решении задачи обнаружения и распознавания простого объекта на изображениях [28, 31].

Заключение

Предложен улучшенный алгоритм обнаружения вершины угла на изображении. Его основу составляет алгоритм определения отсчета бинарного изображения претендентом на вершину произвольного угла, за исключением развернутого и полного углов. Идентификация претендента как вершины угла выполняется на основе аппроксимации контура бинарного изображения векторами. Это позволяет анализировать произвольный контур и выполнять точное обнаружение вершины угла без существенного возрастания времени обработки. Результаты экспериментов показали, что разработанный автором алгоритм лучше обнаруживает вершину угла с наименьшим количеством определяемых ложных вершин углов по сравнению с традиционными алгоритмами. При этом он инвариантен относительно проективного отображения.

Литература

1. Burger W., Burge M.J. Principles of digital image processing. London: Springer, 2013. P. 229–296.
2. Li K., [et al.]. Computer Engineering and Networking. Cham: Springer International Publishing, 2014. P. 679–686.
3. Abeles P. Speeding Up SURF. Advances in Visual Computing. Berlin: Springer, 2013. P. 454–464.
4. Mikolajczyk K., Schmid C. A performance evaluation of local descriptors // In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. Madison (WI), USA, 2003. P. 257–264.
5. Tola E., Lepetit V., Fua P. DAISY: an efficient dense descriptor applied to wide baseline stereo // IEEE Trans. on Pattern Analysis and Machine Intelligence. 2010. V. 32. № 5. P. 815–830.
6. Calonder M., Lepetit V., Strecha C., Fua P. BRIEF: Binary Robust Independent Elementary Features // In: Proc. European Conference on Computer Vision. Crete, Greece, 2010. P. 778–792.
7. Mori G., Belongie S., Malik J. Efficient Shape Matching Using Shape Contexts // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005. Vol. 27, № 11. P. 1832–1837.
8. Wu H., [et al.]. Foundations of intelligent systems. Berlin: Springer, 2014. P. 889–900.
9. Кобзарь Г. А. Модель межмасштабного пространства кривизны для представления формы геометрических объектов // Искусственный интеллект. 2008. № 1. С. 153–165.
10. Moravec H.P. Towards automatic visual obstacle avoidance // In Proc. of the 5th Intern. Joint Conf. of Artificial Intelligence. Cambridge, 1977. P. 587–598.

11. Harris C., Stephens M. A combined corner and edge detector // In Proceedings of the Fourth Alvey Vision Conference. 1988. P. 147–152.
12. Shi F., Huang X., Duan Y. Robust Harris-Laplace Detector by Scale Multiplication. *Advances in Visual Computing*. Berlin: Springer, 2009. P. 265–274.
13. Trajkovic M., Hedley M. Fast corner detection. *Image and vision computing* 1998. Vol. 16. P. 75–87.
14. 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.
15. 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.
16. 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.
17. Quinlan J. R. Induction of decision trees // *Machine Learning*. 1986. Vol. 1. P. 81–106.
18. 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.
19. 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.
20. 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.
21. Feltes M., [et al.]. Improved Contour-Based Corner Detection for Architectural Floor Plans // *Graphics Recognition. Current Trends and Challenges*. 2014. P. 191–203.
22. Козловский А. Н. Детектор угловых точек на основе аппроксимации контуров объектов изображения // *Информатика*. 2010. № 4. С. 36–47.
23. Выгодский М. Я. Справочник по математике. — М.: Астрель, 2010. — 1055 с.
24. Корн Г., Корн Т. Справочник по математике для научных сотрудников и инженеров. — М.: Наука, 1978. — 832 с.
25. Робертс Л. Автоматическое восприятие трехмерных объектов. Интегральные роботы. — М.: Мир, 1973. — 162–208 с.
26. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. — М.: Техносфера, 2006. — 616 с.
27. Фурман Я. А. Введение в контурный анализ: приложения к обработке изображений и сигналов. — М.: Физматлит, 2003. — 588 с.
28. Козловский А. Н. Математические модели элементарного и простого объекта изображения на основе контурного анализа. // *Международный научный журнал*. 2016. № 3. С. 75–77. DOI:10.21267/IN.2016.3.1107.
29. Canny J. Computational Approach to Edge Detection // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986. Vol. 8, № 6. P. 679–698.
30. Гонсалес Р., Вудс Р. Цифровая обработка изображений. — М.: Техносфера, 2006. — 1072 с.
31. 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.