

**ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ**  
**COMPUTER SCIENCE, COMPUTER ENGINEERING AND MANAGEMENT**  
**УДК 004.93(004.4, 004.6, 303.732)**  
**DOI:10.21822/2073-6185-2021-48-3-59-67**

Оригинальная статья / Original Paper

**Алгоритм обнаружения пожара для мультисенсорной системы**

**Р.А. Багутдинов, М.Ф. Степанов**

Институт электронной техники и приборостроения,  
Саратовский государственный технический университет имени Ю. А. Гагарина  
410054, г. Саратов, ул. Политехническая, 77, Россия

**Резюме. Цель.** В работе предлагается алгоритм обнаружения пожара для мультисенсорной системы. Из-за сложных условий в полевых условиях в первые часы спасательные операции затруднены и часто подвергают опасности жизнь спасателей. Основная научная цель заключается в том, чтобы система работала автономно на определенных сегментах процесса мониторинга, при этом три агента в разной степени должны взаимодействовать друг с другом на основе алгоритмов коммуникации и принятия решений. **Метод.** Существует множество алгоритмов обработки изображения, но не достаточно проработаны и описаны алгоритмы, использующие несколько источников информации. Основное внимание уделяется алгоритмам обнаружения пожара. Алгоритм разработан с использованием NI Vision Assistant, программного инструмента для быстрого создания прототипов и тестирования приложений обработки изображений. **Результат.** Помимо программной реализации на языке C, который NI Vision Assistant генерирует по умолчанию, в работе представлена Python-реализация алгоритма. **Вывод.** Результаты работы могут быть использованы для разработки мультисенсорных систем в задачах мониторинга труднодоступных районов.

**Ключевые слова:** обработка изображений, обнаружение пожара, сегментация изображений, мультиагентная мультисенсорная система

**Для цитирования:** Р.А. Багутдинов, М.Ф. Степанов. Алгоритм обнаружения пожара для мультисенсорной системы. Вестник Дагестанского государственного технического университета. Технические науки. 2021; 48(3): 59-67. DOI:10.21822/2073-6185-2021-48-3-59-67

**Algorithm of fire detection for multi-sensor system**

**R.A. Bagutdinov, M.F. Stepanov**

Institute of Electronic Engineering and Instrumentation,  
Yu. A. Gagarin Saratov State Technical University,  
77 Polytechnic Str., 410054 Saratov, Russia

**Abstract. Objective.** The paper proposes a fire detection algorithm for a multisensor system. Due to the difficult conditions in the field, for the first time, watch rescue operations are difficult and often endanger the lives of rescuers. The main scientific goal is for the system to work autonomously on certain segments of the monitoring process, while the three agents, to varying degrees, must interact with each other based on communication and decision-making algorithms. **Method.** There are many algorithms for image processing, but algorithms that use several sources of information are not sufficiently developed and described. The focus is on fire detection algorithms. The algorithm was developed using NI Vision Assistant, a software tool for rapid prototyping and testing of imaging applications. **Result.** In addition to the software implementation in C, which NI Vision Assistant generates by default, the paper presents a Python implementation of the algorithm. **Conclusion.** The results of the work can be used to develop multisensor systems for monitoring hard-to-reach areas.

**Keywords:** image processing, fire detection, image segmentation, multi-agent multisensor system

**For citation:** R.A. Bagutdinov, M.F. Stepanov. Algorithm of fire detection for multi-sensor system. Herald of Daghestan State Technical University. Technical Sciences. 2021; 48 (3): 59-67. (In Russ.) DOI:10.21822/2073-6185-2021-48-3-59-67

**Введение.** Последствия стихийных бедствий, таких как землетрясения, наводнения или пожары, особенно серьезны и могут привести к человеческим жертвам и значительному материальному ущербу. Бедствия затрагивают как густонаселенные городские, так и пригородные районы с различным рельефом. Из-за сложных условий в полевых условиях в первые часы спасательные операции затруднены и часто подвергают опасности жизнь спасателей.

