

V. G. Zhukov, M. N. Zhukova, N. A. Koromyslov

## APPLICATION OF FUZZY ARTIFICIAL IMMUNE SYSTEMS IN THE TASK OF CONSTRUCTION OF ADAPTIVE SELF-LEARNING INFORMATION SECURITY SYSTEM

*We consider the application of principles of immune system functioning in the task of construction of adaptive self-organizing security tools and application of fuzzy logic for its improvement.*

*Keywords: artificial immune systems, fuzzy logic, information security.*

© Жуков В. Г., Жукова М. Н., Коромыслов Н. А., 2012

УДК 004.932

А. Г. Зотин, А. В. Носов, Д. В. Бузаев

## АНАЛИЗ ПРИГОДНОСТИ МЕТОДОВ СЕГМЕНТАЦИИ ДЛЯ ЛОКАЛИЗАЦИИ ОБЪЕКТОВ НА ОСНОВЕ ЦВЕТОВЫХ И СТРУКТУРНЫХ ПРИЗНАКОВ

*Рассмотрены существующие методы сегментации по цветовым и структурным признакам. Приведено описание алгоритмов функционирования наиболее широко известных методов сегментации на основе пороговой обработки, графов и каскадов. Представлены результаты анализа пригодности методов сегментации для локализации объектов на примере растений, лиц и кистей рук людей.*

*Ключевые слова: анализ изображений, сегментация, пороговая обработка, алгоритм Краскала, каскады Хаара.*

Развитие компьютерной техники делает возможной сложную обработку видеоданных, приближенную к реальному времени. В обработке изображений при распознавании образов одной из самых важных задач является задача обнаружения и локализации объектов интереса, возникающих в таких сферах, как мониторинг, охрана, управление и т. д. Среди наиболее активно развивающихся направлений можно выделить распознавание лиц людей и их жестов, применяемое в сфере охраны и управления, мониторинг растений и природных ресурсов в сельском хозяйстве.

Существует множество методов сегментации [1]. К наиболее распространенным методам относятся модификации методов пороговой обработки, сегментация на основе теории графов и на основе классификаторов (каскадов). Рассмотрим их более подробно.

В большинстве случаев, когда объекты интереса имеют определенный цвет, для их локализации используется цветовая сегментация, суть которой заключается в выделении областей, имеющих цвет объекта интереса. Наиболее широко известны и легко реализуемы методы, основанные на пороговой обработке, результатом которых является матрица, размеры которой совпадают с размерами изображения [2].

Сегментацию по порогу можно проводить разными способами. Самым простым является сравнение каждого пикселя с некоторым значением (порогом):

$$M(x, y) = \begin{cases} 1, & \text{если } I(x, y) \geq \text{Threshold}, \\ 0, & \text{если } I(x, y) < \text{Threshold}, \end{cases}$$

где  $I(x, y)$  – яркость изображения в точке  $(x, y)$ ;  $M$  – результирующая маска;  $\text{Threshold}$  – значение порога.

Другой способ заключается в поиске близких по цвету пикселей с эталонным значением. Для этого каждый пиксель изображения рассматривается в выбранных цветовых пространствах, после чего вычисляется расстояние между ним и эталоном и принимается решение о принадлежности пикселя объекту. Однако следует обратить внимание на то, что в зависимости от цветовой модели цвет объекта может иметь различные представления в различных цветовых пространствах. Качество сегментации зависит от того, в какой модели будет проводиться обработка изображений:

$$M(x, y) = \begin{cases} 1, & \text{если } d(M(x, y), \text{Etalon}) \geq \text{Threshold}, \\ 0, & \text{если } d(M(x, y), \text{Etalon}) < \text{Threshold}, \end{cases}$$

где  $I(x, y)$  – значение изображения выбранного цветового пространства в точке  $(x, y)$ ;  $M$  – результирующая маска;  $d(X, Y)$  – функция, вычисляющая расстояние между векторами  $X$  и  $Y$ ;  $\text{Etalon}$  – значение координат эталонного цвета в рассматриваемом цветовом пространстве;  $\text{Threshold}$  – значение порога.

