



<http://dx.doi.org/10.35596/1729-7648-2020-18-2-45-52>

Оригинальная статья  
Original paper

УДК 004.932+004.258

## ОПРЕДЕЛЕНИЕ РАЗМЕРА LIFO-СТЕКА ДЛЯ ВЫРАЩИВАНИЯ ОБЛАСТЕЙ ИЗОБРАЖЕНИЙ

ЦВЕТКОВ В.Ю.

*Белорусский государственный университет информатики и радиоэлектроники  
(г. Минск, Республика Беларусь)*

*Поступила в редакцию 26 января 2020*

© Белорусский государственный университет информатики и радиоэлектроники, 2020

**Аннотация.** Рассматривается задача выделения памяти для организации LIFO-стека в алгоритме сегментации изображений на основе выращивания областей. Сегментация разделяет изображение на области с одинаковыми или схожими свойствами и является наиболее требовательным к емкости оперативной памяти процессом. Выращивание областей начинается с окрестностей предварительно выделенных начальных пикселей роста и использует стеки для хранения координат смежных пикселей, присоединяемых к выращиваемой области. Загрузка стеков максимальна, когда размер сегмента совпадает с размером  $YX$  изображения. При отсутствии выражения для точного определения размера стека гарантировать устойчивую работу алгоритма выращивания областей, исключающую переполнение выделенной для обработки памяти, можно, только если размер стека принять равным  $YX$ . Однако этот подход не учитывает то, что заполнение стеков координат сопровождается также выборкой из них, вследствие чего размер стека всегда меньше  $YX$ . В статье предлагается выражение, позволяющее повысить точность определения необходимого размера LIFO-стека для хранения координат смежных пикселей в зависимости от размера изображения. Выражение учитывает условия максимальной загрузки LIFO-стека, когда: а) осуществляется сегментация квадратной области с начальным пикселем роста в углу этой области; б) в окне сканирования смежные пиксели всегда выбираются по порядку с расположением первого выбираемого пикселя в углу окна сканирования. Использование предложенного выражения для расчета необходимой емкости LIFO-стека в условиях его максимальной загрузки в алгоритме сегментации изображений на основе выращивания областей обеспечивает уменьшение числа ячеек памяти LIFO-стека в 2 раза.

**Ключевые слова:** LIFO-стек, сегментация изображений, выращивание областей, размер стека, выделение памяти.

**Конфликт интересов.** Автор заявляет об отсутствии конфликта интересов.

**Для цитирования.** Цветков В.Ю. Определение размера LIFO-стека для выращивания областей изображений. Доклады БГУИР. 2020; 18(2): 45-52.

## LIFO-STACK SIZE DETERMINATION FOR GROWING OF THE IMAGE REGIONS

VIKTAR YU. TSVIATKOU

*Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)*

*Submitted 26 Januar 2020*

© Belarusian State University of Informatics and Radioelectronics, 2020

**Abstract.** This paper considers the problem of memory allocation for the organization of the LIFO-stack in the algorithm for image segmentation based on growing regions is considered. Segmentation divides the image into regions with identical or similar properties and is the most demanding process for the capacity of RAM. The cultivation of areas begins with the neighborhoods of pre-selected initial growth pixels and uses stacks to store the coordinates of adjacent pixels attached to the cultivated region. Stack loading is maximized when the segment size matches the size of the YX image. In the absence of an expression for the exact determination of the size of the stack, it is possible to guarantee the stable operation of the algorithm for growing regions, eliminating the overflow of the memory allocated for processing if the stack size is assumed equal to YX. However, this approach does not take into account the fact that filling the coordinate stacks is also accompanied by a selection of them, which makes the stack size always smaller than YX. The article proposes an expression that allows one to increase the accuracy of determining the required size of the LIFO-stack for storing the coordinates of adjacent pixels depending on the image size. The expression takes into account the conditions of the maximum load of the LIFO-stack when: a) the segmentation of the square region with the initial growth pixel in the corner of this region is carried out; b) in the scan window, adjacent pixels are always selected in order with the first selectable pixel located in the corner of the scan window. Using the proposed expression to calculate the required capacity of the LIFO-stack under conditions of its maximum load in the image segmentation algorithm based on growing regions provides a 2-fold reduction in the number of LIFO-stack memory cells.