**Постановка задачи.** В настоящее время существуют различные наземные или воздушные роботизированные системы с дистанционным управлением, которые облегчают инспекцию и позволяют проводить мониторинг в труднодоступных районах. Из-за разнообразных условий указанные системы могут быть эффективными в одних случаях, но неуместными в других. По этой причине предполагается использовать многоагентную мультисенсорную систему (ММС), состоящую из трех различных транспортных средств, в дальнейшем именуемых агентами мониторинга. Благодаря своим различным характеристикам три агента могут дополнять друг друга, значительно расширяя возможности системы.

Основная информация, собираемая тремя агентами, основана на цифровых изображениях, передаваемых камерами, установленными на каждом агенте. Основное внимание здесь уделяется алгоритмам обнаружения пожара, поскольку пожар считается важным индикатором нежелательного события.

ММС состоит из центра управления и следующих агентов: наземный движущийся робот, летательный аппарат (квадрокоптер с камерой) и экраноплан (автономный маловысотный летательный аппарат). Все эти агенты оснащены камерами, поэтому основные данные, получаемые с ММС, представляют собой цифровые изображения. Изображения могут обрабатываться либо локально, внутри каждого агента, либо передаваться в центр управления.

Цифровые изображения содержатся в специальных файлах графического формата. Цветовая модель описывает, как цвета могут быть представлены в виде кортежа чисел и могут иметь связанное одно или несколько цветовых пространств. Некоторые типичные цветовые пространства - стандартные RGB, HSV, HSL. Эти цветовые пространства содержат три отличительных канала. Каждый канал может быть сопоставим с серым изображением.

В данной работе предлагается алгоритм обработки изображений на основе гистограмм для обнаружения пожара. Идея состоит в том, чтобы получить последовательные двоичные маски, отфильтровать исходное изображение и получить области изображения, которые могут содержать детектирование источников огня. Конвейер обработки извлекает наиболее подходящие цветовые плоскости (каналы) из цветового пространства исходного изображения и преобразует их в двоичные маски путем определения оптимального порогового значения. В процессе выполнения работы был проведен аналитический анализ источников аналогичных зарубежных исследований, некоторые из которых представлены ниже.

**Методы исследования.** В статье [1] представлен метод обнаружения пожара в реальном времени на основе изображений с использованием обработки изображений на основе цветовой модели RGB и цветовых каналов. В некоторых работах также представлены методы обработки изображений [2, 3] и разнородных данных, в том числе с мультисенсорных систем [4, 5]. В статье [6] представлены алгоритмы обнаружения

пожара с использованием цветовых моделей RGB. В статьях [7, 8] представлены алгоритмы обнаружения пожара с использованием цветовой модели RGB. В статье [9] представлен метод обнаружения пожара на изображениях с использованием процедур кластеризации K-metod и оптимизации частиц для лучшего обнаружения ложных тревог. В статье [10] представлен метод эффективного обнаружения пламени в нескольких сценах изображения. Используется набор параметрических представлений, названных Gradient Features (GF), чтобы изучить особенности изменения цвета пламени на изображении.

В статье [11] предлагается новый метод обнаружения огня в реальном времени путем обработки видеоданных, генерируемых обычной камерой наблюдения. Он использует 1-мерное временное вейвлет-преобразование и изменение цвета в движущихся областях огненного цвета с использованием 2-мерного пространственного вейвлета.

В статьях [12, 13] представлена модель для идентификации огня и дыма с использованием обработки изображений на основе нечеткой логики. В статьях [14] представлен метод обнаружения пламени и дыма с видеосъемкой в реальном времени. В статье [15] представлены несколько пороговых методов, которые могут применяться к цифровым изображениям для получения двоичного изображения. Но, не в одной из выше указанных работ не рассматривалась ситуация использования нескольких сенсоров для обнаружения огня.

