

АЛГОРИТМ РАСПОЗНАВАНИЯ ОБЪЕКТОВ НА ПЛОСКИХ ИЗОБРАЖЕНИЯХ В СИСТЕМАХ ТЕХНИЧЕСКОГО ЗРЕНИЯ

М. А. Пантюхин*, Е. А. Самойлин*, А. Ю. Дроздов**

*ВУНЦ ВВС «Военно-воздушная академия им. проф. Н. Е. Жуковского и Ю. А. Гагарина» (г. Воронеж)

**ООО «Геоиндустрия»

Поступила в редакцию 01.03.2016 г.

Аннотация. В статье рассмотрен алгоритм распознавания плоских изображений, основанный на контурном анализе формы наблюдаемых и эталонных объектов. Как показали результаты численных исследований и примеры распознавания перекрывающихся объектов, предложенный алгоритм позволяет решать задачу распознавания в системах технического зрения.

Ключевые слова: распознавание образов, эталоны, контурный анализ, адаптивные маски, аффинные преобразования, кластеризация.

Annotation. The flat image recognition base on the contour shape analysis of the observed and prototype objects is reviewed. The numerical estimations and the examples of the recognition of the overlapping objects show that proposed algorithm allows solving the recognition tasks in the computer vision systems.

Keywords: pattern recognition, prototypes, contour analysis, adaptive masks, affine transformations, clustering.

ВВЕДЕНИЕ

Как известно, в настоящее время одной из проблем распознавания объектов на плоских изображениях, получаемых различными системами технического зрения, является формирование признаков описаний объектов, инвариантных к основным группам преобразований [1]. Одним из интересных подходов к решению указанной проблемы представляется направление, основанное на контурном анализе изображений [2]. Алгоритм, описанный в работе [2], представляет практический интерес для реализации предложенных инвариантных процедур в автономных системах технического зрения, в частности мобильных роботах [3] и беспилотной авиации [4]. Недостатком указанного алгоритма является недостаточная его устойчивость к аффинным преобразованиям объектов на плоском цифровом изображении, а также к потере части контура распознаваемого объекта.

Целью работы является повышение устойчивости процедуры распознавания к аффинным преобразованиям объектов на плоских цифровых изображениях.

СУЩНОСТЬ АЛГОРИТМА РАСПОЗНАВАНИЯ

Изображение-сцена рассматривается в виде прямоугольной матрицы пикселей, полученной многоэлементным приемником оптического излучения системы технического зрения (СТЗ). При этом значением элемента матрицы является квантованный уровень амплитуды яркости в соответствующей точке матрицы [5, 6]. С целью упрощения изложения алгоритма, далее рассматриваются полутоновые изображения.

Объекты на изображении-сцене рассматриваются в контурном (оконтуренном) виде. Для получения контурного препарата (КП) могут быть использованы самые разнообразные методы и алгоритмы, в частности, градиентные (на основе первой производной) [6], лапласиан гауссиана (на основе

© Пантюхин М. А., Самойлин Е. А., Дроздов А. Ю., 2016

второй производной) [6] и т.д. Может также быть использован алгоритм адаптивного градиентного выделения контуров объектов на цифровых изображениях [5], показывающий свою устойчивость в условиях наличия на изображениях импульсных помех (ИП) [6].

Этап 1. Полигональная аппроксимация контурного препарата

На первом этапе алгоритма осуществляется построение кусочно-линейных сегментов КП. Проведенные численные исследования показали, что классические алгоритмы [7] полигональной аппроксимации, заключающиеся в упрощении кривых, малопригодны для решения рассматриваемой задачи. Поэтому предложена процедура, заключающаяся в разбиении контура на сегменты равной длины. На рис. 1 представлен фрагмент контура и результат применения к нему процедуры разбиения на сегменты равной длины. Предлагаемая процедура не является упрощением кривой на основе известного радиального [7] алгоритма. Отличие состоит в том, что в подалгоритме осуществляется вычисление координат новой точки, не обязательно лежащей в пределах КП объекта.

Пусть D – исходное упорядоченное подмножество точек КП:

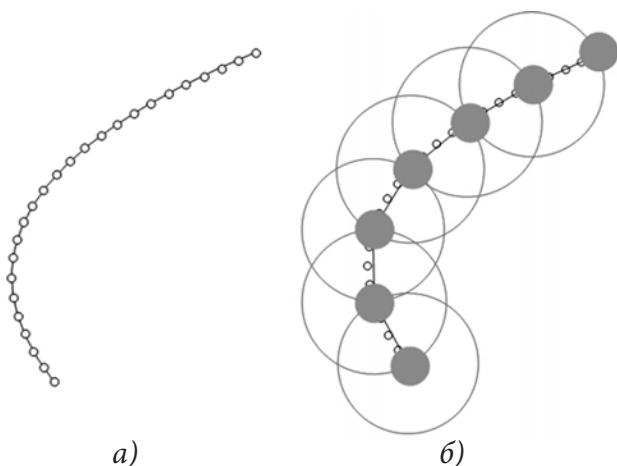


Рис. 1. Построение сегментов равной длины:
а) исходная последовательность точек;
б) принцип построения сегментов

$$D = \{d_1(x_1, y_1), \dots, d_K(x_K, y_K)\}, \quad (1)$$

где (x_i, y_i) – декартовы координаты точки d_i , $i = 1, K$, K – мощность подмножества.

