

## РЕШЕНИЕ ЗАДАЧИ ОБРАБОТКИ ИЗОБРАЖЕНИЯ С КАМЕРЫ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА И ОБНАРУЖЕНИЯ/СЧИТЫВАНИЯ ОПТИЧЕСКИХ МЕТОК ТИПА QR-КОД С ПОМОЩЬЮ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Г.А. Мелкозеров

gendr6338@gmail.com

SPIN-код: 5707-1504

В.М. Дементьев

vlad.eltrogun@gmail.com

*МГТУ им. Н.Э. Баумана, Москва, Российская Федерация*

Исследованы возможные решения задачи распознавания изображения с видеокamеры с последующим обнаружением и считыванием оптических меток типа QR-код с целью дальнейшей реализации алгоритма для распознавания QR-кода на видеоизображении. Изучены решения задачи обработки и анализа изображения с видеокamеры с целью обнаружения и считывания оптических меток типа QR-кода. Рассмотрен основополагающий подход к использованию компьютерного зрения для обработки и анализа видеопотока. Приведен ряд решений, помогающих реализовать поставленную задачу. В результате исследования реализован алгоритм обнаружения и считывания QR-кода с последующей записью данных и представления их в форме таблицы Excel с использованием библиотек OpenCV и XlsxWriter. Результаты работы могут быть полезны для формирования машинного зрения робототехнических систем, в частности, для беспилотного летательного аппарата типа квадрокоптера.

**Ключевые слова:** робототехника, автоматизация, беспилотные автоматические системы, беспилотный летательный аппарат, дрон, распознавание QR-кода, компьютерное зрение, обработка изображений, образовательная робототехника, OpenCV

**Введение.** Одним из самых новейших и быстроразвивающихся направлений для внедрения автоматических систем является применение беспилотных летательных аппаратов (БПЛА). Сфера применения БПЛА расширяется с каждым днем вместе со списком задач, поставленным перед их аппаратной частью. Использование компьютерного зрения для анализа данных с видеокamеры является актуальной задачей во всех сферах применения БПЛА, в частности, в задачи считывания оптических меток [1]. Оптические метки применяются в случаях позиционирования и ориентирования в пространстве при автоматизированном управлении БПЛА, а также для получения информации о наблюдаемых объектах, например, при считывании QR/Var-кодов, расположенных на предметах складских помещений. Автоматизация такого процесса, как инвентаризация складских помещений, может существенно увеличить скорость дан-

ной процедуры, снизить затраты на проведение инвентаризации, а также уменьшить риски, обусловленные применением человеческого фактора.

В данной статье рассмотрены существующие решения задачи обработки изображения и обнаружения/считывания оптических меток типа QR-кода при помощи компьютерного зрения с целью реализации алгоритма в ходе испытаний.

**Цифровая обработка изображений.** Подходя к задаче обработки изображения для дальнейшего считывания оптических меток, можно найти несколько решений, в каждом из которых применяются алгоритмы компьютерного зрения. Компьютерное зрение (Computer Vision, CV) — совокупность технологий, методов и алгоритмов, с помощью которых компьютер может обрабатывать изображения и видеопоток. Использование CV позволяет определять, что изображено, классифицировать эти изображения и анализировать их с целью дальнейшего применения в качестве машинного зрения (Machine Vision, MV), решая поставленные задачи автоматизации. Целью MV является замена использования зрительной системы человека применением камеры и компьютера, которые позволяют собрать больше полезной информации о визуальном мире и игнорировать ненужную информацию [2]. Пересечение задач, поставленных перед машинным зрением, схематически показано на рис. 1.



Рис. 1. Элементы технологии машинного зрения

Системы, позволяющие эффективно распознавать визуальную информацию и на основе ее анализа решать сложные задачи управления и контроля, принято называть системами машинного зрения. В общем виде они представляют собой взаимосвязанную технологическую последовательность, включающую следующие звенья [2]:

- 1) получение изображения от видеокамеры;
- 2) обработку (оцифровку) изображения;

3) логический анализ цифрового изображения и выделения нужной информации;

4) перемещение камеры в пространстве.