Модуль NI Vision Development содержит сотни алгоритмов обработки изображений, включая фильтры, морфологию, сопоставление с образцом, классификация и многое другое. Вероятность появления серого уровня  $i$  равна

$$p(i) = \frac{h(i)}{\sum_{i=0}^{N-1} h(i)} \quad (1)$$

где:  $h(i)$  обозначает количество пикселей в изображении при каждом значении уровня серого, а  $N = 256$ .

Метод Inter Variance использует вероятность уровней серого цвета для определения оптимального порогового значения  $k$ , для которого максимизируется следующее выражение:

$$\sigma_{\delta}^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)(1 - \omega(k))} \quad (2)$$

где:

$$\mu(k) = \sum_{i=0}^k ip(i) \quad (3)$$

$$\mu_T = \sum_{i=0}^{N-1} ip(i) \quad (4)$$

$$\omega(k) = \sum_{i=0}^k p(i) \quad (5)$$

Наблюдение: метод Inter Variance - это реализация известного алгоритма для глобального автоматического порога гистограммы.

Метод кластеризации для автоматического глобального порога серого изображения включает получение значения  $k$ .

$$\frac{\mu_1 + \mu_2}{2} = k \quad (6)$$

где  $\mu_1$  - это среднее значение всех пикселей от 0 до  $k$ , а  $\mu_2$  - среднее значение всех пикселей между  $k + 1$  и 255.

Представленный алгоритм использует цветовую модель RGB анализируемого изображения. Использовался программный инструмент NI Vision, который можно использовать для создания прототипов и тестирования приложений обработки изображений. Алгоритм записывается в файл сценария, который содержит функции обработки и соответствующие параметры. Обработка конвейера представлена на рис. 1.

Этапы обработки изображения следующие:

1. Загрузить исходное изображение из файла;
2. Сохранить изображение в буфере памяти для дальнейшего использования;
3. Извлечение синего цвета, получение серого изображения;

4. Применение глобального автоматического порога для получения двоичного изображения;
5. Инвертирование двоичного изображения, полученное на шаге # 4, чтобы получить маску № 1;
6. Сохранение маски №1 в файле изображения;
7. Получение исходного изображения из буфера памяти;
8. Создание маски исходного изображения с помощью маски №1, получив новое цветное изображение с именем «image1»;
9. Извлечение красного цвета из «image1»;
10. Анализ логарифмической гистограммы;
11. Применение приблизительного глобального порога, чтобы получить маску №2;
12. Удаление маски № 2 изображение;
14. Сохранение маски №2 в файл изображения;
15. Маскировка «image1» маской №2, чтобы получить области, где существует пожар.

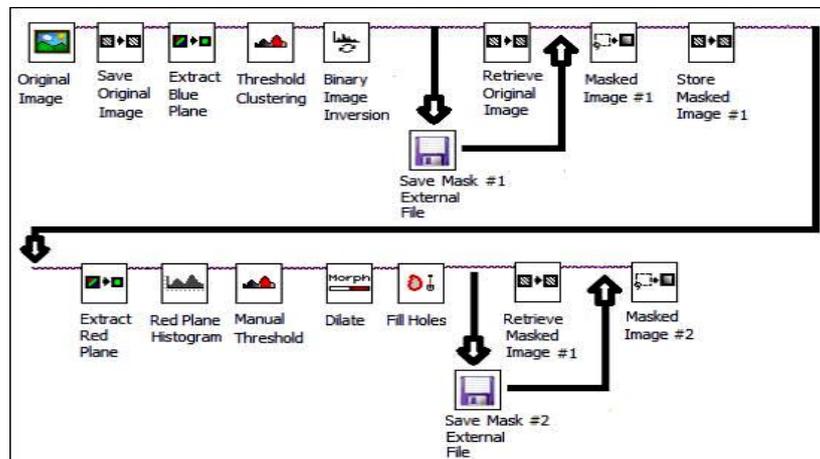


Рис. 1. Скрипт NI Vision с использованием цветовой модели RGB  
Fig. 1. NI Vision Script Using RGB Color Model