Для оценки расстояния между эталонным и проверяемым цветом используют метрики. *Метрика* – это функция, определяющая расстояния в метрическом пространстве, т. е. во множестве, в котором определено расстояние между любой парой элементов. Наиболее популярными метриками являются:

– евклидова метрика – обычное расстояние между двумя точками:

$$d(X, Y) = \sqrt{(X - Y)^T \cdot (X - Y)},$$

где  $X$  и  $Y$  – сравниваемые векторы;

– метрика Махаланобиса:

$$d(X, Y) = \sqrt{(X - Y)^T \cdot C^{-1} \cdot (X - Y)},$$

где  $C$  – ковариационная матрица;  $X$  и  $Y$  – сравниваемые векторы;

– среднеквадратичная ошибка MSE (Mean Square Error):

$$d(X, Y) = \frac{1}{N} (X - Y)^T \cdot (X - Y),$$

где  $N$  – число элементов массива;  $X$  и  $Y$  – сравниваемые векторы.

Анализ данных о работе пороговой сегментации растений и кожи людей в моделях HSV и RGB показал, что значения каналов V, R, G и B лежат практически во всем диапазоне. Поэтому для более качественной сегментации по цветовым признакам используется модификация пороговой сегментации – многокомпонентная сегментация, которая заключается в назначении приоритета на составляющие выбранных цветовых моделей. К примеру, если сегментация зеленых растений проводится в моделях RGB и HSV, то эффективные результаты будет давать сегментация в каналах H, S и G, поэтому им следует задавать более высокие приоритеты, а каналам V, R и B – более низкие. Назначать приоритеты можно по формуле

$$\Lambda = \frac{\sum (k_i \cdot M_i)}{\sum k_i},$$

где  $\Lambda$  – результирующая маска;  $k_i$  – приоритет  $i$ -го канала;  $M_i$  – маска, полученная в результате сегментации изображения в  $i$ -м канале.

В тех случаях когда цвет объекта интереса может быть произвольным, применяется сегментация на основе теории графов, основная идея которой заключается в поиске и маркировке однородных по цвету участков [3].

Одним из алгоритмов такой сегментации является алгоритм Краскала, основанный на построении минимального остовного дерева взвешенного связного неориентированного графа. В текущей постановке задачи вершины графа ( $v_i$ ) представляют собой пиксели, а ребра  $e(v_i, v_j)$  – расстояние между ними. Нам известны расстояния между парами пикселей ( $v_i, v_j$ ) – это вес ребер  $w(e(v_i, v_j))$ . Для реализации алгоритма Краскала необходимо выполнить следующие шаги.

*Шаг 1.* Выполняем сортировку имеющихся в графе ребер в порядке возрастания длины.

*Шаг 2.* Анализируем ребра в порядке возрастания длины и смотрим на концы  $e = (a, b)$ .

2.1. Если вершины  $a$  и  $b$  принадлежат одному множеству (внутри их подграфа сумма ребер уже минимальна), то такое ребро пропускается, так как оно образует цикл в дереве.

2.2. Если вершины  $a$  и  $b$  принадлежат разным подмножествам, то их нужно объединять, так как мы нашли ребро минимальной длины, объединяющее оба подмножества в одно.

2.3. Продолжаем цикл объединения до получения одного единственного множества, равного искомому, в котором будут присутствовать все вершины графа.

*Шаг 3.* Получаем одно единственное множество вершин и список ребер, использованных для его объединения. Данное множество представляет собой дерево с минимальной суммарной длиной, которое объединяет все вершины.

Для реализации понятия «множество» в алгоритме Краскала используется структура данных, ориентированная на хранение компонент связности в графах, – Disjoint-Set Data Structure, которая поддерживает выполнение следующих операций:

– выяснение, к какому множеству в текущий момент принадлежит рассматриваемая вершина;

– объединение заданных множеств в одно.

Алгоритм Краскала не работает с изображением напрямую, поэтому возникает необходимость в создании графа. Существуют два подхода для построения графа по изображению: с использованием 4- и 8-пиксельных связей (рис. 1). В первом случае каждый пиксель соединяется с соседними пикселями сверху, снизу, слева и справа. Такое построение хорошо тем, что количество ребер в графе минимально. Во втором случае в дополнение к предыдущему подходу каждый пиксель соединяется с пикселями, расположенными на диагонали. При такой реализации количество ребер в графе увеличится, в связи с чем алгоритм будет выполняться медленнее, но получаемый результат сегментации будет более качественным, так как в нем учитывается больше связей между пикселями.