В частности, компьютерное зрение решает поставленную задачу обработки видеоизображения: распознать метку, расположенную на объекте, и считать закодированную в ней информацию для дальнейшей работы с ней.

Разработчики компьютерного зрения чаще всего используют язык Python или C++, а также специализированные библиотеки. Регулярно пополняются датасеты для обучения нейронных сетей, проводятся соревнования для поиска лучших решений [2].

**Существующие решения распознавания и считывания оптических меток типа QR-кода.** Для распознавания оптических меток на данный момент существует множество решений: Amazon Rekognition, BoofCV, CUDA, GPUImage, Keros, MatLab, OpenCV, SimpleCV, TensorFlow, Theano и др., но мы рассмотрим основные из них, которые широко распространены чаще всего применяются при решении поставленной задачи распознавания QR-кодов: TensorFlow, OpenCV.

TensorFlow — бесплатная библиотека с открытым исходным кодом для потоков данных и дифференциального программирования. Это символьная математическая библиотека, которая дополнительно используется для приложений машинного обучения, например, нейронных сетей. Известность ее быстро возросла и превзошла известность всех существующих библиотек благодаря простоте API [2]. Пример работы библиотеки TensorFlow показан на рис. 2.



Рис. 2. Пример работы TensorFlow

OpenCV — самая известная библиотека, многоплатформенная и простая в использовании. Охватывает все основные стратегии и алгоритмы для выполнения некоторых задач обработки изображений и видео, превосходно работает с C++ и Python [3]. Библиотека OpenCV [4] обладает встроенной системой детектора QR-кодов — остается обнаружить, расшифровать и вывести данные о QR-коде. При этом пользователю не нужно прицеливаться камерой — все производится автоматически: поиск, определение типа и считывание. Система устойчива к перепаду освещения, избытку и недостатку света, ракурсу съемки и перекосам, что позволит снизить требования по угловой стабилизации квадрокоптера. Пример работы программы по распознаванию QR-кода приведен на рис. 3.

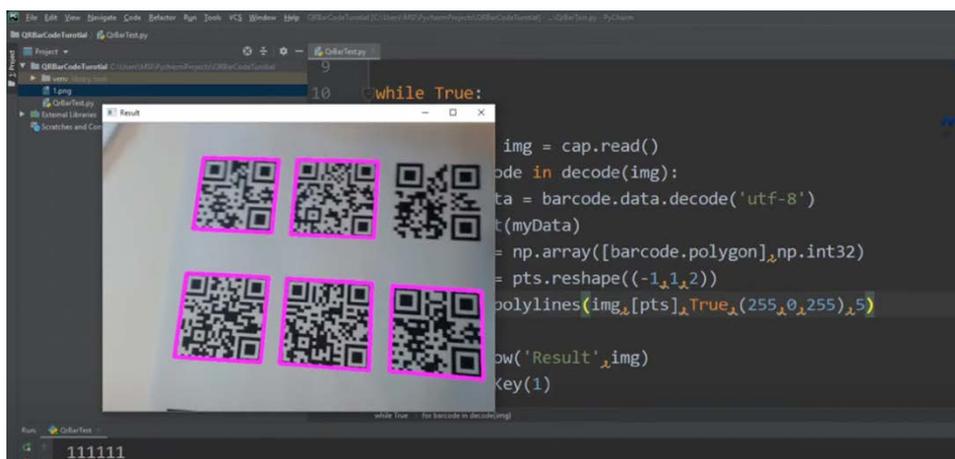


Рис. 3. Пример работы алгоритма по распознаванию QR-кода с помощью OpenCV

В данном случае используются готовые алгоритмы распознавания шифровок. Сначала происходит инициализация изображения с камеры, а алгоритм получает изображение и начинает искать метки. Далее в работу включается блок расшифровок, который распознает кодировку, затем полностью считывает QR-код и составляет бинарный код. Данную систему в дальнейшем планируется применить для обработки изображений с камеры БПЛА: либо в реальном времени, либо для пост-обработки отснятого видеопотока.