Описанный выше сценарий NI Vision может быть автоматически преобразован в программу с кодом С. Однако необходимо было достичь следующих целей: устранить произвольный выбор пользователем порогового значения на шаге №11 и закодировать алгоритм на Python. Далее применялся одномерный фильтр Гаусса к логарифмической гистограмме красного цвета, чтобы «сгладить» гистограмму, после чего последовало численное вычисление первой производной и второй производной. Выбиралась пороговая точка как локальный минимум, ближайший к 255, где первая производная равна нулю, а вторая производная положительна.

В процессе автоматического определения порога на шаге №4 использовалась группировка в файл otsu.py. Код Python представлен ниже:

```
from scipy.ndimage import gaussian_filter1d from matplotlib import pyplot as plt
from otsu import otsu, running_variance, stretch,
unstretch
import math
import numpy as np
import cv2
minval=0
maxval=255
# STEP 1
image_orig= cv2.imread('forest5_test.bmp')
# STEP 2
```

```
image= image_orig.copy()
# STEP 3
blue_plane= image[:, :, 0]
# STEP 4
image_grayblue=stretch(blue_plane)
print()
threshold_otsu=int(otsu(image_grayblue)*maxval)
print(otsu(image_grayblue))
print(threshold_otsu)
print()
_,threshOtsu=cv2.threshold(blue_plane, threshold_otsu, 255, cv2.THRESH_BINARY)
contours1, hierarchy1 = cv2.findContours( threshOtsu, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
for cnt1 in contours1:
cv2.drawContours(threshOtsu,[cnt1],0,255,-1)
# STEP 5
mask = cv2.bitwise_not(threshOtsu)
STEP6 cv2.imwrite('Mask1.bmp', mask)
STEP 7 mask1=cv2.imread('Mask1.bmp', cv2.IMREAD_UNCHANGED )
STEP 8
image1=cv2.bitwise_and(image,image,mask=mask1)
#STEP 9
red_plane = image1[:, :, 2]
# STEP 10
histo_red = cv2.calcHist([red_plane], [0], None, [256], [0,256])
log_histo_red
=np.log(1+histo_red.astype(np.float)) #adjust extreme values of the histogram log_histo_red[0]
=log_histo_red[1] log_histo_red[255]=log_histo_red[254] print(log_histo_red)
mMax = max(log_histo_red)
arr_log_histo_red = np.array(log_histo_red)
# smooth
smooth = gaussian_filter1d(arr_log_histo_red.T[0], 3) #compute first derivate
smooth_d1 = np.gradient(smooth)
zeroD1= np.where( np.diff(np.sign( smooth_d1)))[0]
# compute second derivative
smooth_d2 = np.gradient(np.gradient(smooth))
zeroD2=np.where( np.diff( np.sign( smooth_d2)))[0]
histo_red[0]=histo_red[1]
histo_red[255]=histo_red[254]
plt.plot(histo_red, label='Original Histogram')
plt.legend(bbox_to_anchor=(0.5, 0.5))
plt.show()
plt.plot(arr_log_histo_red,
label='Logarithmic histogram')
plt.plot(smooth,
label='Smoothed Histogram')
plt.plot(smooth_d1/ np.max(smooth_d1),
label= 'First Derivate')
plt.plot(smooth_d2 / np.max(smooth_d2),