**Keywords:** LIFO-stack, image segmentation, region growing, stack size, memory allocation.

**Conflict of interests.** The author declares no conflict of interests.

**For citation.** Tsviatkou V.Yu. LIFO-stack size determination for growing of the image regions. Doklady BGUIR. 2020; 18(2): 45-52.

### Введение

Одна из важных задач, решаемых в процессе разработки программного обеспечения системы обработки изображений, связана с планированием расхода оперативной памяти. Особенно требовательны к емкости оперативной памяти алгоритмы сегментации. Сегментация разделяет изображение на области с одинаковыми или схожими свойствами [1–4]. Ее результатом является матрица сегментации, совпадающая по размеру с исходным изображением, значение каждого элемента которой указывает на номер сегмента, которому принадлежит соответствующий пиксель исходного изображения. Простейшие алгоритмы сегментации основаны на выращивании областей в окрестности предварительно выделенных начальных пикселей роста и используют стеки для хранения координат смежных пикселей, присоединяемых к выращиваемой области [2, 5–7]. Стеки могут быть организованы по принципу FIFO (первым зашел – первым вышел) или LIFO (последним зашел – первым вышел) [8–10]. Они отличаются порядком выборки данных и, соответственно, емкостью, необходимой для хранения координат пикселей при максимальной загрузке. Несмотря на отсутствие универсального подхода к определению необходимой емкости стека для различных алгоритмов, достаточно точные ее значения могут быть получены для конкретного алгоритма в предельных условиях.

Целью работы является определение размера LIFO-стека координат смежных пикселей в зависимости от размера изображения для алгоритма сегментации на основе выращивания областей.

### Особенности сегментации изображений на основе выращивания областей

В результате сегментации изображения  $I = \|i(y, x)\|_{(y=0, \overline{Y-1}, x=0, \overline{X-1})}$  размером  $Y \times X$  пикселей формируется матрица  $S = \|s(y, x)\|_{(y=0, \overline{Y-1}, x=0, \overline{X-1})}$  сегментации такого же размера, значения элементов которой указывают на номера  $n_s \in [0, N_s]$  сегментов, которым они принадлежат, где  $N_s$  – число сегментов и номер последнего сегмента. На рис. 1 приведены фрагмент полутонового изображения, соответствующие ему матрица значений пикселей и матрица сегментации, содержащая 6 сегментов.

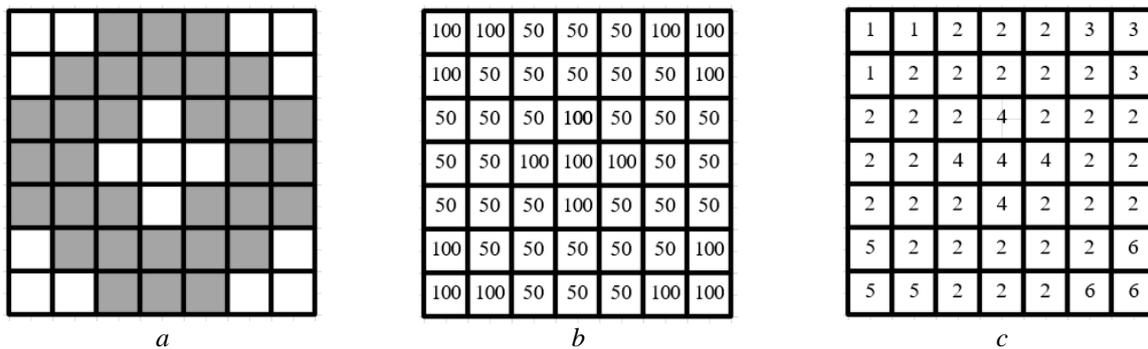


Рис. 1. Фрагменты: *a* – полутоновое изображение; *b* – матрица пикселей; *c* – матрица сегментации  
Fig. 1. Fragments: *a* – grayscale image; *b* – matrix of pixels; *c* – segmentation matrix