**Реализация алгоритма обнаружения и считывания оптических меток типа QR-кода.** Реализация алгоритма обнаружения и считывания оптической метки типа QR-кода по видеопотоку с камеры происходит в несколько этапов:

- 1) получение изображения с видеокamеры;
- 2) обработка изображения;

3) логический анализ цифрового изображения и выделение нужной информации;

4) считывание и запись расшифрованной информации с QR-кода.

Таким образом, для получения видеоизображения мы используем внешнюю видеокамеру, в нашем случае мы используем Logitech C922 Pro Stream с разрешением 1080p, которую подключаем к ноутбуку с установленной интегрированной средой разработки PyCharm для языка программирования Python, на котором и будет реализована программа обнаружения и считывания QR-кодов. Техническое оснащение представлено на рис. 4.



Рис. 4. Техническое оснащение для реализации алгоритма

Также необходимо сформировать QR-коды с URL-ссылками. Для экспериментальных тестов программы были распечатаны и повешены на стену три тестовых QR-кода. Сканируемые QR-коды содержат в себе URL-адреса, которые необходимо записать в файл. Для удобства работы с данными реализована запись считанных адресов в файл Excel, где построчно записываются считанные адреса, исключая повторения. При написании кода программы была использована библиотека OpenCV [4] для работы с видеопотоком, обнаружением и считыванием QR-кодов, а также библиотека XlsxWriter [5], позволяющая сформировать на выходе работы программы файл с информацией, расшифрованной с QR-кодов.

Результаты работы программы, реализованной в среде PyCharm на языке Python, представлены на рис. 5 [6]. В ходе работы программа получает видеоизображение с внешней веб-камеры, непрерывно обрабатывает и анализирует изображение в реальном времени и считывает QR-коды. При корректном считывании границы QR-кода выделяются зелеными рамками. Код программы представлен ниже:

**Программа компьютерного зрения**

```
import cv2 #подключение библиотеки openCV
import xlsxwriter #подключение библиотеки Excel

camera_id = 0
delay = 1
window_name = 'OpenCV QR Code'
```

```
#блок создания файла Excel и таблицы в нем
workbook = xlswriter.Workbook("qr-codes.xlsx")
worksheet = workbook.add_worksheet()
worksheet.set_column("A:A", 50)
bold = workbook.add_format({"bold": True})
sheetCnt = 0

#блок работы с изображением с камеры
qcd = cv2.QRCodeDetector()
cap = cv2.VideoCapture(camera_id)
qrs = [str]

while True: #цикл считывания кадров видеопотока
    ret, frame = cap.read()

    if ret:
        ret_qr, decoded_info, points, _ = qcd.detectAndDecodeMulti(frame)

        if ret_qr:
            for s, p in zip(decoded_info, points):
                if s:
                    found = False
                    for q in qrs:
                        if str(s) == q:
                            found = True
                    if not found:
                        qrs += [str(s)]
                        worksheet.write(sheetCnt, 0, s)
                        sheetCnt = sheetCnt + 1
                    print(s)
                    color = (0, 255, 0)
                else:
                    color = (0, 0, 255)
                frame = cv2.polylines(frame, [p.astype(int)], True, color, 8)
            cv2.resizeWindow(window_name, 1920, 1280)
            cv2.imshow(window_name, frame)

            if cv2.waitKey(delay) & 0xFF == ord('q'):
                break

cv2.destroyWindow(window_name)
workbook.close()
```

По завершении работы программы расшифрованная информация представляется в виде excel таблицы (рис. 6), где построчно записаны URL-адреса считанных QR-кодов, что позволяет продолжить работу над считанными данными.

**Использование среды разработки PyCharm для управления БПЛА.** Испытательным образцом в качестве БПЛА был выбран дрон класса мини DJI Ryze Tello. Основная особенность DJI Ryze Tello — возможность программирования этого квадрокоптера с использованием языка Python [7]. Этот метод идеально подходит для выполнения поставленной задачи и позволит запрограммировать дрон для выполнения запрограммированной траектории движения.