label= 'Second Derivate')
plt.axhline(y=0, color='k')
```

```
# STEP 11
last_local_minima=0
for i, xzero in enumerate(zeroD1, 1): if smooth_d2[xzero]>0: plt.axvline(x=xzero, color='k')
last_local_minima=xzero
plt.legend(bbox_to_anchor=(0.5, 0.5))
plt.show()
threshold_redplane=last_local_minima
,mask=cv2.threshold(red_plane, threshold_redplane, 255 , cv2.THRESH_BINARY)
STEP 12 dilation_size=1 element=cv2.getStructuringElement(
cv2.MORPH_RECT,(2*dilation_size+1,
2*dilation_size+1), (dilation_size, dilation_size)) dilation_mask2 = cv2.dilate(mask, element)
# STEP 13
contours2, hierarchy2 = cv2.findContours(mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE) for cnt2 in contours2:
cv2.drawContours(mask,[cnt2],0,255,-1)
STEP 15 cv2.imwrite('Mask2.bmp', mask)
STEP 15 mask2=cv2.imread('Mask2.bmp', cv2.IMREAD_UNCHANGED)
image2= cv2.bitwise_and(image1, mask=mask2)
#Display images
cv2.imshow("Initial",image)
cv2.imshow("Otsu",threshOtsu)
cv2.imshow("Mask1",mask1)
cv2.imshow("Image 1",image1)
cv2.imshow("Mask2",mask2)
cv2.imshow("Image 2",image2)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

**Обсуждение результатов.** Изображение, используемое в качестве входных данных для алгоритма, содержит большую область дыма (вверху слева) и область неба (вверху справа). Эти области могут быть обнаружены как ложноположительные участки возгорания. Они удаляются с помощью изображения бинарной маски, полученного путем применения порога к синей плоскости изображения. Репрезентативные изображения показаны ниже (рис. 2).

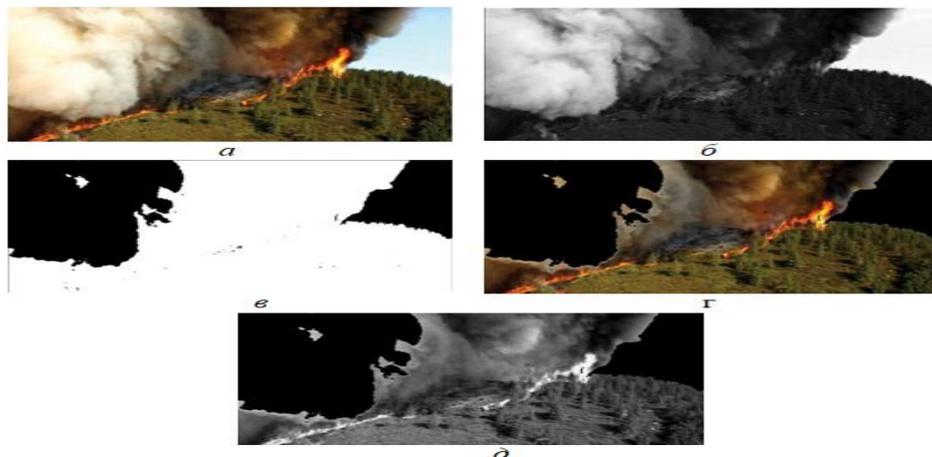


Рис. 2. Этапы обработки изображения (а – исходное изображение для алгоритма, б-в-наложение маски, г – исходное изображение с наложенной маской, д – выделение красного цвета изображения)

Fig. 2. Stages of image processing (a - the original image for the algorithm, b-c - masking, d - the original image with the applied mask, e - highlighting the red color of the image)

На рис. 3 представлено логарифмическое представление гистограммы красной плоскости, сгенерированной скриптом NI Vision Assistant. Произвольное пороговое значение отображается стрелкой.

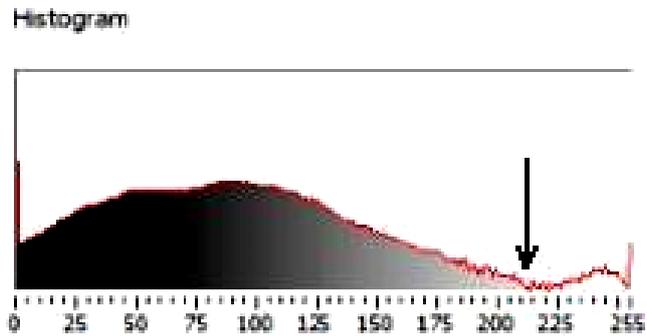


Рис. 3. Ручное пороговое значение NI Vision Assistant  
Fig. 3. NI Vision Assistant Manual Threshold

На рис. 4 представлено автоматическое определение порога на основе первой и второй производных сглаженной логарифмической гистограммы.

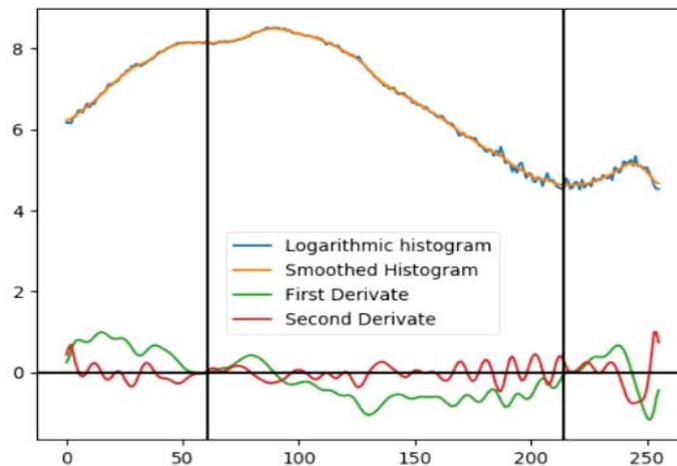


Рис.4. Порог, полученный с использованием кода Python  
Fig. 4. Threshold obtained using Python code

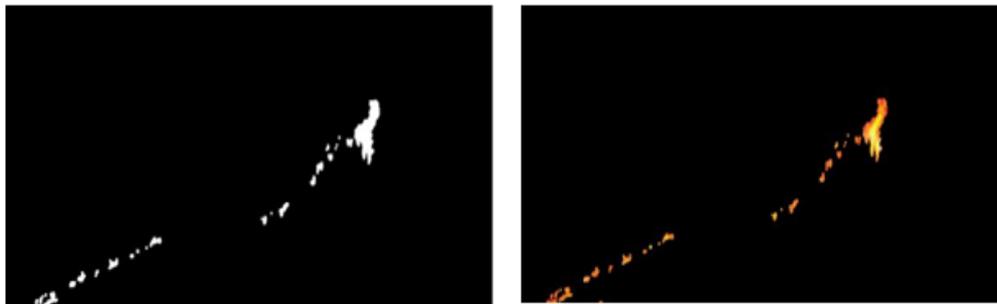


Рис. 5. Результаты наложения маски, примененной к исходному изображению  
