

Джангаров А.И.<sup>1</sup>, Калхиташвили Д.Ш.<sup>2</sup>, Магамедова Д.М.<sup>3</sup>

### Идентификация объектов в видеопотоке

<sup>1</sup>Чеченский государственный университет имени А.А.Кадырова  
(Россия, Грозный)

<sup>2</sup>Российская академия народного хозяйства и государственной службы при Президенте  
Российской Федерации  
(Россия, Москва)

<sup>3</sup>Чеченский государственный педагогический университет  
(Россия, Грозный)

doi: 10.18411/trnio-12-2021-77

#### Аннотация

В настоящее время вектор развития обработки видеопотока направлен на создание автоматизированной системы анализа видеоряда. Особенно актуальна задача нахождения баланса между вычислительной сложностью алгоритмов обработки и аппаратными возможностями современной техники. Специфика видеoinформации создает существенные ограничения на скорость и качество обработки. Актуальной задачей является разработка системы для интеллектуального видеонаблюдения, эффективно использующей вычислительные ресурсы современных компьютеров, чему и посвящается данная работа.

**Ключевые слова:** видеопоток, предобработка кадра, алгоритм, шумы, библиотеки, идентификация движений.

#### Abstract

At present, the vector of development of video stream processing is aimed at creating an automated system for analyzing video sequences. The problem of finding a balance between the computational complexity of processing algorithms and the hardware capabilities of modern technology is especially urgent. The specificity of video information creates significant restrictions on the speed and quality of processing. An urgent task is to develop a system for intelligent video surveillance that effectively uses the computing resources of modern computers, which is what this work is devoted to.

**Keywords:** video stream, frame preprocessing, algorithm, noise, libraries, motion identification.

Первым делом, для идентификации какого-либо объекта в видеопотоке, необходимо совершить предобработку кадра. Первым этапом нужно сделать захват кадра в видеопотоке для его дальнейшей подготовки. Для начала необходимо открыть исходный видеофайл с помощью метода библиотеки OpenCV, который называется VideoCapture. Далее произвести захват кадра с помощью метода read.

Следующим этапом нужно этот кадр обработать, чтобы подготовить его к реализации алгоритма межкадровой разницы, а также, чтобы избавиться от лишних шумов, которые снижают эффективность точной локализации движения. Следует осуществить переход цветного кадра в формате RGB в оттенки серого цвета. Для этого потребуется метод под названием cvtColor (рисунок 1).

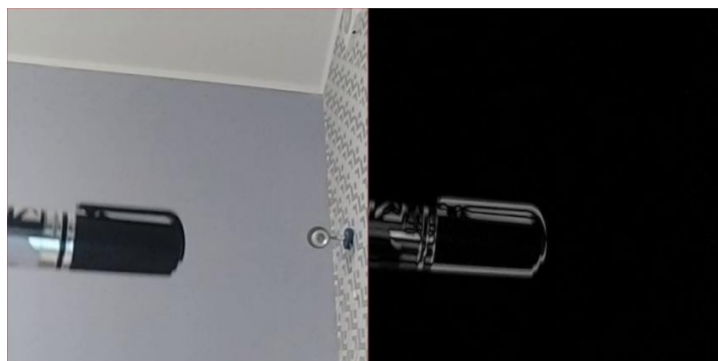


Рисунок 1. Наложение цветового фильтра

Также необходимо понизить четкость кадра, чтобы избавиться от шума фона, это может сделать метод GaussianBlur (рисунок 2).



Рисунок 2. Наложение фильтра размытия

И последний этапом предобработки кадра является создание бинарной маски, которая строится методом сравнения интенсивности каждого пикселя с заданным параметром  $T$ . Построенные такой маски предоставляет метод `threshold`. После выполнения всех этапов предобработки кадра можно приступить к основному алгоритму межкадровой разницы.

Для реализации алгоритма детекция движений с помощью межкадровой разницы необходимо взять два кадра, следующих друг за другом и прошедших предобработку, и сравнить их. Сравнение осуществляется с помощью метода `absdiff`, который принимает в качестве параметров вышеупомянутые два кадра. Метод `absdiff` возвращает изображение, которое является межкадровой разностью и свидетельствует о локации движения какого-либо объекта (картинка межкадровой разницы).

Для построения контура детекции движения нужно взять полученное изображение межкадровой разности в виде бинарной маски и отрисовать на этом изображении контуры, обозначающие локацию движения. Это можно сделать с помощью методов `findContours`, `boundingRect`, `rectangle`. Результат работы алгоритма является исходное изображение с отрисованными контурами детекции движения (рисунок 3).