Для начала необходимо определить и поставить список задач для полетной миссии дрона:

- 1) осуществить автоматический взлет дрона;
- 2) получить видеопоток с камеры дрона;
- 3) с помощью ранее разработанного алгоритма компьютерного зрения выполнить задачу обнаружения и считывания QR-кодов по видеопотоку с дрона;

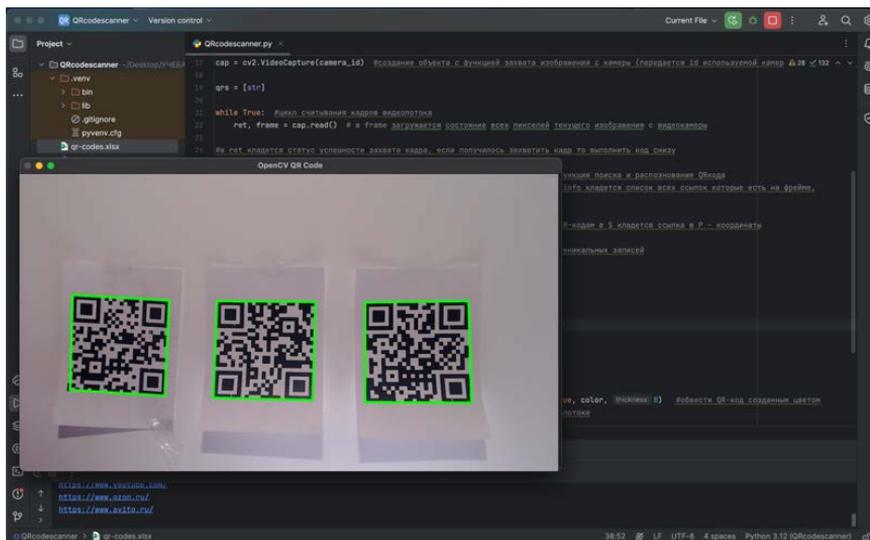


Рис. 5. Результат работы программы по обнаружению и считыванию QR-кода

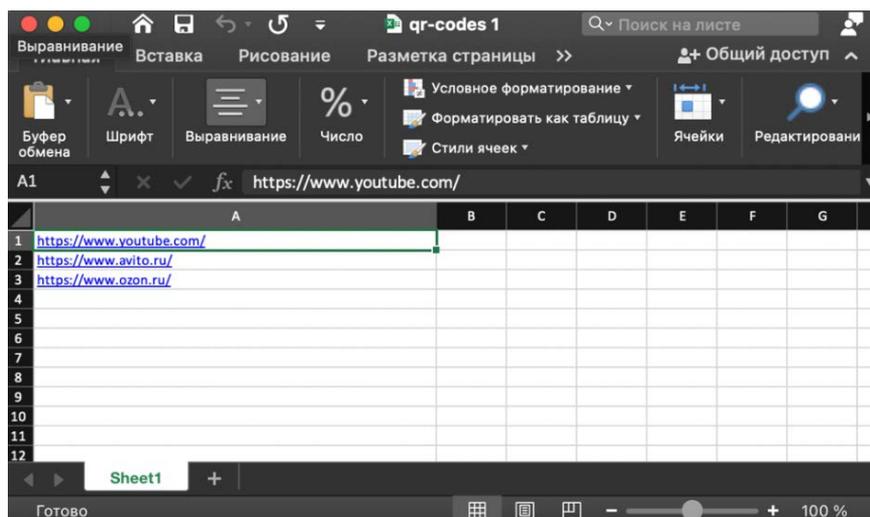


Рис. 6. Результат считывания QR-кодов в виде таблицы Excel

- 4) считанную информацию записать и сохранить в формате таблицы Excel;
- 5) совершить автоматическую посадку дрона;
- 6) сохранить видеопоток с камеры дрона с результатами распознавания QR-кодов.