Fig. 5. The results of overlaying a mask applied to the original image

На рис.5 показано обнаружение пожара, выполняемое последовательными бинарными масками, примененными к исходному изображению.

**Вывод.** Обнаружение пожара на цифровых изображениях - сложная задача. Наличие на изображении ясного неба, дыма и облаков может дать ложноположительные результаты.

Алгоритм обработки изображений, описанный в статье, использует анализ гистограмм соответствующих цветовых плоскостей, извлеченных из исходных изображений. Цветовые плоскости преобразуются в бинарные маски с помощью порогового метода. Различные сценарии обработки изображений разрабатываются с

помощью NI Vision Assistant, мощного программного инструмента. В работе также представлена реализация Python обнаружения пожара в изображении, которая подходит для различных устройств мониторинга.

#### **Библиографический список:**

1. Turgay Celic, Fast and Efficient Method for Fire Detection Using Image Preccessing, ETRI Journal, Volume 32, Number 6, December 2010, pp. 881-890
2. Багутдинов Р.А. Разработка мультисенсорной системы для задач мониторинга и интерпретации разнородных данных / Системный администратор. 2019. № 3 (196). С. 82-85.
3. Багутдинов Р.А. Проектирование модульной мультисенсорной системы для задач мониторинга окружающей среды на базе Arduino / Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика. 2019. Т. 46. № 1. С. 173-180.
4. Островский О.А. Вопросы развития компьютерно-технической экспертизы и их взаимосвязь с телекоммуникациями / Право и государство: теория и практика. 2020. № 1 (181). С. 312-314.
5. Островский О.А. Извлечение информационных следов с мобильных устройств при расследовании преступлений в сфере компьютерной информации / Вестник Таджикского национального университета. Серия социально-экономических и общественных наук. 2018. № 7. С. 237-241.
6. Supriya Sameer Nalawade, Fire Detection System using RGB Color Model, International Journal of Engineering Science and Computing, May 2018, Volume 8, Issue #5, 1751. P.17518
7. Jing Shao, Guanxiang Wang, Wei Guo, An Image-Based Fire Detection Method Using Color Analysis, 2012 International Conference on Computer Science and Information Processing, P. 419-422
8. Ping-He Huang, Jing-Yong Su, Zhe-Ming Lu, Jeng-Shyang Pan, A Fire-Alarmng Method Based on Video Processing, Proceedings of the 2006 International Conference on Intelligent, Information Hiding and Multimedia Signal Processing (IHH-MSP'06), P. 359-364
9. Oluwarotimi Giwa, Abdsamad Benkrid, Fire detection in a still image using color information, <https://www.researchgate.net/publication/323723024> (дата обращения 04.01.2021).