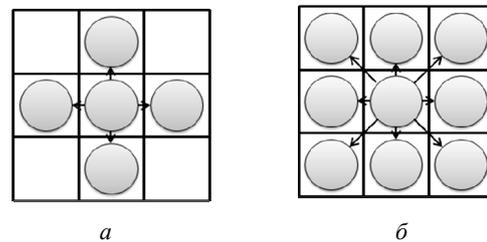


Рис. 1. Виды связей:  
а – 4-пиксельная; б – 8-пиксельная

В начале выполнения алгоритма Краскала каждый пиксель, представляющий собой вершину графа, будет находиться в собственном сегменте, но в дальнейшем пиксели, обладающие схожим цветом, т. е. пикселем одного объекта, постепенно будут объединяться в один сегмент. Например, если на каком-то шаге алгоритма попало ребро, соединяющее два соседних пикселя, в котором на одном конце ребра пиксель оранжевый, а на другом – красный, то длину ребра в таком случае можно определить как разницу цвета между пикселями.

Далее необходимо выяснить, в одном ли сегменте лежат анализируемые пиксели. Если они находятся в разных сегментах, то выполняется проверка. В случае прохождения проверки пиксели объединяются в один сегмент и построение графа продолжается.

Выделение объекта происходит с помощью следующих действий:

– поиск сегмента некоторого пикселя  $x$  по предкам до самого верха. Самый верхний пиксель – это корень дерева, представитель данного сегмента на текущий момент;

– объединение сегментов. Если у пикселей разные представители, значит они принадлежат различным сегментам, иначе корень был бы один. Для объединения пикселей представитель сегмента меньшей высоты отсылается на более длинного представителя, чтобы не увеличивать высоту дерева. В результате будет получен объединенный сегмент с общим представителем;

– для того чтобы в следующий раз не удаляться далеко от пикселя к корню, после того как представитель будет успешно обнаружен, на него устанавливается прямая ссылка из пикселя. Это сокращает путь последующих поисков. Для этого в качестве длины ребра выбирается евклидово расстояние, зависящее как от положения пикселей  $(x, y)$ , так и от их цвета  $(r, g, b)$ . Теперь пиксели будут считаться соседями, если они расположены рядом либо имеют схожий оттенок, хотя физически могут находиться на некотором удалении.

Сегментация на основе классификаторов (каскадов) лежит в основе алгоритма Виолы–Джонса, который заключается в том, чтобы при прохождении каскада весь кадр обрабатывать целиком, в результате чего можно определить области, в которых существует вероятность обнаружения объекта, и усилить простые классификаторы (рис. 2) [4].

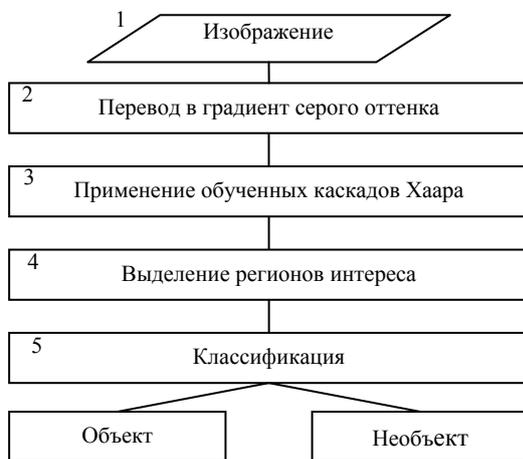


Рис. 2. Схема работы алгоритма Виолы–Джонса

Классификатор – это набор признаков. Однако не все признаки могут точно описывать объект, вследствие чего необходимо обучение и усиление слабых признаков. Усиление простых классификаторов – подход к решению задачи классификации (распознавания) путем комбинирования примитивных слабых классификаторов в один сильный (под силой классификатора подразумевается эффективность решения задачи классификации). Слабый классификатор имеет вид

$$h(x, f, p, \theta) = \begin{cases} 1, & \text{если } pf(x) < p\theta, \\ 0, & \text{если иначе,} \end{cases}$$

где  $f$  – признак;  $p$  – полярность, показывающая направление неравенства;  $\theta$  – пороговое значение.