Алгоритм выращивания областей описывает следующую последовательность операций по сегментации изображения (предполагается, что начальные пиксели роста на изображении уже выделены) [1–4]:

1) координаты очередного начального пикселя роста помещаются в стеки  $Y_F = \|y_F(p_F)\|_{(p_F=0, \overline{P_F-1})}$  и  $X_F = \|x_F(p_F)\|_{(p_F=0, \overline{P_F-1})}$  координат смежных пикселей, где  $p_F$  – указатель стеков;  $P_F$  – максимально возможное число координат в стеках  $Y_F$  и  $X_F$ ;

2) определяется текущее значение  $n_s$  номера сегмента;

3) реализуется цикл обработки стеков  $Y_F$  и  $X_F$ , в котором из них извлекаются координаты  $(y, x)$  очередного обрабатываемого пикселя; соответствующему элементу матрицы сегментации  $s(y, x)$  присваивается значение  $n_s$  номера сегмента; проверяется выполнение условия присоединения к текущему пикселю  $(y, x)$  смежных пикселей с координатами  $(y + y', x + x')$  при  $(y' = -1, 1) \wedge (x' = -1, 1) \wedge ((y' \neq 0) \vee (x' \neq 0))$ ; если для какого-либо смежного пикселя условие выполняется, то его координаты  $(y + y', x + x')$  заносятся в стеки  $Y_F$  и  $X_F$ ;

4) шаги 1 – 3 повторяются до тех пор, пока все начальные пиксели роста не будут обработаны.

### Постановка задачи

Предположим, что стеки  $Y_F$  и  $X_F$  координат смежных пикселей в алгоритме выращивания областей относятся к типу LIFO. Их загрузка максимальна, когда размер

сегмента совпадает с размером изображения. В этом случае  $N_s = 1, \forall y \forall x (s(y, x) = N_s)$ , размер  $R_s(n_s)$   $n_s$ -го сегмента, определяемый с помощью выражения  $R_s(n_s) = \frac{1}{n_s} \sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} s(y, x)$ , составляет  $YX$  ( $R_s(1) = YX$ ) при исключении повторного попадания координат в стеки (за счет проверки значений в матрице сегментации). При отсутствии выражения для точного определения размера стека (значения  $P_F$  числа ячеек сетка) гарантировать устойчивую работу алгоритма выращивания областей, исключающую переполнение стеков  $Y_F$  и  $X_F$  координат, можно, только если

$$P_F = YX. \quad (1)$$

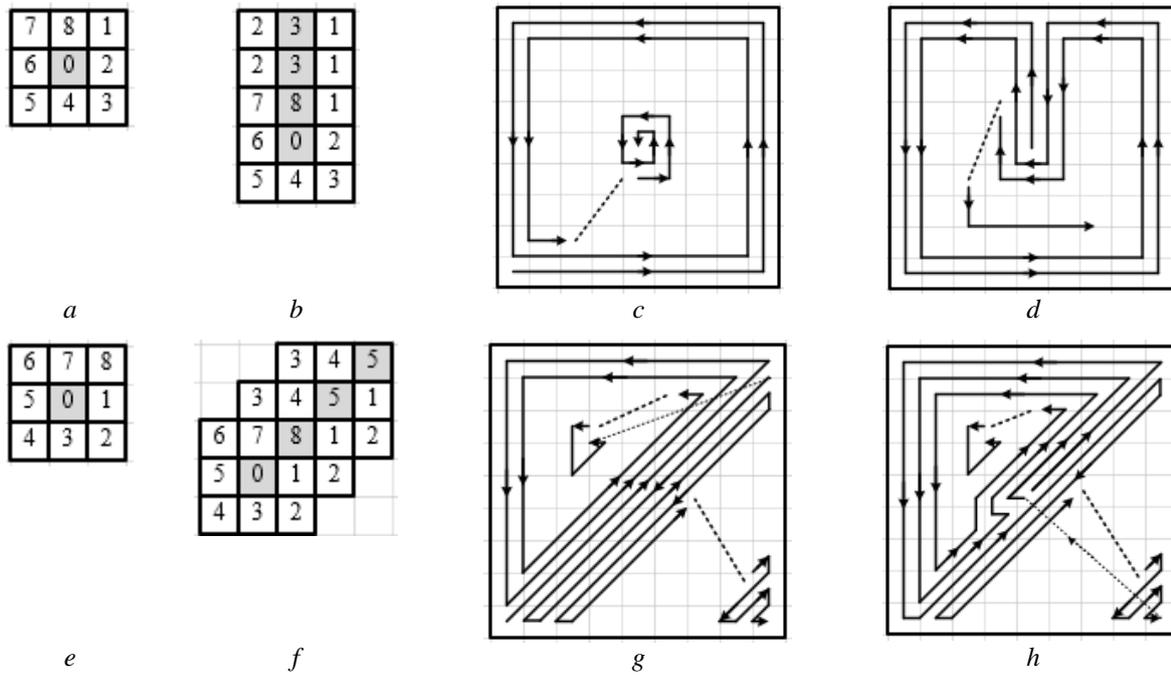
Это выражение не учитывает то, что заполнение стеков координат сопровождается выборкой из них данных, вследствие чего всегда выполняется неравенство  $P_F < YX$ . Таким образом, для повышения эффективности использования оперативной памяти необходимо точно определить размер  $P_F$  LIFO-стеков  $Y_F$  и  $X_F$  координат смежных пикселей при максимальной загрузке – при размере изображения  $Y \times X$  пикселей, наличии на нем одного сегмента ( $N_s = 1$ ), размер которого  $R_s(1)$  совпадает с размером изображения ( $R_s(1) = YX$ ).