10. Liping Zhu, Hongqi Li, Fenghui Wang, Jie Lv, Ali Sikandar, Hong Zhang, A Flame Detection Method Based on Novel Gradient Features, Intell. Syst. 2020; 29(1): P. 773–786, <https://doi.org/10.1515/jisys-2017-0562>
11. Uğur B., Töreyn, Yiğithan Dedeoğlu, Uğur Güdükbay, Enis Cetin A., Computer vision based method for real-time fire and flame detection, Pattern Recognition Letters 27 (2006). P. 49–58
12. Sushil Garg, Balaji R. Sharma1, Kelly Cohen, Manish Kumar, A Fuzzy Logic Based Image Processing Method for Automated Fire and Smoke Detection, [https://www.researchgate.net/publication/2583\\_40149](https://www.researchgate.net/publication/2583_40149) (дата обращения 04.01.2021).
13. Turgay Çelik, Hüseyin Özkaramanlı, Hasan Demirel, Fire and smoke detection without sensors: image processing based approach, 15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, September 3-7, 2007, P. 1794-1798
14. Chunyu Yu, Zhibin Mei, Xi Zhang, A Real-time Video Fire Flame and Smoke Detection Algorithm, Procedia Engineering, Volume 62, 2013, P. 891-898
15. Mehmet Sezgin, Bülent Sankur, Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging 13(1), 2004, P. 146–165

#### **References:**

1. Turgay Celic, Fast and Efficient Method for Fire Detection Using Image Preccessing, ETRI Journal, Volume 32, Number 6, December 2010; 881-890.
2. Bagutdinov R.A. Development of a multisensor system for monitoring and interpretation of heterogeneous data [Sistemnyy administrator] *System administrator*. 2019; 3 (196): 82-85. (In Russ)
3. Bagutdinov R.A. Designing a modular multisensor system for environmental monitoring tasks based on Arduino. [Nauchnyye vedomosti Belgorodskogo gosudarstvennogo universiteta. Seriya: Ekonomika. Informatika] *Scientific Bulletin of Belgorod State University. Series: Economics. Informatics*. 2019; 46(1): 173-180. (In Russ)
4. Ostrovsky O.A. Issues of development of computer-technical expertise and their relationship with telecommunications. [Pravo i gosudarstvo: teoriya i praktika] *Law and State: theory and practice*. 2020; 1 (181): 312-314. (In Russ)
5. Ostrovsky O.A. Extracting information traces from mobile devices in the investigation of crimes in the field of computer information. [Vestnik Tadzhijskogo natsional'nogo universiteta. Seriya sotsial'no-ekonomicheskikh i obshchestvennykh nauk] *Bulletin of the Tajik National University. A series of socio-economic and social sciences*. 2018; 7: 237-241. (In Russ)
6. Supriya Sameer Nalawade, Fire Detection System using RGB Color Model, International Journal of Engineering Science and Computing, May 2018; 8(5):1751-17518.