Предлагаемый подалгоритм полигональной аппроксимации состоит из следующих шагов.

Шаг 0. Задается расстояние R , определяющее длину сегмента.

Шаг 1. Фиксируется первая точка исходного набора $d^*(x^*, y^*) = d_1(x_1, y_1)$, которая затем включается в выходной набор \hat{D} : $\hat{D} = \{d^*(x^*, y^*)\}$.

Шаг 2. Последовательно рассматриваются точки, следующие за точкой $d^*(x^*, y^*)$, при этом рассчитывается евклидово расстояние ρ от этих точек до нее и определяется пара соседних точек $d_{R_{\max}}(x_{R_{\max}}, y_{R_{\max}})$, $d_{R_{\min}}(x_{R_{\min}}, y_{R_{\min}})$ таких, что выполняется следующая система неравенств:

$$\begin{cases} \rho(d^*, d_{R_{\min}}) \leq R, \\ \rho(d^*, d_{R_{\max}}) \geq R. \end{cases} \quad (2)$$

Шаг 3. Определяется очередная точка последовательности \hat{D} как точка пересечения окружности с центром в $d^*(x^*, y^*)$ радиуса R и линии, соединяющей $d_{R_{\max}}(x_{R_{\max}}, y_{R_{\max}})$ и $d_{R_{\min}}(x_{R_{\min}}, y_{R_{\min}})$, то есть решается система уравнений:

$$\begin{cases} (\hat{x} - x^*)^2 + (\hat{y} - y^*)^2 = R^2, \\ (\hat{x} - x_{R_{\min}})/(x_{R_{\max}} - x_{R_{\min}}) = \\ = (\hat{y} - y_{R_{\min}})/(y_{R_{\max}} - y_{R_{\min}}). \end{cases} \quad (3)$$

Полученная точка $\hat{d}(\hat{x}, \hat{y})$ добавляется в выходной набор \hat{D} .

Пары точек $d_{R_{\max}}(x_{R_{\max}}, y_{R_{\max}})$ и $d_{R_{\min}}(x_{R_{\min}}, y_{R_{\min}})$, удовлетворяющих системе (3), может не существовать, что возможно в двух случаях.

Случай 1. Последовательно расположенные точки, следующие за $d^*(x^*, y^*)$, удалены от нее на расстояние, превышающее R . При этом рекурсивно вычисляются по формуле (3) координаты точки $\hat{d}(\hat{x}, \hat{y})$, полагая $d_{R_{\max}}(x_{R_{\max}}, y_{R_{\max}}) = d^*(x^*, y^*)$.

Случай 2. Исходная последовательность точек исчерпана, и ее последняя точка расположена от $d^*(x^*, y^*)$ на расстоянии меньшем R , тогда крайняя точка добавляется в \hat{D} автоматически.

Шаг 4. Переопределяется точка $d^*(x^*, y^*)$: $d^*(x^*, y^*) = d(\hat{x}, \hat{y})$, затем осуществляется переход к **Шагу 2**.

Сравнение предлагаемого подалгоритма проведем с наиболее известным алгоритмом *Douglas-Peucker* [7]. Пусть для алгоритма *Douglas-Peucker* задан допуск упрощения ε_{DP} . Учтем этот параметр следующим образом. При добавлении очередной точки в выходную последовательность дополнительно будем вычислять расстояния от исключаемых на данном шаге точек до прямой, соединяющей добавляемую точку и предыдущую, уже добавленную. Если среди исключаемых точек есть хотя бы одна такая, что расстояние от нее до формируемого линейного сегмента превышает ε_{DP} , то алгоритм инициализируется новым значением R , уменьшенным вдвое, и выполняется сначала. Таким образом, оба алгоритма будут иметь одинаковую точность аппроксимации, при этом целесообразно их сравнивать по времени выполнения и результирующему количеству полученных сегментов.

На рис. 2 представлен результат работы двух алгоритмов по монотонной кривой и по «сложной» (не монотонной) фигуре; приняты следующие обозначения: исходные кривая и фигура обозначены штрихпунктирной линией; результат упрощения алгоритмом *Douglas-Peucker* – пунктирной линией; результат аппроксимации предлагаемым подалгоритмом (1)–(3) – непрерывной линией.