Основные функции программного кода на языке Python для решения вышеперечисленных задач представлены ниже.

### **Программа автоматизированной системы для БПЛА**

```
import sys
import threading
import time

import cv2
from djitellopy import tello, Tello
import xlswriter

def initialize_drone():
    """ Initializes the drone and connects to it, returning
        the drone object. """
    drone = tello.Tello()
    drone.connect()
    print(f"Drone Battery Level: {drone.get_battery()}%")
    drone.streamoff() # Ensure any previous stream is turned off
    drone.streamon()
    return drone

def control_drone(drone: Tello):
    """ Controls the drone's movement in a separate thread. """
    drone.takeoff()
    time.sleep(1) # Short pause after takeoff for stability.
    drone.move_up(120)
    time.sleep(1) # Hold position.
    drone.move_right(60)
    drone.move_down(50)
    drone.land()

def detect_and_record_qr_codes(frame, workbook, worksheet, recognized_qrs):
    """ Detect and record QR codes from the video frame. """
    qcd = cv2.QRCodeDetector()
    ret_qr, decoded_info, points, _ = qcd.detectAndDecodeMulti(frame)
    if ret_qr:
        for s, p in zip(decoded_info, points):
            if s and s not in recognized_qrs:
                recognized_qrs.add(s)
                worksheet.write(len(recognized_qrs) - 1, 0, s)
                print(s)
                color = (0, 255, 0)
            else:
                color = (0, 0, 255)
            frame = cv2.polylines(frame, [p.astype(int)], True, color, 8)
    return frame

def stream_video(drone, save_video=False, recognize_qr=False):
    """ Captures video from the drone and displays it in a window.
        Optionally saves the video and recognizes QR codes.
    """
    workbook = xlswriter.Workbook("qr-codes.xlsx") if recognize_qr else
```

```

None
worksheet = workbook.add_worksheet() if workbook else None
recognized_qrs = set() # Set to store recognized QR codes
if workbook:
    worksheet.set_column("A:A", 50)

if save_video:
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('output.avi', fourcc, 20.0, (648, 488))

try:
    while True:
        frame = drone.get_frame_read().frame
        if frame is not None:
            if recognize_qr:
                frame = detect_and_record_qr_codes(frame, workbook,
                                                    worksheet, recognized_qrs)
                resized_frame = cv2.resize(frame, (648, 488))
                if save_video:
                    out.write(resized_frame)
                cv2.imshow("Drone Camera Stream", resized_frame)
                if cv2.waitKey(1) & 0xFF == ord('q'):
                    break
            else:
                print("Failed to capture frame")
                break
finally:
    if save_video:
        out.release()
    if workbook:
        workbook.close()
    drone.streamoff()
    cv2.destroyAllWindows()

def main(save_video=False, recognize_qr=False):
    """ Main function to handle initialization, streaming, and optional
        features. """
    try:
        drone = initialize_drone()
        threading.Thread(target=control_drone, args=(drone,),
                        daemon=True).start()
        stream_video(drone, save_video, recognize_qr)
    except Exception as e:
        print(f"An error occurred: {e}")
        sys.exit(1)

if __name__ == '__main__':
    main(save_video=True, recognize_qr=True) # Toggle features as needed

```

Опишем основные его функции.

### 1. Импорт и настройка.

Импортирует необходимые библиотеки Python, такие как **sys**, **threading**, **time**, **cv2** (OpenCV для обработки изображений), **djitellopy** (библиотека для управления дронами DJI Tello) и **xlsxwriter** (для записи данных в файлы Excel).

### 2. Функция **initialize\_drone**:

– подключается к дрону и инициализирует потоковое видео;

- выводит уровень заряда батареи дрона;
- убеждается, что все предыдущие видеопотоки отключены, прежде чем включать их снова;

- Возвращает инициализированный **drone** объект.

### 3. Функция **control\_drone**:

- управляет перемещениями дрона в отдельном потоке;
- команды включают взлет, движение вверх и посадку. Команды движения закоментированы.