#### Вывод выражений для уточнения размера LIFO-стека координат смежных пикселей при максимальной загрузке

Пусть квадратное изображение размером  $Y^2$  пикселей содержит единственный сегмент такого же размера. Число координат, загружаемых в LIFO-стек при обработке окрестности текущего пикселя, зависит от порядка выборки смежных пикселей и положения текущего обрабатываемого пикселя относительно границ сегмента.

Текущий и смежные с ним пиксели покрываются квадратным окном сканирования размером  $3 \times 3$  пикселя. Возможны два базовых варианта выборки 8 смежных пикселей в окне сканирования, когда координаты последнего загружаемого в стек пикселя указывают на сторону (например, верхнюю, рис. 2, а) или угол (например, верхний правый, рис. 2, е) квадрата, определяющего окно сканирования. Всего существуют 16 вариантов последовательной выборки смежных пикселей с учетом движения по и против часовой стрелки. Если окно сканирования не находится на границах сегмента, выборка смежных пикселей при расположении в углу окна сканирования первого выбираемого пикселя приводит к вертикальному или горизонтальному перемещению окна сканирования (например, вверх для окна сканирования с расположением первого выбираемого пикселя в правом верхнем углу, рис. 2, б), а при расположении в углу окна сканирования последнего выбираемого пикселя – диагональному перемещению окна сканирования (например, вверх и вправо для окна сканирования с расположением последнего выбираемого пикселя в правом верхнем углу, рис. 2, ф).

Для каждого из 16 вариантов последовательной выборки смежных пикселей существуют  $Y^2$  вариантов траекторий перемещения окна сканирования в пределах границ квадратного сегмента размером  $Y^2$  пикселей в зависимости от положения начального пикселя роста сегмента (начального положения окна сканирования). На рис. 2 приведены примеры траекторий перемещения окна сканирования в пределах границ квадратного сегмента для двух вариантов выборки смежных пикселей в окне сканирования, приведенных на рис. 2, а, е, когда положение начального пикселя роста соответствует нижнему левому углу (рис. 2, с, г) и центру (рис. 2, д, h) сегмента.