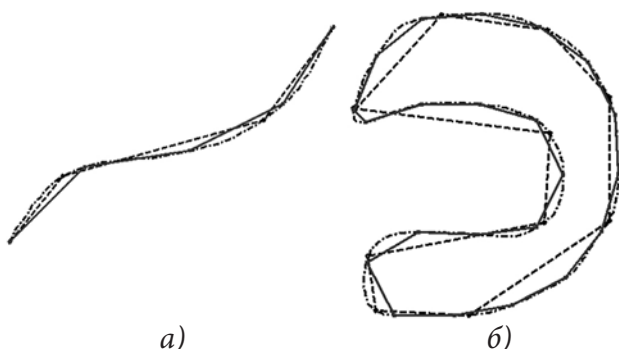


Рис. 2. Полигональная аппроксимация предлагаемым и известным алгоритмами: а) монотонная кривая; б) «сложная» фигура

На рис. 2а аппроксимирована монотонная кривая с числом точек, равным 147. Для алгоритма *Douglas-Peucker* потребовалось 0,04 с,

при этом выходная последовательность составила 4 точки. Для подалгоритма (1)–(3) понадобилось 0,01 с при 5 точках в последовательности. На рис. 2б аппроксимирована фигура, состоящая из 505 точек. Для алгоритма *Douglas-Peucker* потребовалось 0,2 с и 11 точек в последовательности, для подалгоритма (1)–(3) – 0,07 с при 22 точках. Расчет временных затрат производился на платформе Intel(R) Celeron(R) CPU 1007U @ 1.50Ghz, 4.00Gb.

Этап 2. Определение параметров сложных сегментов и формирование идентификационных ключей

В алгоритме [2] осуществляется разбиение контура на последовательности линейных участков (сложных сегментов), и затем в качестве признака распознавания выбирается набор значений углов между соседними сегментами. Пример такого процесса показан на рис. 3.

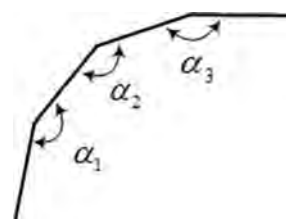


Рис. 3. Описание контура последовательностью углов

Указанный признак распознавания устойчив к преобразованию равномерного масштабирования, вращения и смещения сложных сегментов на изображении, однако на практике, в условиях таких трансформаций, сличение с эталоном растровых изображений представляет нетривиальную задачу. Поэтому предлагаемая модификация алгоритма [2] состоит в формировании не единственного набора, а множества перекрывающихся сложных сегментов.

Пусть в результате **этапа 2** имеется последовательность точек полигональной аппроксимации $T = \{(x_1, y_1), \dots, (x_P, y_P)\}$, где P – общее число точек и $\rho(d_i(x_i, y_i), d_j(x_j, y_j)) = R, \forall i, j: j = i \pm 1$.

Шаг 0. Задается число точек S ($S \geq 3$), составляющих сложный сегмент, при этом

количество линейных сегментов в его составе будет равно $S - 1$, а определяемое ими количество углов будет равно $S - 2$.

Шаг 1. Формируется последовательность сложных сегментов таких, что крайняя точка предыдущего будет являться начальной точкой следующего:

$$U_0 = \left\{ \{d_1, d_2, \dots, d_s\}, \dots, \{d_{i,s}, d_{i,s+1}, \dots, d_{(i+1),s-(i-1)}\}, \dots \right\}.$$

Шаг 2. Осуществляется циклический сдвиг влево начальной точки первого сложного сегмента, в результате которого начальной точкой становится его вторая точка, а крайней точкой станет первая точка следующего сложного сегмента. Последний сложный сегмент не рассматривается, так как для него нельзя заменить крайнюю точку. Такой циклический сдвиг возможно осуществить не более чем S раз, получив при этом последовательности U_1, \dots, U_s перекрывающихся сложных сегментов.

Дополнительно **Этап 2** и **циклический сдвиг** повторяются для величины расстояния R , переопределяемого значениями $0.25 \cdot R, 0.5 \cdot R, 2 \cdot R, 4 \cdot R$. Назовем такую процедуру процедурой *многомасштабности*, в результате которой не происходит изменения физического размера изображения, но меняется точность полигональной аппроксимации.