### 4. Функция **detect\_and\_record\_qr\_codes**:

- использует OpenCV для обнаружения QR-кодов в кадрах, снятых с дрона;

- при обнаружении QR-кодов они записываются в файл Excel и отмечаются на видеокадре ломаной линией;

- QR-коды хранятся в виде набора, чтобы избежать дублирования записей, и их наличие или отсутствие изменяет цвет ломаной линии на красный.

### 5. Функция **stream\_video**:

- обрабатывает потоковое видео с дрона;
- при необходимости видео сохраняется в файл и распознает QR-коды;
- отображает видео в окне и позволяет выйти из цикла, нажав кнопку «q»;
- очищает ресурсы, такие как закрытие видеофайла, книги Excel, отключение видеопотока дрона и уничтожение всех окон OpenCV.

### 6. Функция **main**:

- служит точкой входа в программу;
- инициализирует беспилотник и запускает управление и потоковое видео в отдельных потоках;
- обрабатывает исключения и завершает работу программы при возникновении ошибки.

### 7. Триггер выполнения.

Если скрипт запускается напрямую (не импортируется), он вызывает функцию **main** с опциями для сохранения видео и распознавания QR-кодов на основе параметров функции. Каждая часть этого скрипта является модульной и решает конкретные задачи, что упрощает его обслуживание и модификацию. Применение потоковой передачи позволяет управлять движением дрона и потоковым видео одновременно, повышая оперативность реагирования.

Фотография проводимых испытаний представлена на рис. 7. QR-коды распечатаны и вывешены на стену, на расстоянии 1 м от стены расположен дрон DJI Ryze Tello [8]. Было проведено 10 испытаний, в ходе которых менялось расположение QR-кодов, а соответственно, изменялась и траектория.



**Рис. 7.** Экспериментальный полет квадрокоптера DJI Ryze Tello со сканированием QR-кодов

	A	B
1	ITEM#1	checked
2	ITEM#2	checked
3	ITEM#3	checked
4	ITEM#4	checked
5		

**Рис. 8.** Формируемый отчет в виде таблицы Excel

Снимок экрана с отчетом, формируемым в ходе полетной миссии, показан на рис. 8.

В ходе испытаний была выполнена следующая полетная миссия: дрон был размещен в начальной точке на расстоянии 1 м от стены. На компьютере в среде PyCharm была запущена программа управления дроном. В ходе вы-

полнения программы дрон успешно следовал заданной траектории с установленной скоростью. Одновременно велась передача видео с камеры дрона. В отдельном окне в реальном времени отображалась работа алгоритма компьютерного зрения, распознающего QR-коды, данные которых сохранялись в таблицу Excel для отчета. По завершении миссии дрон приземлился. Видео было сохранено в формате AVI, а отчет — в формате Excel.

**Заключение.** В рамках данной работы было проведено исследование в области компьютерного и машинного зрения. Изучены существующие решения задачи обработки и анализа изображения, обнаружения и считывания оптических меток типа QR-кода.

Описано техническое оснащение необходимое для реализации поставленной задачи. С использованием библиотек OpenCV и XlsxWriter реализован алгоритм обнаружения и считывания QR-кода с последующей записью данных и представления в форме таблицы Excel.

На базе реального БПЛА квадрокоптерного типа реализовано компьютерное зрение, обеспечено автоматическое управление дрона, с использованием языка Python в среде PyCharm. Проведены летные испытания образца дрона с отработкой алгоритма компьютерного зрения и полетного задания.

Применение технологий технического зрения и машинного обучения позволяет автоматизировать процесс обнаружения и сканирования специальных меток при проведении складской инвентаризации, что существенно увеличивает скорость данной процедуры, снижает затраты на проведение инвентаризации, а также риски, обусловленные наличием человеческого фактора.