Для формирования сильного классификатора как взвешенной комбинации слабых классификаторов применяется процедура AdaBoost. Пусть даны примеры изображений  $(x_1, y_1), \dots, (x_n, y_n)$ , где  $y_i = [0, 1]$  для отрицательных и положительных примеров соответственно. Следует определить значение веса  $w_{1,i} = 1/2m, 1/2l$  для  $y_i = [0, 1]$  соответственно, где  $m$  используется для отрицательных примеров;  $l$  – для положительных.

Нормализация происходит при условии, что существует конечное число слабых классификаторов:

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

Из всех слабых классификаторов выбирается самый сильный – с минимальной ошибкой:

$$\varepsilon_t = \min_{f,p,\theta} \sum_i \varepsilon_t |h(x_i, f_t, p_t, \theta_t) - y_i|$$

Затем определяется слабый классификатор, переменные которого минимизируют  $\varepsilon_t$ :

$$h_t(x) = h(x_i, f_t, p_t, \theta_t),$$

где  $f_t$  – признак;  $p_t$  – полярность;  $\theta_t$  – пороговое значение.

После того как один слабый классификатор найден, продолжается поиск других и обновляются веса:

$$w_{t+1,i} = w_{t,i} \cdot \beta_t^{1-e_i},$$

где  $e_i = 0$ , если пример  $x_i$  классифицирован правильно, и  $e_i = 1$ , если иначе;

$$\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$$

Значение сильного классификатора, состоящего из слабых, находится по следующей формуле:

$$C(x) = \begin{cases} 1, & \text{если } \sum_{t=1}^T \alpha_t h_t \geq \frac{1}{2} \sum_{t=1}^T \alpha_t, \quad \alpha_t = \log \frac{1}{\beta_t}, \\ 0, & \text{если иначе.} \end{cases}$$

Обучающий алгоритм для создания сильного классификатора позволяет вводить максимально и минимально допустимый уровень срабатываний. Для существующего набора отрицательных и положительных примеров задается общий уровень ложных срабатываний, для чего используется процедура усиления классификаторов AdaBoost. Полученный каскад необходимо оценить на тестовом наборе.

Признак, используемый для анализа изображения, можно представить кортежем

$$\text{Feature (особенность)} = \{T, O, S\},$$

где  $T$  – тип признака;  $O$  – координата левого верхнего угла признака;  $S$  – размер признака по горизонтали и вертикали в пикселях.

Значение признака рассчитывается по формуле

$$\text{Feature} = k_w \sum V_w + k_b \sum V_b,$$

где  $\sum V_w, \sum V_b$  – суммы интенсивностей всех пикселей изображения в белых и черных областях признака;  $k_w$  и  $k_b$  – коэффициенты их нормирования по площади.

Для быстрого вычисления признаков используется интегральное изображение (рис. 3). Оно представляет собой матрицу, размерность которой совпадает с раз-

мерностью исходного изображения. Элементы матрицы определяются по следующей формуле:

$$\text{SAT}(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y'),$$

где  $\text{SAT}(x, y)$  – значение элемента интегрального изображения с координатами  $x, y$ ;  $I(x', y')$  – яркость пикселя исходного изображения.

При проведении экспериментов было использовано более 100 изображений кистей рук, растений и лиц людей размером от  $640 \times 480$  до  $1\,600 \times 1\,200$  (рис. 4).

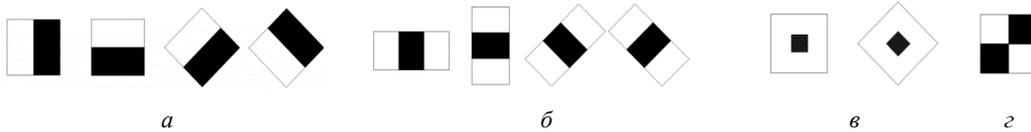


Рис. 3. Виды признаков для обученного каскада Хаара:  
*a* – граничные; *б* – линейные; *в* – центральные; *г* – диагональные