Получив конечный набор сложных сегментов, осуществляется расчет значений определяющих их углов. Так, угол α_{abc} , образованный тремя точками a, b, c , рассчитывается с помощью образования промежуточных векторов в соответствии с выражениями:

$$\vec{ab} = \{b_x - a_x, b_y - a_y\}, \quad (4)$$

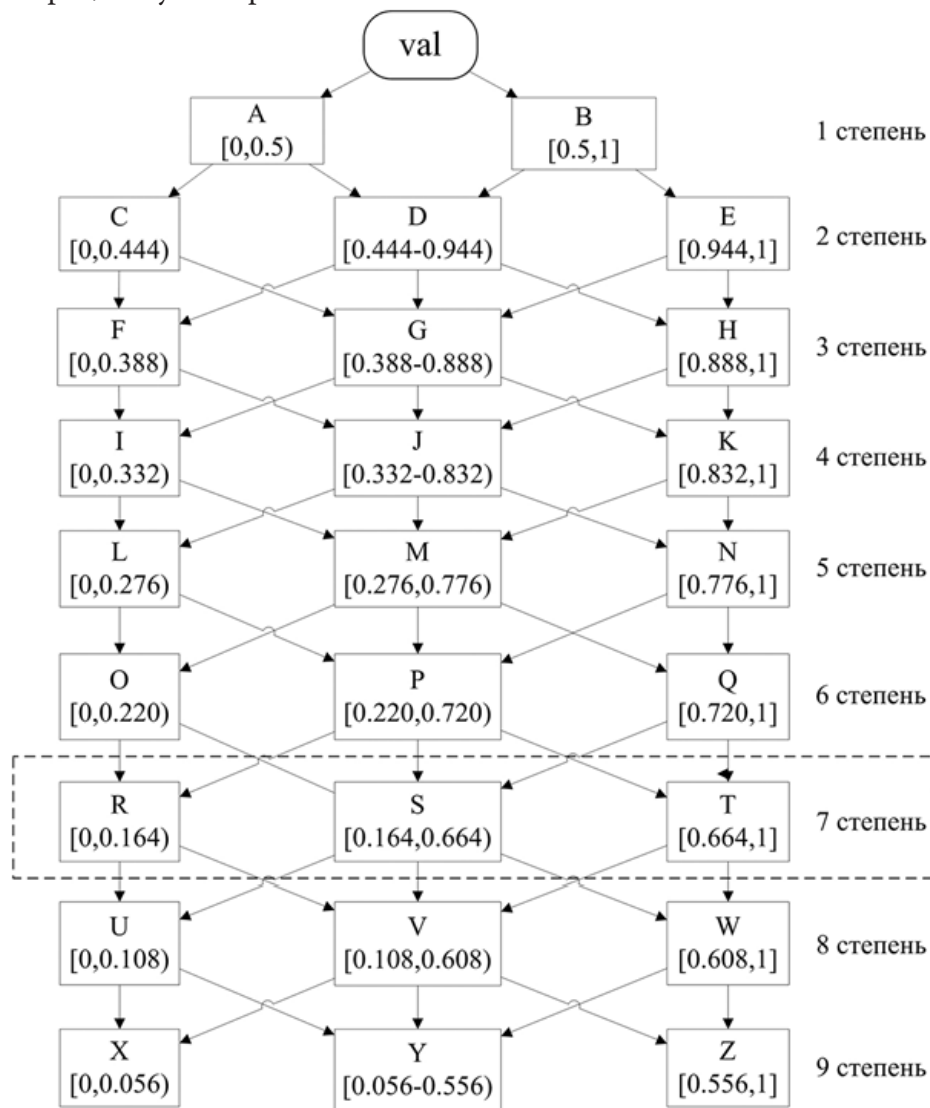


Рис. 4. Процедура квантования вещественного значения «val»

$$\vec{cb} = \{b_x - c_x, b_y - c_y\}, \quad (5)$$

$$\alpha_{abc} = \arctg\left(\frac{ab_x \cdot cb_y - ab_y \cdot cb_x}{ab_x \cdot cb_x + ab_y \cdot cb_y}\right), \quad (6)$$

где (a_x, a_y) , (b_x, b_y) , (c_x, c_y) – координаты точек a , b , c соответственно.

Далее осуществляется квантование значений углов для дальнейшего поиска и сравнения сложных сегментов.

Процедура квантования предполагает использование оператора отображения набора нормированных значений углов в последовательности символов латинского алфавита $\{A, B, \dots, Z\}$:

$$\Psi: U \rightarrow \Lambda, \quad (7)$$

где Ψ – оператор отображения, U – набор углов, Λ – литеральные последовательности, составленные из символов латинского алфавита.

Оператор Ψ осуществляет отображение в соответствии с рис. 4.