**Рис. 2.** Фрагменты: *a* – окно сканирования с расположением первого выбираемого пикселя в правом верхнем углу; *b* – перемещение окна сканирования вверх; *c* – перемещение окна сканирования в пределах сегмента при расположении первого выбираемого пикселя в правом верхнем углу окна сканирования и начального пикселя роста в нижнем левом углу сегмента; *d* – перемещение окна сканирования в пределах сегмента при расположении первого выбираемого пикселя в правом верхнем углу окна сканирования и начального пикселя роста в центре сегмента; *e* – окно сканирования с расположением последнего выбираемого пикселя в правом верхнем углу; *f* – перемещение окна сканирования вверх и вправо; *g* – перемещение окна сканирования в пределах сегмента при расположении последнего выбираемого пикселя в правом верхнем углу окна сканирования и начального пикселя роста в нижнем левом углу сегмента; *h* – перемещение окна сканирования в пределах сегмента при расположении последнего выбираемого пикселя в правом верхнем углу окна сканирования и начального пикселя роста в центре сегмента

**Fig. 2.** Fragments: *a* – scanning window with the location of the first selectable pixel in the upper right corner; *b* – move the scan window up; *c* – moving the scan window within the segment when the first selectable pixel is located in the upper right corner of the scan window and the initial growth pixel in the lower left corner of the segment; *d* – moving the scan window within the segment when the first selectable pixel is located in the upper right corner of the scan window and the initial growth pixel in the center of the segment; *e* – scanning window with the location of the last selectable pixel in the upper right corner; *f* – move the scan window up and right; *g* – moving the scanning window within the segment when the last selected pixel is located in the upper right corner of the scanning window and the initial growth pixel in the lower left corner of the segment; *h* – moving the scan window within the segment when the last selectable pixel is located in the upper right corner of the scan window and the initial growth pixel in the center of the segment

Для траектории перемещения окна сканирования, приведенной на рис. 2, *c*, необходимо  $\lfloor Y/4 \rfloor$  циклов, в каждом из которых обрабатываются  $4(Y-4k)$  пикселей, где  $\lfloor \cdot \rfloor$  – операция округления с недостатком;  $k=1, \lfloor Y/4 \rfloor$  – номер цикла. При каждом перемещении окна сканирования из стека извлекаются координаты текущего пикселя и помещаются в стек координаты двух смежных пикселей, в результате чего число координат в стеке увеличивается на единицу. При обработке первого пикселя в левом нижнем углу сегмента в стек помещаются координаты еще одного пикселя. Таким образом, размер  $P_F$  LIFO-стека координат смежных пикселей для варианта траектории окна сканирования, приведенной на рис. 2, *c*, определяется с помощью выражения

$$P_F = 1 + 4 \sum_{k=1}^{\lfloor Y/4 \rfloor} (Y - 4k). \quad (2)$$

Для траектории перемещения окна сканирования, приведенной на рис. 2, *d*, необходимо  $\lfloor Y/8 \rfloor$  циклов, в каждом из которых обрабатываются  $5(Y-8k)$  пикселей ( $4(Y-8k)$  по периметру сегмента и  $(Y-8k)$  в центральной части сегмента), где  $k=1, \lfloor Y/8 \rfloor$ . При каждом перемещении окна сканирования из стека извлекаются координаты текущего пикселя и помещаются в стек координаты двух смежных пикселей, в результате чего число координат в стеке увеличивается на единицу. При обработке первого пикселя в центре сегмента в стек помещаются координаты семи пикселей, а затем, при каждом перемещении окна сканирования вверх, в стек добавляются координаты двух пикселей (всего  $Y$  пикселей). При обработке пикселей в центральной части сегмента в каждом цикле в стек помещаются координаты еще 10 пикселей. Таким образом, размер  $P_F$  LIFO-стека координат смежных пикселей для варианта траектории окна сканирования, приведенной на рис. 2, *d*, определяется с помощью выражения

$$P_F = 7 + Y + 5 \sum_{k=1}^{\lfloor Y/8 \rfloor} (Y - 8k) + 10 \left\lfloor \frac{Y}{8} \right\rfloor. \quad (3)$$

Траектория перемещения окна сканирования, приведенная на рис. 2, *g*, состоит из двух фрагментов. При переходе от обработки первого фрагмента ко второму большая часть координат смежных пикселей из стека выгружается.