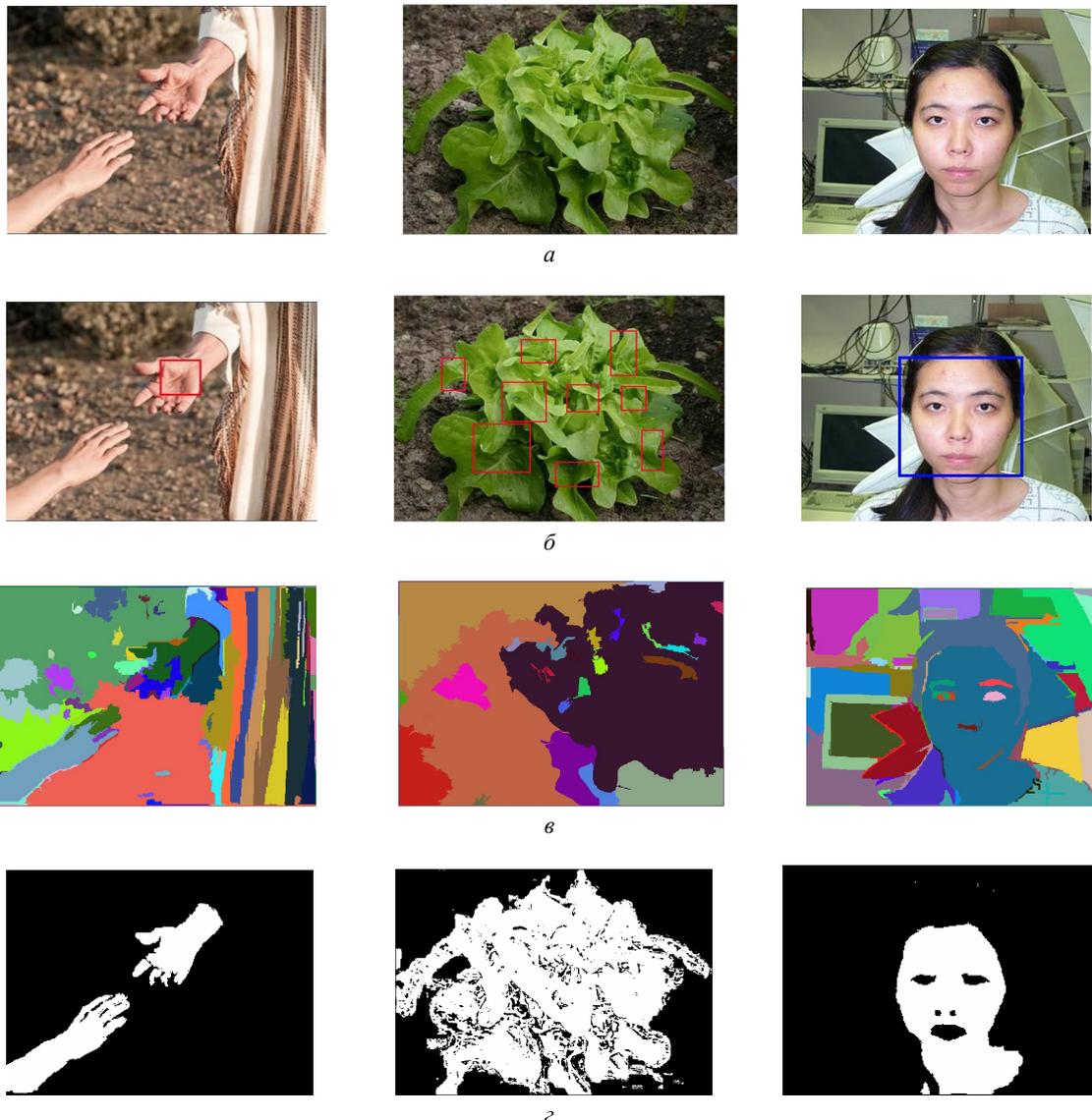


Рис. 4. Примеры сегментации изображений:  
*a* – оригинальные изображения; *б* – изображения, полученные с помощью алгоритма Виолы–Джонса;  
*в* – алгоритма Краскала; *г* – пороговой сегментации

В алгоритме Краскала при определении расстояния между пикселями в таком аспекте, как перепад интенсивностей, пиксели пестрого, пусть даже единого объекта, будут реагировать на объединение в один объект. Во избежание этого предлагается использовать фильтр Гаусса, который позволяет убрать шум и размыть изображение. Это вызывает взаимопроникновение цветных пикселей, и они более охотно идут на контакт.