На рис. 4 значение степени квантования определяет мощность алфавита символов, то есть первая степень задается алфавитом $\{A, B\}$, вторая – $\{A, B, C, D, E\}$, третья – $\{A, B, C, D, E, F, G, H\}$ и т. д. Таким образом, чем больше значение степени, тем точнее квантуется значение угла. Квантование предполагает, что близкие значения углов будут незначительно отличаться в смысле расстояния Хэмминга [8]. Далее значения последовательных углов, образующих сложные сегменты, записываются в одну литеральную строку, образуя тем самым *идентификационный ключ* сложного сегмента, по которому осуществляется их сравнение. Так, если сложный сегмент образован пятью точками, то есть состоит из четырех сегментов и следовательно, определяется тремя углами, значения которых, например, равны 110° , 120° и 80° , то литеральная последовательность для степени квантования «3» будет «BDGBDGADG». Результаты сравнения сложных сегментов по такому *идентификационному ключу* представляют собой первоначальные гипотезы о соответствии сложных сегментов эталонов и сцены.

Таким образом, результатом данного этапа будет таблица соответствий сложных сегментов моделей сложным сегментам сцены на основе *идентификационного ключа*.

Этап 3. Кластеризация гипотез соответствия в пространстве коэффициентов аффинных преобразований

Устойчивость выбранного пространства признаков в виде последовательности углов, определяющих сложные сегменты, дополнительно «усиленная» процедурой *циклического сдвига* начального сегмента и процедурой *многомасштабности* на этапе 2, обеспечивает к преобразованиям изображения «сдвиг», «поворот», «масштабирование». Эти преобразования относятся к классу аффинных [9], которые в двумерном пространстве удобно рассматривать как трансформацию точки или элементарного вектора относительно начала декартовой системы координат. Так, рассматривая некоторую точку A с координатами (x_A, y_A) , переходят к вектору $A\vec{O}$, где $O(0,0)$ – начало координат. Рассмотрим основные аффинные преобразования.

Аффинное преобразование «переноса» точки $A(x_A, y_A)$ в точку $B(x_B, y_B)$ можно записать матричным выражением

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta_{x_{BA}} \\ 0 & 1 & \Delta_{y_{BA}} \end{bmatrix} \times [x_A \quad y_A \quad 1]^T, \quad (8)$$

где $\Delta_{x_{BA}}$, $\Delta_{y_{BA}}$ – приращения по осям при переносе точки, $[\circ]^T$ – операция транспонирования вектора.

Преобразование «вращения» можно представить как вращение вектора, выходящего из начала координат, на некоторый угол φ , которое матричным выражением записывается в виде:

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \end{bmatrix} \times [x_A \quad y_A \quad 1]^T. \quad (9)$$

Преобразование «масштабирования» с коэффициентами масштаба W , H вдоль осей Ox и Oy можно записать следующим образом:

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} W & 0 & 0 \\ 0 & H & 0 \end{bmatrix} \times [x_A \quad y_A \quad 1]^T. \quad (10)$$

Преобразование «скос», ведущее к искажению геометрии фигуры определяется углами φ и ψ смещения относительно осей координат и записывается матричным выражением:

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} 1 & \operatorname{tg} \varphi & 0 \\ \operatorname{tg} \psi & 1 & 0 \end{bmatrix} \times [x_A \quad y_A \quad 1]^T. \quad (11)$$

При этом суперпозиция операций (8)–(11) примет следующий вид:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} (1 + \operatorname{tg} \varphi \cdot \operatorname{tg} \psi) W \cdot \cos \varphi & -\operatorname{tg} \varphi \cdot H \cdot \sin \varphi & x_c \\ \operatorname{tg} \psi \cdot W \cdot \cos \varphi - H \cdot \sin \varphi & \operatorname{tg} \psi \cdot W \cdot \sin \varphi + H \cdot \cos \varphi & y_c \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (12)$$

где (x, y) – исходные координаты точки, φ – угол преобразования «поворот», W, H – коэффициенты преобразования «масштаб» по оси OX и OY соответственно, φ, ψ – углы преобразования «скос», (x_c, y_c) – координаты переноса центра изображения, (x', y') – конечные координаты точки.

Таким образом, с одной стороны последовательность преобразований с неизвестными параметрами, переводящая сложный сегмент эталона в сложный сегмент сцены, может быть записана в матричной форме (12):

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix}. \quad (13)$$

С другой стороны, на основе гипотез **этапа 3**, возможно точно рассчитать элементы матрицы итоговой трансформации, располагая тремя точками в составе сложного сегмента, используя матричное выражение вида [10]:

$$\begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \times \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \end{bmatrix}. \quad (14)$$

При рассмотрении сложных сегментов большей длины, количество уравнений, описываемых матричной системой (14), увеличится, что приведет к переопределенности матрицы. В таком случае решение системы можно получить методом наименьших квадратов [10].