7. Jing Shao, Guanxiang Wang, Wei Guo, An Image-Based Fire Detection Method Using Color Analysis, 2012 International Conference on Computer Science and Information Processing, 2012; 419-422.
8. Ping-He Huang, Jing-Yong Su, Zhe-Ming Lu, Jeng-Shyang Pan, A Fire-Alarming Method Based on Video Processing, Proceedings of the 2006 International Conference on Intelligent, Information Hiding and Multimedia Signal Processing (IHM-MSP'06), 2006; 359-364.
9. Oluwarotimi Giwa, Abdsamad Benkrid, Fire detection in a still image using color information, <https://www.researchgate.net/publication/323723024> (date of the application 04.01.2021)
10. Liping Zhu, Hongqi Li, Fenghui Wang, Jie Lv, Ali Sikandar, Hong Zhang, A Flame Detection Method Based on Novel Gradient Features, *Intell. Syst.* 2020; 29(1): 773–786, <https://doi.org/10.1515/jisys-2017-0562>
11. Uğur B., Töreyn, Yiğithan Dedeoğlu, Uğur Gündükbay, Enis Cetin A., Computer vision based method for real-time fire and flame detection, *Pattern Recognition Letters* 27. 2006; 49–58.
12. Sushil Garg, Balaji R. Sharma, Kelly Cohen, Manish Kumar, A Fuzzy Logic Based Image Processing Method for Automated Fire and Smoke Detection, <https://www.researchgate.net/publication/258340149> (date of the application 04.01.2021).
13. Turgay Çelik, Hüseyin Özkaramanlı, Hasan Demirel, Fire and smoke detection without sensors: image processing based approach, 15th European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, September 3-7, 2007; 1794-1798.
14. Chunyu Yu, Zhibin Mei, Xi Zhang, A Real-time Video Fire Flame and Smoke Detection Algorithm, *Procedia Engineering*, 2013; 62: 891-898.
15. Mehmet Sezgin, Bülent Sankur, Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*. 2004; 13(1): 146–165

**Сведения об авторах:**

**Багутдинов Равиль Анатольевич**, аспирант, кафедра «Системотехника и управление в технических системах», e-mail: [rav379@mail.ru](mailto:rav379@mail.ru)

**Степанов Михаил Федорович**, доктор технических наук, доцент, профессор кафедры «Системотехника и управление в технических системах», e-mail: [rav379@mail.ru](mailto:rav379@mail.ru)

**Information about the authors:**

**Ravil A. Bagutdinov**, Postgraduate student, Department of "Systems Engineering and Control in Technical Systems", e-mail: [rav379@mail.ru](mailto:rav379@mail.ru)

**Mihail F. Stepanov**, Dr. Sci. (Eng.), Assoc. Prof., Department of "Systems Engineering and Control in Technical Systems", e-mail: [rav379@mail.ru](mailto:rav379@mail.ru)

**Конфликт интересов / Conflict of interest.**

Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.

Поступила в редакцию/Received 25.07.2021.

Одобрена после рецензирования/ Revised 07.08.2021.

Принята в печать/Accepted for publication 10.08.2021.