Зависимость времени работы алгоритма от размеров изображения можно увидеть на рис. 5. Наибольшее быстродействие достигается при использовании модели YCrCb.

При оценке точности локализации были сформированы три набора экспериментальных данных по каждой категории:

– категория 1 – фон содержит минимальное количество объектов, затрудняющих локализацию, и имеет относительно простую структуру;

– категория 2 – фон содержит некоторое количество объектов и характеризуется умеренной неоднородностью;

– категория 3 – фон содержит некоторое количество объектов и обладает сложной структурой.

При рассмотрении результатов сегментации точность обнаружения объекта в значительной степени зависит от внешней среды и установленных параметров работы алгоритма (см. таблицу).

В ходе исследования было выяснено, что для локализации лиц людей наилучшим образом подходит сегментация с использованием цветных моделей YCrCb, HLS и их комбинации. Алгоритм Виолы–Джонса достаточно устойчив к поворотам изображения до 15°. При локализации лиц в таких условиях алгоритм обладает крайне низкой вероятностью ложного обнаружения лица и является одним из лучших по показателю точности локализации лица, которая составляет 98,62 %. Алгоритм Виолы–Джонса обеспечивает хорошие результаты при поворотах изображения относительно камеры.

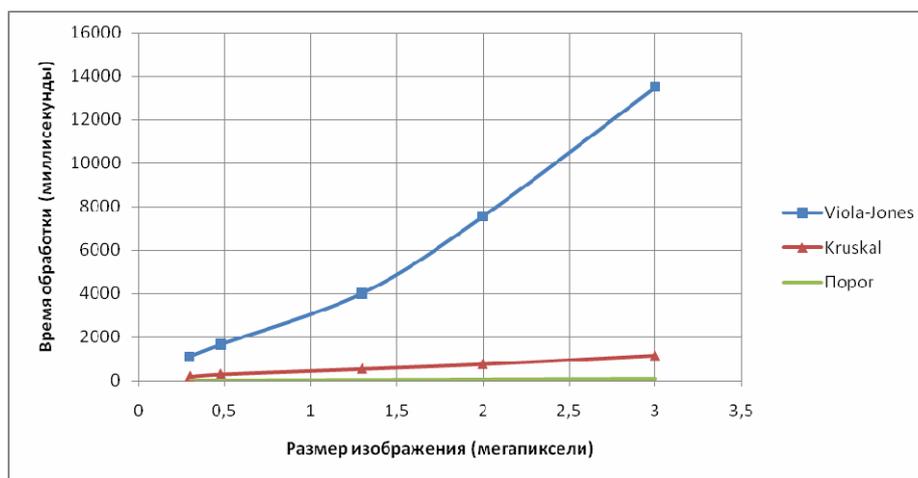


Рис. 5. Зависимость времени работы алгоритмов от размера изображения

### Оценка точности сегментации

Категория	Пороговая сегментация		Алгоритм Краскала		Алгоритм Виолы–Джонса	
	Среднее количество сегментов	Процент покрытия	Среднее количество сегментов	Процент покрытия	Среднее количество сегментов	Процент покрытия
<i>Изображения растений</i>						
Категория 1	15	87,8...99,4	10	95,2...100,0	9	30,1...44,4
Категория 2	21	83,3...97,2	19	87,0...96,9	10	36,3...49,8
Категория 3	24	81,4...96,5	31	73,3...95,1	7	21,5...38,8
<i>Изображения лиц людей</i>						
Категория 1	1	95,1...100,0	1	97,2...100,0	1	97,9...100,0
Категория 2	3	93,3...100,0	4	89,5...97,5	1	97,1...100,0
Категория 3	4	89,2...96,8	7	83,1...91,3	2	96,7...100,0
<i>Изображения кистей рук людей</i>						
Категория 1	2	95,1...99,7	3	94,7...99,8	1	50,0...64,4
Категория 2	4	93,3...98,4	5	86,4...97,7	1	38,5...51,6
Категория 3	6	88,7...97,1	8	78,8...93,1	1	21,9...41,7

*Примечание.* При формировании табличных значений в ходе оценки экспериментальных данных не учитывалось покрытие сегментами областей, которые не относятся к объекту интереса.

При локализации кистей рук людей наилучшие результаты показали методы пороговой сегментации и алгоритм Краскала: 91,75 и 95,38 % соответственно.