Далее, рассчитав элементы вектора-столбца правой части выражения (14) для каждой гипотезы, путем выполнения процедуры кластеризации, возможно получить значения центров кластеров, которые будут представлять собой действительные коэффициенты матрицы трансформации. Для этого предлагается использовать подалгоритм кластеризации, напоминающий процесс самоорганизации сети Кохонена [11].

Шаг 0. Входные данные представляют собой набор векторов нормированных значений элементов матрицы (14) $(m_1, m_2, m_3, m_4, m_5, m_6) = (m_{11}, m_{12}, m_{13}, m_{21}, m_{22}, m_{23})$. В момент предъявления первого вектора создается первый кластер.

Шаг 1. Подается случайно выбранный вектор исходного набора, рассчитывается евклидово расстояние от него до центров уже имеющихся к этому моменту кластеров и определяется кластер-победитель для которого расстояние минимально:

$$j^* = \min_j \left(R_j = \sqrt{\sum_{i=1}^6 (m_i - z_{ij})^2} \right), \quad (15)$$

где $(z_{1j}, z_{2j}, z_{3j}, z_{4j}, z_{5j}, z_{6j})$ – координаты центра j -го кластера.

Шаг 2. Осуществляется подтверждение кластера в соответствии с набором правил касательно элементарных аффинных преобразований. Так для каждого формируемого кластера координаты его центра представляют собой коэффициенты правой части выражения (14), приравняв которые в соответствии со своими индексами к соответствующим элементам матрицы с неизвестными значениями элементарных преобразований (12) и решая получающуюся систему уравнений возможно определить значения неизвестных параметров элементарных аффинных преобразований. Что позволяет сформулировать следующие условия.

Условие ограничения параметров масштаба, представляющее собой систему неравенств для значений данных параметров:

$$\begin{cases} W_H \leq W \leq W_B, \\ H_H \leq H \leq H_B, \end{cases} \quad (16)$$

где W_H, W_B, H_H, H_B – нижние и верхние гра-

ницы интервалов, задающих допустимые значения возможных изменений параметров масштаба W , H .

Условие ограничения углов скоса, представляющее собой систему неравенств для значений данных параметров:

$$\begin{cases} \varphi_H \leq \varphi \leq \varphi_B, \\ \psi_H \leq \psi \leq \psi_B, \end{cases} \quad (17)$$

где $\varphi_H, \varphi_B, \psi_H, \psi_B$ – нижние и верхние границы интервалов, задающих допустимые значения возможных изменений углов скоса φ , ψ .

Условие ограничения изменения площади трансформации. Как известно [12], значение $|m_{11} \cdot m_{22} - m_{12} \cdot m_{21}|$ определяет изменение площади трансформированной фигуры, поэтому целесообразно ограничить величину этого изменения:

$$M_1 \leq |m_{11} \cdot m_{22} - m_{12} \cdot m_{21}| \leq M_2, \quad (18)$$

где M_1, M_2 – нижняя и верхняя границы интервала, задающего значение изменения площади фигуры.

Если кластер подтверждается по (16)–(18), то производится коррекция координат его центра:

$$z_{ij^*}^{q+1} = z_{ij^*}^q + \nu \cdot (m_i - z_{ij^*}^q), \quad \forall i = \overline{1, 6}, \quad (19)$$

где ν – параметр коррекции центра кластера, $\nu \in (0, 1)$, различающийся для каждого кластера, и определяемый числом, обратным текущей мощности кластера, q – номер эпохи кластеризации.

Если кластер не подтверждается по (16)–(18), то кластер ликвидируется, а формирующие его элементы перераспределяются заново.

Шаг 3. Шаги 1 и 2 повторяются до тех пор, пока не выполнится хотя бы одно из следующих условий завершения процедуры кластеризации:

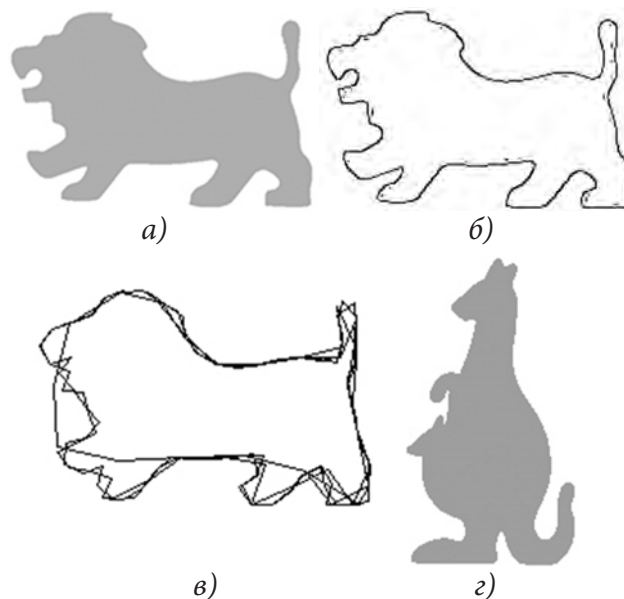
- 1) выполнено заданное число эпох кластеризации;
- 2) на последней эпохе кластеризации координаты центров кластеров существенно не изменились.