Для первого фрагмента необходимо  $\lfloor Y/5 \rfloor$  циклов. При каждом перемещении окна сканирования вверх из стека извлекаются координаты текущего пикселя и помещаются в стек координаты 5 смежных пикселей в первом цикле (число координат в стеке увеличивается на 4) и 3-х пикселей в последующих циклах (число координат в стеке увеличивается на 2). При каждом перемещении окна сканирования влево и вниз из стека извлекаются координаты текущего пикселя и помещаются в стек координаты двух смежных пикселей, в результате чего число координат в стеке увеличивается на единицу. При этом число координат смежных пикселей в LIFO-стеке определяется выражением

$$P_F = 4Y + 2 \sum_{k=1}^{\lfloor Y/5 \rfloor} (Y - 5k) + 2 \sum_{k=1}^{\lfloor Y/5 \rfloor} (Y - 4k). \quad (4)$$

Для второго фрагмента необходимо  $\lfloor Y/3 \rfloor$  циклов. При каждом перемещении окна сканирования из стека извлекаются координаты текущего пикселя и помещаются в стек координаты 3-х смежных пикселей, в результате чего число координат в стеке увеличивается на 2. С учетом того, что после обработки первого фрагмента в LIFO-стеке остаются координаты  $4Y$  смежных пикселей размер LIFO-стека для второго фрагмента определяется выражением

$$P_F = 4Y + 2 \sum_{k=1}^{\lfloor Y/3 \rfloor} (Y - 3k). \quad (5)$$

Траектория перемещения окна сканирования, приведенная на рис. 2, *h*, состоит из двух фрагментов. При переходе от обработки первого фрагмента ко второму большая часть координат смежных пикселей из стека выгружается.

Первый фрагмент включает перемещение окна сканирования относительно центра сегмента вверх (при каждом перемещении число координат в стеке увеличивается на 4), влево (число координат в стеке увеличивается на единицу), вниз (число координат в стеке увеличивается на единицу) и еще  $\lfloor Y/3 \rfloor$  циклов перемещения вверх и вниз (число координат в стеке увеличивается на 2). Размер LIFO-стека для первого фрагмента определяется выражением

$$P_F = 6Y + 2 \sum_{k=1}^{\lfloor Y/3 \rfloor} (Y - 3k). \quad (6)$$

Для второго фрагмента необходимо  $\lfloor Y/5 \rfloor$  циклов. При каждом перемещении окна сканирования вверх из стека извлекаются координаты текущего пикселя и помещаются в стек координаты 3-х пикселей (число координат в стеке увеличивается на 2). При каждом перемещении окна сканирования влево и вниз из стека извлекаются координаты текущего пикселя и помещаются в стек координаты двух смежных пикселей, в результате чего число координат в стеке увеличивается на единицу. С учетом того, что после обработки первого фрагмента в LIFO-стеке остаются координаты  $6Y$  смежных пикселей, размер LIFO-стека для второго фрагмента определяется выражением

$$P_F = 6Y + 2 \sum_{k=1}^{\lfloor Y/5 \rfloor} (Y - 5k) + 2 \sum_{k=1}^{\lfloor Y/5 \rfloor} (Y - 4k). \quad (7)$$

### Оценка экономии памяти при уточненном расчете размера LIFO-стека

На рис. 3 приведены зависимости размера  $P_F^{(E)}$  (число ячеек памяти) LIFO-стека, определяемого с помощью выражений (1) – (7), от размера  $Y^2$  (число пикселей) квадратного изображения, где  $E$  – номер выражения. Уточненный размер  $P_F$  LIFO-стека координат смежных пикселей определяется одним из выражений (2) – (7), обеспечивающим наибольшие значения, что соответствует максимальной загрузке стека. Из рис. 1 следует, что наибольшие значения дает выражение (2), что соответствует перемещению окна сканирования в пределах сегмента по траектории, приведенной на рис. 2, с. Таким образом, максимальная загрузка стека обеспечивается при расположении первого выбираемого пикселя в углу окна сканирования и начального пикселя роста в углу области сканирования. Траектория перемещения окна сканирования по сегментируемой области в этом случае представляет собой спираль.