## Литература

- [1] Головастов А. Машинное зрение и цифровая обработка изображений. *Современная электроника и технологии автоматизации: электрон. журн.*, 2010, № 4. URL: <https://www.cta.ru/articles/cta/obzory/apparatnye-sredstva/124854/> (дата обращения 10.03.2024).
- [2] Корешкова Т.А. *Компьютерное зрение: технологии, компании, тренды.* URL: <https://rdc.grfc.ru/2021/04/analytics-computer-vision/> (дата обращения 10.03.2024).
- [3] Винер Н. *Кибернетика, или управление и связь в животном и машине.* Москва, Наука. Глав. ред. изданий для зарубежных стран, 1983, 344 с.
- [4] OpenCV в Python. Часть 1. URL: <https://habr.com/ru/articles/519454/> (дата обращения 10.03.2024).
- [5] Музаммил Хан. *Python QR Code Reader: как создать высокопроизводительный сканер QR-кода.* URL: <https://blog.aspose.com/ru/barcode/python-qr-code-reader/> (дата обращения 10.03.2024).

- [6] *Глазами компьютера: как работает машинное зрение*. URL: <https://formatkoda.ru/blog/glazami-kompyutera-kak-rabotaet-mashinnoe-zrenie/> (дата обращения 10.03.2024).
- [7] McNamara J. *Creating Excel files with Python and XlsxWriter. XlsxWriter documentation*. URL: <https://xlsxwriter.readthedocs.io/> (дата обращения 10.03.2024).
- [8] Tello SDK 2.0 User Guide. URL: <https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20SDK%202.0%20User%20Guide.pdf> (дата обращения 10.03.2024).

***Поступила в редакцию 24.10.2024***

**Мелкозеров Геннадий Алексеевич** — студент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

**Дементьев Владислав Максимович** — студент кафедры «Робототехнические системы и мехатроника», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

**Ссылку на эту статью просим оформлять следующим образом:**

Мелкозеров Г.А., Дементьев В.М. Решение задачи обработки изображения с камеры БПЛА и обнаружения/считывания оптических меток типа QR-код при помощи компьютерного зрения. *Политехнический молодежный журнал*, 2025, № 04 (99). URL: <https://ptsj.bmstu.ru/catalog/arse/airdev/1055.html>

## SOLUTION TO THE PROBLEM OF PROCESSING IMAGES FROM A UAV CAMERA AND DETECTING/READING OPTICAL MARKS TYPE QR CODE USING COMPUTER VISION

**G.A. Melkozerov**

gendr6338@gmail.com

SPIN-code: 5707-1504

**V.M. Dementev**

vlad.eltrogun@gmail.com

*Bauman Moscow State Technical University, Moscow, Russian Federation*

This article examines existing solutions to the problem of image recognition from a video camera with subsequent detection and reading of optical marks such as QR code, with the aim of further implementing an algorithm for recognizing a QR code on a video image. The work carried out a study of existing solutions to the problem of processing and analyzing images from a video camera in order to detect and read optical tags such as QR codes. The fundamental approach to using computer vision for processing and analyzing video streams has been studied. A number of existing solutions that help to achieve the task are given. As a result of the research, an algorithm for detecting and reading a QR code was implemented, followed by recording the data and presenting it in the form of an Excel table, using the OpenCV and XlsxWriter libraries. The results of the work can be useful in the formation of computer vision of robotic systems, in particular for an unmanned aerial vehicle (UAV) such as a quadcopter.

**Keywords:** robotics, automation, unmanned automatic systems, unmanned aerial vehicle, drone, QR code recognition, computer vision, image processing, educational robotics, Open

***Received 24.10.2024***

**Melkozerov G.A.** — Student, Department of Robotic Systems and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.

**Dementev V.M.** — Student, Department of Robotic Systems and Mechatronics, Bauman Moscow State Technical University, Moscow, Russian Federation.

### **Please cite this article in English as:**

Melkozerov G.A., Dementev V.M. Solution to the problem of processing images from a UAV camera and detecting/reading optical marks type QR code using computer vision. *Politekhnikheskiy molodezhnyy zhurnal*, 2025, no. 04 (99). (In Russ.). URL: <https://ptsj.bmstu.ru/catalog/arse/airdev/1055.html>