Таким образом, центры полученных кластеров описывает трансформацию последовательности сложных сегментов эталона в сложные сегменты сцены. На основе сформированных гипотез, принятие окончатель-

ного решения о наличии эталонного объекта на сцене возможно по пороговому критерию превышения значения площади фигуры, образованной сложными сегментами эталона, на уровне 80–90 %.

ПРИМЕР РАСПОЗНАВАНИЯ В УСЛОВИЯХ ТРАНСФОРМАЦИИ И ПЕРЕКРЫТИЯ ОБЪЕКТОВ

На рис. 5 представлен пример распознавания перекрывающихся объектов предлагаемым алгоритмом. На рис. 5а и 5г приведены эталонные изображения «Лев» и «Кенгуру» соответственно, на рис. 5б – контур изображения «Лев». На рис. 5в представлено семейство сложных сегментов изображения «Лев», полученных на этапе 1 в соответствии с выражениями (1)–(3). На рис. 5д представлено изображение «Сцена», полученное при перекрытии объектов «Лев» и «Кенгуру». На рис. 5е показано семейство сложных сегментов изображения «Сцена», полученное на этапе 2. На рис. 5ж совместно приведены сформированные на этапе 3 в соответствии с (8)–(19) кластеры изображений «Лев» и «Сцена». На рис. 5з представлен окончательный результат распознавания эталонных изображений «Лев» и «Кенгуру» в виде выделенных на изображении «Сцена» контуров.



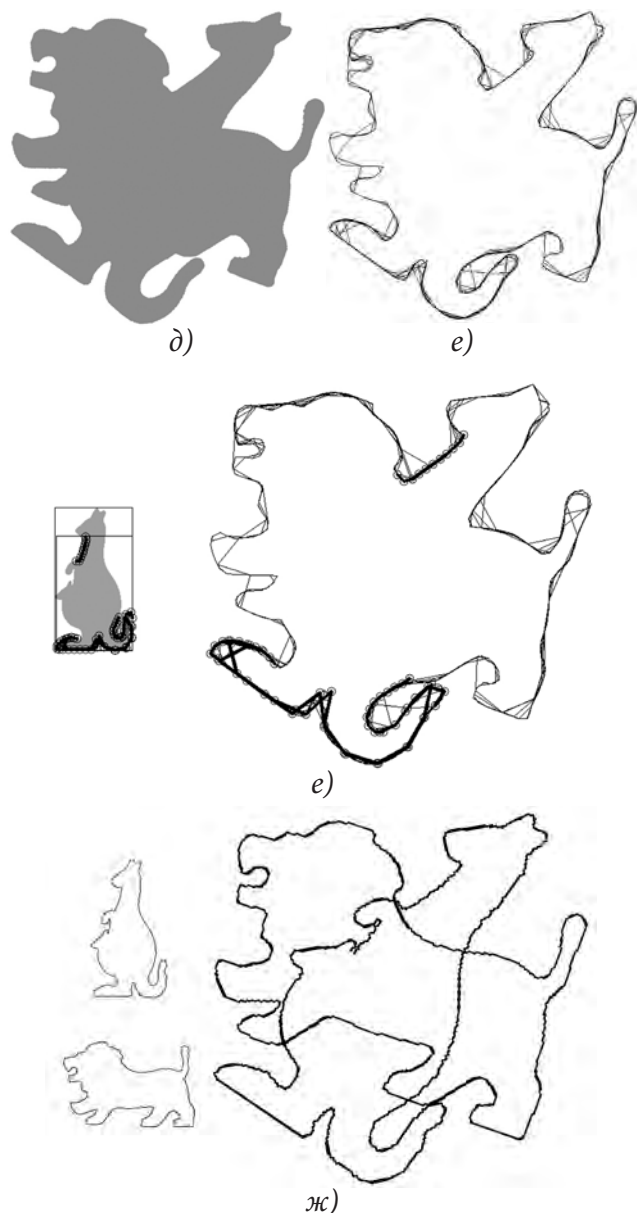


Рис. 5. Пример работы предлагаемого алгоритма: а) эталонное изображение «лев»; б) контур изображения «лев»; в) семейство сложных сегментов изображения «лев»; г) эталонное изображение «кенгуру»; д) изображение «сцена»; е) семейство сложных сегментов изображения «сцена»; ж) кластеры изображений «лев» и «сцена»; з) результат распознавания обоих объектов по эталонам

Из представленных рис. 5а–5з видно, что в случае перекрытия распознаваемых образов предлагаемый алгоритм успешно обнаруживает оба объекта по их эталонным изображениям. При этом сами распознаваемые объекты могут подвергаться аффинным трансформациям масштабирования и вращения.

