Отчет по лабораторной работе 3

Студента группы №3331506/00401 Орехова Алексея Михайловича

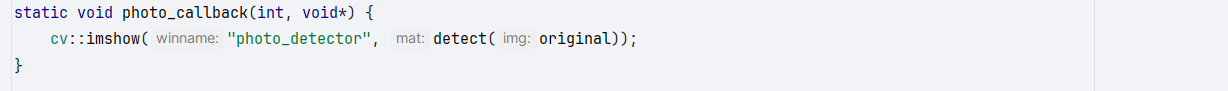
Код работы выложен на [гитхаб](https://github.com/alexeyorehoff/cv-lessons)

Первое задание работы заключалось в детектировании тепловых сигнатур на черно-белом изображении для статичной картинки и видео, как дополнительное задание. С этой целью была создана программа, представленная на рисунке 1.



# Рисунок 1

Оригинальное фото загружается из файла в виде матрицы, а для видео использован объект cv::videoCapture. Выполняются проверки валидности импорта данных, создаются пустые окна, необходимые для корректной работы слайдеров для настройки пороговой фильтрации. В качестве callback функции для обработки статичного изображения используется функция, показанная на рисунке 2. Она позволяет выполнять пересчет функции детектирования только когда изменяется значение порога для фильтрации. В случае видео она не используется, поскольку новые кадры и так обрабатываются в реальном времени вне зависимости от движения слайдера. Обработка кадров происходит в бесконечном цикле с выходом по нажатию на клавишу Esc, а после окончания видео, оно проигрывается сначала. Также, на рисунке 2 приведена функция обработки. После предварительной обработки она определяет контуры на изображении, находит самый крупный из них по площади и обрисовывает его очертания на изображении. Центр контура вычисляется с использованием моментов контура.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

# Рисунок 2

Результат выполнения задания 1 представлен на рисунке 3.

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, Графическое программное обеспечение, программное обеспечение

Автоматически созданное описание

Рисунок 3

Задание 2 идентично заданию 1, за исключением того, что теплые объекты изображены на изображении не в градациях серого, а в псевдоцветах, где синий – холодный и красно-белый – горячий. Для этого задания на этапе предварительной обработки изображение было переведено из формата BGR в HSV, где из него был выделен канал оттенка. По нему с помощью порогового фильтра были выделены горячие места на изображении. Результат выполнения задания 2 представлен на рисунке 4.

Изображение выглядит как снимок экрана, Операционная система, программное обеспечение, компьютер

Автоматически созданное описание

# Рисунок 4

В рамках задания 3 требовалось детектировать разноцветных роботов на видео (было выполнено сразу доп. Задание, поскольку обработка для статичного изображения, по сути, входит в него). Для него были выбраны значения пикселей в цветовом пространстве HSV, чтобы стабильно определять большое количество пикселей нужного цвета. На этом этапе возникли трудности с определением зеленого, поскольку все роботы, независимо от цвета, содержат в себе печатную плату, которую видно на видео, и которую детектор стабильно определял как зеленую область. С целью избежать этого, был добавлен дополнительный фильтр по площади цветной области, подобранный вручную.

Источник света был определен в цветовом пространстве RGB как содержащий высокие значения всех компонент цвета (т.е. яркий белый). Далее для каждой «команды» роботов был вычислен робот, находящийся ближе к лампочке, на нем была нарисована точка соответствующего цвета. Также, авторы оригинального видео использовали лампочку с красным диском вокруг нее, который принимался за робота. Для исправления этого, была добавлена «запрещенная зона» вокруг лампочки. Результат работы программы, созданной для задания 3 представлен на рисунке 5.

Изображение выглядит как снимок экрана, в помещении

Автоматически созданное описание

# Рисунок 5

Для задания 4 было необходимо выполнить отбраковку гаечных ключей на изображении, пользуясь эталонной маской. В рамках этого задания, из полученных изображений были получены контуры, которые требовалось сравнить с помощью определенной метрики. Для этого была использована встроенная функция cv::matchShapes с методом cv::CONTOURS\_MATCH\_I2. Данный конкретный метод был выбран как дающий наибольшую разницу между «хорошими» и «плохими» изделиями. На основании величины метрики на исходном изображении отрисовывался текст «bad» или «good». Результаты представлены на рисунке 6.

Изображение выглядит как металлоизделия, рычаг

Автоматически созданное описание

# Рисунок 6