Средняя точность при локализации растений пороговой сегментацией составляет 90,9 %, алгоритмом Краскала – 91,2 %.

В ходе экспериментов было выяснено, что алгоритм Виолы–Джонса не подходит для сегментации растений, поскольку в процессе развития они неоднократно меняют свою геометрию, что затрудняет обучение каскадов Хаара. При этом точность локализации не превышает 37 %.

Все вычисления производились на компьютере Intel Core 2 Duo P8400 2,2 MHz, с 4 Гбайтами оперативной памяти. Следует отметить что реализации алгоритмов Виолы–Джонса и Краскала очень требовательны к ресурсам компьютера. Время работы алгоритма Виолы–Джонса превышает порог в 1 с при об-

работке небольших изображений, например при обработке изображения с размером 1 024×768 – более 3 с, при этом алгоритм Краскала выполняется за 0,4 с.

#### Библиографические ссылки

1. Бузаев Д. В., Зотин А. Г., Носов А. В. Сравнение методов для системы локализации и обнаружения лиц // Молодежь Сибири – науке России : материалы науч.-практ. конф. Красноярск, 2011.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений : пер. с англ. М. : Техносфера, 2006.
3. Felzenszwalb P. F., Huttenlocher D. P. Efficient Graph-Based Image Segmentation // Intern. J. of Computer Vision. 2004. Vol. 59, № 2. P. 37–44.
4. Viola P., Jones M. Robust Real-Time Object Detection // Intern. J. of Computer Vision. Kluwer Academic Publishers, 2004. P. 137–154.

A. G. Zotin, A. V. Nosov, D. V. Buzaev

#### ANALYSIS OF SUITABILITY OF SEGMENTATION METHODS BASED ON COLOR AND STRUCTURAL FEATURES FOR LOCALIZATION OF OBJECTS

*The existing methods of segmentation, according to color and structural features, are considered. The algorithms of the most widely known methods of segmentation based on thresholding, graphs, and cascades are described. Results of the analysis of segmentation methods suitability for objects localization are shown for the cases of plants, people's faces and areas of the hands.*

*Keywords: image analysis, segmentation, threshold processing, the Kruskal's algorithm, Haar's cascades.*

© Зотин А. Г., Носов А. В., Бузаев Д. В., 2012

УДК 517.972.5

И. А. Лопатин

#### РЕШЕНИЕ ВАРИАЦИОННОЙ ЗАДАЧИ ПО ВЫБОРУ ОПТИМАЛЬНОЙ ФОРМЫ СЖАТОГО СТЕРЖНЯ

*Представлено решение задачи о выборе закона изменения поперечного сечения стержня, обеспечивающего восприятие максимальной сжимающей силы при заданном объеме материала.*

*Ключевые слова: стержень, устойчивость, оптимизация.*

Осесимметричные пространственные фермы, составленные из прямолинейных стержней, нашли широкое применение в качестве адаптеров полезной нагрузки. Расчет таких конструкций проводится в предположении, что стержни фермы соединены шарнирно, а основным видом разрушения стержня является потеря устойчивости при действии на него сжимающей силы.

Стержни фермы в большинстве конструкций имеют постоянное по длине поперечное сечение. Однако использование в ферме стержней с переменным поперечным сечением позволяет создавать более эффективные конструкции [1; 2].

Рассмотрим шарнирно опертый стержень длиной  $l$ , нагруженный сжимающей силой  $P$ . Пусть стержень имеет круглое поперечное сечение, радиус которого  $r$

зависит от продольной координаты  $x$ . Длина стержня  $l$  и его объем  $V_0$  являются заданными величинами. Необходимо подобрать закон изменения радиуса поперечного сечения, который обеспечивает максимум критической силы  $P$  при известном объеме стержня  $V_0$ .

Сформулируем условие этой задачи как задачи вариационного исчисления. Объем стержня  $V_0$  определяется функционалом

$$V_0 = \int_0^l \pi [r(x)]^2 dx, \quad (1)$$

где  $r(x)$  – зависимость радиуса стержня от продольной координаты.

Уравнение устойчивости стержня имеет вид

$$w_{xx} + \frac{Pw}{EJ(x)} = 0, \quad (2)$$