Рис. 3. Зависимости размера LIFO-стека от размера изображения  
Fig. 3. Dependencies of the size of the LIFO stack on the size of the image

Экономия  $M_E$  памяти при уточненном расчете необходимой емкости LIFO-стека по выражению (2) в сравнении с приблизительным расчетом по выражению (1) для квадратного изображения размером  $Y^2$  пикселей определяется с помощью выражения

$$M_E = \frac{Y^2}{P_F^{(2)}}. \quad (8)$$

Из рис. 3 и выражения (8) следует, что экономия памяти при уточненном расчете необходимой емкости LIFO-стека составляет 2 раза.

В случае сегментации прямоугольного изображения размером  $Y \times X$  пикселей для определения необходимой емкости LIFO-стека по выражению (2) и экономии памяти по выражению (8) предлагается использовать эквивалентный размер квадратного изображения  $Y^2$  пикселей такой, что  $Y^2 \geq Y \times X$ .

### Заключение

Получено выражение, позволяющее повысить точность определения необходимого размера LIFO-стека для хранения координат смежных пикселей в зависимости от размера изображения для алгоритма сегментации на основе выращивания областей. При получении данного выражения приняты условия максимальной загрузки LIFO-стека, когда: а) осуществляется сегментация квадратной области с начальным пикселем роста в углу этой области; б) в окне сканирования смежные пиксели всегда выбираются по порядку с расположением первого выбираемого пикселя в углу окна сканирования. Траектория перемещения окна сканирования по сегментируемой области в этом случае представляет собой спираль. Использование предложенного выражения для расчета необходимой емкости LIFO-стека в условиях его максимальной загрузки в алгоритме сегментации изображений на основе выращивания областей обеспечивает уменьшение числа ячеек памяти в 2 раза.

### Список литературы / References

1. Otsu N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*. 1979;9:62-66. DOI:10.1109/TSMC.1979.4310076.
2. Haralick R.M., Shapiro L.G. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*. 1985;29(1):100-132. DOI:10.1016/S0734-189X(85)90153-7.
3. Horowitz S.L., Pavlidis T. Picture segmentation by a tree traversal algorithm. *Journal of the ACM*. 1976;23(2):368-388. DOI:10.1145/321941.321956.
4. Meyer F. Topographic distance and watershed lines. *Signal Processing*. 1994;38(1):113-125. DOI:10.1016/0165-1684(94)90060-4.
5. Moigne J.Le., Tilton J.C. Refining image segmentation by integration of edge and region data. *IEEE transactions on geoscience and remote sensing*. 1995;33(3):605-614. DOI:10.1109/36.387576.
6. Hojjatoleslami S.A., Kittler J. Region Growing: A New Approach. *IEEE Transactions on Image Processing*. 1998;7(7):1079-1084. DOI:10.1109/83.701170.
7. Cui W., Guan Z., Zhang Z. An Improved Region Growing Algorithm for Image Segmentation. International Conference on Computer Science and Software Engineering. 2008;6:93-96. DOI:10.1109/CSSE.2008.891.
8. Wirth N. *Algorithms and Data Structures*. Upper Saddle River: Prentice Hall. 1985.
9. Lipschutz S. *Schaum's Outline of Theory and Problems of Data Structures*. New York City: McGraw-Hill Book Company; 1986.
10. Kruse R.L., Ryba A.J. *Data Structures and Program Design in C++*. Upper Saddle River: Prentice Hall; 2000.

#### Сведения об авторах

Цветков В.Ю., д.т.н., доцент, заведующий кафедрой инфокоммуникационных технологий Белорусского государственного университета информатики и радиоэлектроники.

#### Information about the authors

Tsviatkou V.Yu., PhD, Associate Professor, Head of Infocommunication Department of Belarusian State University of Informatics and Radioelectronics.

#### Адрес для корреспонденции

220013, Республика Беларусь,  
г. Минск, ул. П. Бровки, д. 6,  
Белорусский государственный университет  
информатики и радиоэлектроники  
+375-017-293-84-08;  
e-mail: vtsvet@bsuir.by  
Цветков Виктор Юрьевич

#### Address for correspondence

220013, Republic of Belarus,  
Minsk, P. Brovki str., 6,  
Belarusian State University  
of Informatics and Radioelectronics  
+375-017-293-84-08;  
e-mail: vtsvet@bsuir.by  
Tsviatkou Viktor Yur'evich