ЗАКЛЮЧЕНИЕ

Таким образом, результаты численных исследований отдельных этапов алгоритма и пример распознавания перекрывающихся объектов показывают повышение устойчивости процедуры распознавания к аффинным преобразованиям объектов на плоских цифровых изображениях.

Предлагаемый алгоритм отличается от известных модификацией процедуры построения сегментов КП, формированием семейства сложных сегментов за счет циклического сдвига начальной точки и использования процедуры многомасштабности, а также введением процедуры кластеризации в пространстве коэффициентов неизвестного аффинного преобразования в виде суперпозиции элементарных, по средствам расчета параметров которых выполняется процедура верификации кластеров. Итоговое принятие решения осуществляется по критерию покрытия площади эталонного изображения сложными сегментами, восстановленными на основе гипотез соответствия, доступных через структуру элементов кластеров.

Практическая значимость предложенного алгоритма заключается в возможности его использования в различных системах технического зрения при распознавании перекрывающихся объектов на плоских цифровых изображениях, регистрируемых в том числе в условиях помех и потери части информации.

СПИСОК ЛИТЕРАТУРЫ

1. Дуда Р., Харт П. Распознавание образов и анализ сцен. – М. : Мир, 1976. – 507 с.
2. Stein F., Medioni G. Structural indexing: efficient 2-D object recognition // IEEE Transactions on pattern analysis and machine intelligence. 1992. – Vol. 14. – N. 12. – P. 1198–1204.
3. Михайлов Б. Б. Техническое зрение мобильных роботов // Техническое зрение в системах управления мобильными объектами-2010: Труды научно-технической конференции-семинара. Вып. 4 / Под ред. Р. Р. Назирова. – М. : КДУ, 2011. – С. 191–201.

4. Васильев Д. В. Фрагменты прикладной теории систем технического зрения для беспилотных летательных аппаратов // Техническое зрение в системах управления мобильными объектами-2010: Труды научно-технической конференции-семинара. Вып. 4 / Под ред. Р. Р. Назирова. – М. : КДУ, 2011. – С. 109–131.
5. Пантюхин М. А., Самойлин Е. А. Алгоритм адаптивного градиентного выделения контуров объектов на цифровых изображениях, регистрируемых в условиях деструктивных лазерных помех // Радиотехника. 2015. – №12. – С. 60–65.
6. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. – 1072 с.
7. Wenzhong Shi, ChuiKwan Cheung. Performance evaluation of line simplification algorithms for vector generalization // The Cartographic Journal, 2006. – Vol. 43. – N. 1. – P. 27–44.
8. Аршинов М. Н., Садовский Л. Е. Коды и математика. – М. : Наука, Главная редакция физико-математической литературы, 1983. – 144 с.
9. Яглом И. М., Ашкинзуе В. Г. Идеи и методы аффинной и проективной геометрии. I часть Аффинная геометрия. – М. : Учпедгиз, 1962. – 248 с.
10. Prince S. Computer vision: models, learning and inference. Cambridge: Cambridge University Press, 2012. – 667 p.
11. Осовский С. Нейронные сети для обработки информации / Пер. с польского И. Д. Рудинского. – М. : Финансы и статистика, 2002. – 344 с.
12. Постников М. М. Аналитическая геометрия. – М. : Наука, 1973 г. – 754 с.

Пантюхин Максим Александрович – адъюнкт ВУНЦ ВВС «Военно-воздушная академия им. проф. Н. Е. Жуковского и Ю. А. Гагарина».
E-mail: ol-max@mail.ru

Самойлин Евгений Александрович – д.т.н., доцент, профессор кафедры ВУНЦ ВВС «Военно-воздушная академия им. проф. Н.Е. Жуковского и Ю.А. Гагарина».
E-mail: es977@mail.ru

Дроздов Алексей Юрьевич – инженер-программист ООО «Геоиндустрия».
E-mail: alexey_drozdov@mail.ru

Pantyukhin Maksim A. – post-graduate student Air Force Education and Research Center «The Zhukovsky and Gagarin Air Force Academy».
E-mail: ol-max@mail.ru

Samoylin Evgeniy A. – D.Sc in Engineering, assistant professor, professor Air Force Education and Research Center «The Zhukovsky and Gagarin Air Force Academy».
E-mail: es977@mail.ru

Drozdov Alexey Yu. – software engineer Ltd «Geoindustry».
E-mail: alexey_drozdov@mail.ru