Рисунок 3. Результат работы алгоритма межкадровой разности

Для практической реализации алгоритма детекции движения в видеопотоке был разработан программный код, который представлен на рисунке 4.

```

172 cap=cv.VideoCapture("video5.mp4") #открытие видео из архива
173 ret, frame1=cap.read() #захват первого кадра в видеопотоке
174 ret, frame2=cap.read() #захват второго кадра
175
176 while cap.isOpened(): # Пока возможно открытие видео
177
178     diff=cv.absdiff(frame1,frame2) #сравнение текущего и предыдущего кадра для детекции движения
179     gray=cv.cvtColor(diff, cv.COLOR_BGR2GRAY) #метод перехода к серым оттенкам
180     blur=cv.GaussianBlur(gray, (7,7), 0) #размытие по Гауссу. Метод размытия картинки для избавления от шумов
181
182     stack_frame=stackImages(0.7,[frame1, gray,blur])
183
184     _, thresh=cv.threshold(blur, 20, 255, cv.THRESH_BINARY) # задаётся пороговое значение и строится бинарная матрица
185
186     dilated=cv.dilate(thresh, None, iterations=1) #расширение элемента кадра
187
188     contours, _=cv.findContours(dilated, cv.RETR_LIST, cv.CHAIN_APPROX_SIMPLE) #поиск контуров по бинарной маске
189
190     for contour in contours: #построение контуров обводки детекции движений для каждого кадра в видеопотоке
191
192         (x,y,w,h)=cv.boundingRect(contour) #получение координат прямоугольника
193
194         if cv.contourArea(contour)< 10000: #условие минимальной области движения
195             continue
196
197         cv.rectangle(frame1, (x,y), (x+w,y+h), (0,255,0), 8) #рисуем прямоугольник для локализации область движения
198
199
200     stack_frame = stackImages(0.2,((frame1, gray,blur],[thresh, dilated, frame2])) #специальная функция для вывода в одном окне
201     cv.imshow("feed", stack_frame) #вывод видеоряда кадрово
202
203     frame1=frame2 #делаем замену контрольного кадра
204     ret, frame2=cap.read() #перезаписываем следующий кадр для обработки
205
206     if cv.waitKey(40)==27: #чтоб закончить работу программы, нажать кнопку esc
207         break # прекращение работы программы
208
209     cv.destroyAllWindows()

```

Рисунок 4. Программный код алгоритма детекции движения в видеопотоке

Представленный программный код реализован в редакторе программного кода Visual Studio Code с использованием языка программирования Python и с использованием библиотеки OpenCV. В результате проделанной работы появился программный код идентификации движений в видеопотоке, который справляется с тестовыми изображениями на которых имеется и несколько движущихся объектов.

Подводя итоги хотелось бы отметить, что область применения видеонаблюдения, а особенно интеллектуального видеонаблюдения, очень обширна, потому что она применяется во многих отраслях деятельности человека как в глобальных, так и в частных. Интеллектуальное видеонаблюдение может решить весьма важные проблемы, связанные с многими сферами деятельности человека. Примером может послужить видеонаблюдение в банках, магазинах, на улицах и тому подобное.

Главным преимуществом интеллектуальной системы видеонаблюдения является ее автоматичность и независимость от человеческого фактора, что существенно повышает эффективность данной системы. В этом и заключается взаимосвязь между автоматичностью и эффективностью, можно сказать, что чем более автоматична система, тем более она эффективна.

\*\*\*

1. Бондарчук В. В., Шевченко Е. А. Исследование алгоритма детекции объектов интереса в видеопотоке // Проблемы искусственного интеллекта. – 2015. – №1. – URL: <https://cyberleninka.ru/article/n/issledovanie-algoritma-deteksii-obektov-interesa-v-videopotoke> (11.01.2021).
2. Девятков В. В., Алфимцев А. Н., Таранян А. Р. Селективно-ковариационный метод локализации, классификации и отслеживания людей в видеопотоках от множества видеокамер // Вестник МГТУ им. Н.Э. Баумана. Серия «Приборостроение». – 2016. – №6 – URL: <https://cyberleninka.ru/article/n/selektivno-kovariatsionnyy-metod-lokalizatsii-klassifikatsii-i-otslezhvaniya-lyudey-v-videopotokah-ot-mnozhestva-videokamer> (29.12.2020).
3. Кудрина М. А., Кирьяков С. А., Кудрин К. А., Дегтярева О. А. Векторизация растровых изображений с использованием преобразования Хафа // НиКа. 2013. – URL: <https://cyberleninka.ru/article/n/vektorizatsiya-rastrovyh-izobrazheniy-s-ispolzovaniem-preobrazovaniya-hafa> (12.01.2021).
4. Leavers, V.F. Shape Detection in Computer Vision Using the Hough Transform / V.F. Leavers. - Berlin: Springer, 2016. – 215 p. – ISBN 978-3540197232
5. Шипицин С. П., Кавалеров М. В. Ускорение работы метода Виолы-Джонса для распознавания объектов на видеопоследовательностях // МНИЖ. 2017. № 7–3 (61).