Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Алгоритмы цифровой обработки мультимедиа**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кислица Д.А.

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Крамаренко

Лабораторная работа №2

Реализация трекинга

**Задание 1.** Прочитать изображение с камеры и перевести его в формат HSV.

Переведём изображение в формат HSV с помощью ранее используемой команды cv2.cvtColor(), указав в аргументах флаг HSV cv2.COLOR\_BGR2HSV.

RGB и HSV — это два различных цветовых пространства, которые используются для представления цветов. RGB — это аддитивное цветовое пространство, которое используется для представления цветов на экране. Оно состоит из красного, зеленого и синего цветов. HSV — это цветовое пространство, которое используется для представления цветов в виде оттенка, насыщенности и значения. Он также называется HSB (оттенок, насыщенность, яркость). Он позволяет легче управлять цветом, чем RGB.

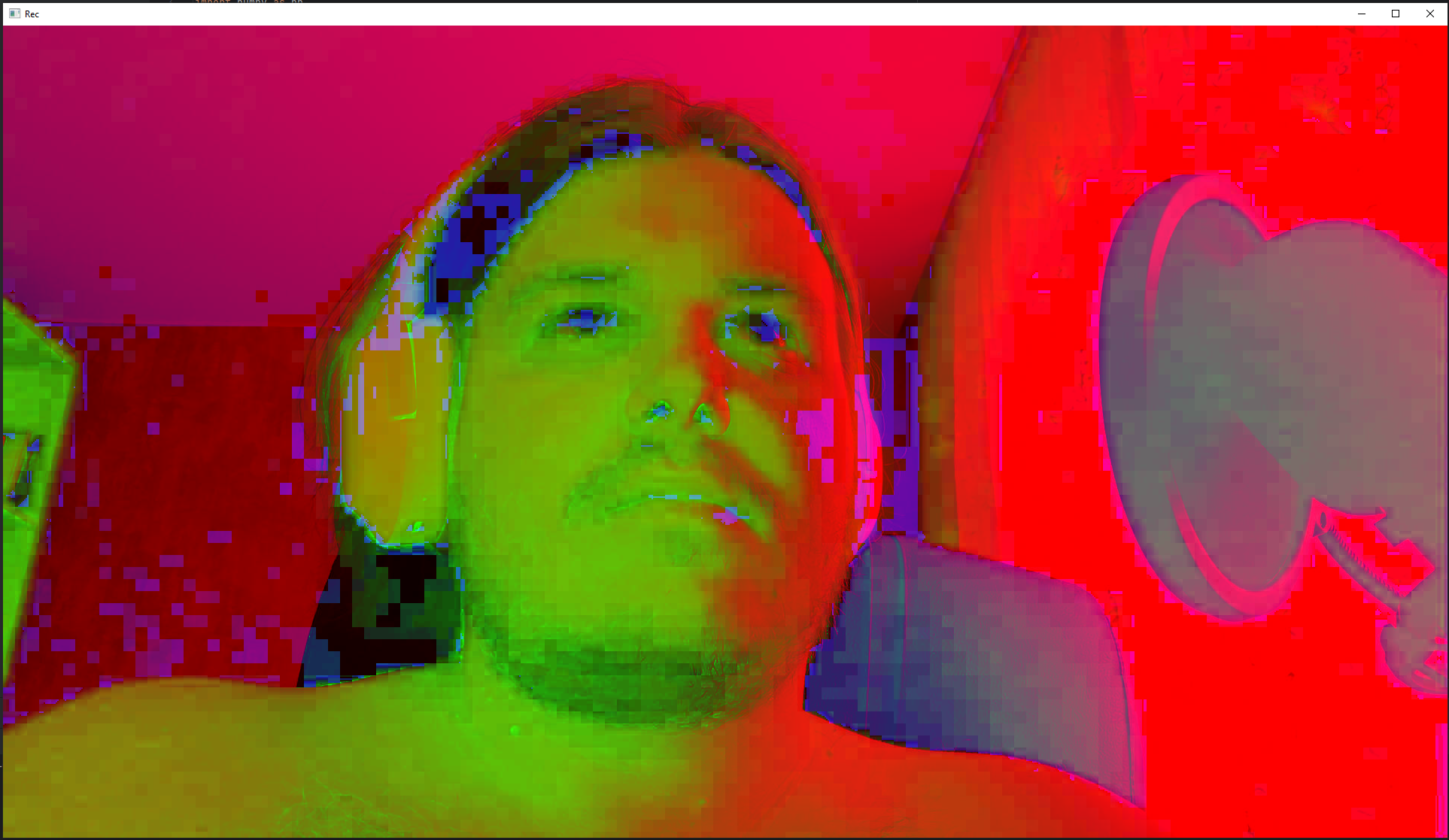


Рисунок 1 – HSV формат

**Задание 2.** Применить фильтрацию изображения с помощью команды inRange и оставить только красную часть, вывести получившееся изображение на экран(treshold), выбрать красный объект и потестировать параметры фильтрации, подобрав их нужного уровня.

Преобразуем кадр в цветовое пространство HSV при помощи cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV), что облегчает фильтрацию по цвету.

Определим цветовую маску при помощи метода cv2.inRange(), который создает маску, которая выделяет области, соответствующие заданному диапазону красного цвета. Далее при помощи cv2.bitwise\_and() применим маску к кадру, оставляя только пиксели, соответствующие красному цвету.



Рисунок 2 – выделение красного цвета

**Задание 3.** Провести морфологические преобразования (открытие и закрытие) фильтрованного изображения, вывести результаты на экран, посмотреть смысл подобного применения операций erode и dilate.

Морфологические преобразования - это операции, применяемые к изображениям с целью изменения их формы и структуры на основе формы и структуры объектов на изображении. Два из наиболее распространенного морфологического преобразования - это "открытие" и "закрытие". Эти операции часто используются в обработке изображений для удаления шума, заполнения дыр в объектах и изменения размера объектов. Давайте рассмотрим каждую из них более подробно:

Операция открытия состоит из двух шагов: сначала применяется эрозия (erode), а затем – дилатация (dilate). Сначала эрозия удаляет маленькие объекты и "шум" на изображении, уменьшая объекты и заполняя небольшие прорехи. Затем дилатация восстанавливает объекты близкой к их исходному размеру.

Открытие полезно для удаления шума, разделения объектов, связанных друг с другом, и выделения объектов, близких к заданной форме.

Операция закрытия также состоит из двух шагов: сначала применяется дилатация (dilate), а затем – эрозия (erode). Сначала дилатация увеличивает объекты и заполняет небольшие отверстия в объектах. Затем эрозия уменьшает объекты обратно к их исходному размеру. Закрытие полезно для закрытия небольших отверстий в объектах и объединения близко расположенных объектов.

Применение открытия и закрытия может быть полезным при обработке изображений в различных задачах компьютерного зрения. Эти операции особенно эффективны в тех случаях, когда объекты на изображении имеют различные размеры или, когда на изображении присутствует некоторый уровень шума. Использование морфологических преобразований требует выбора правильных структурирующих элементов (ядро) и настройки их размера в зависимости от конкретной задачи.

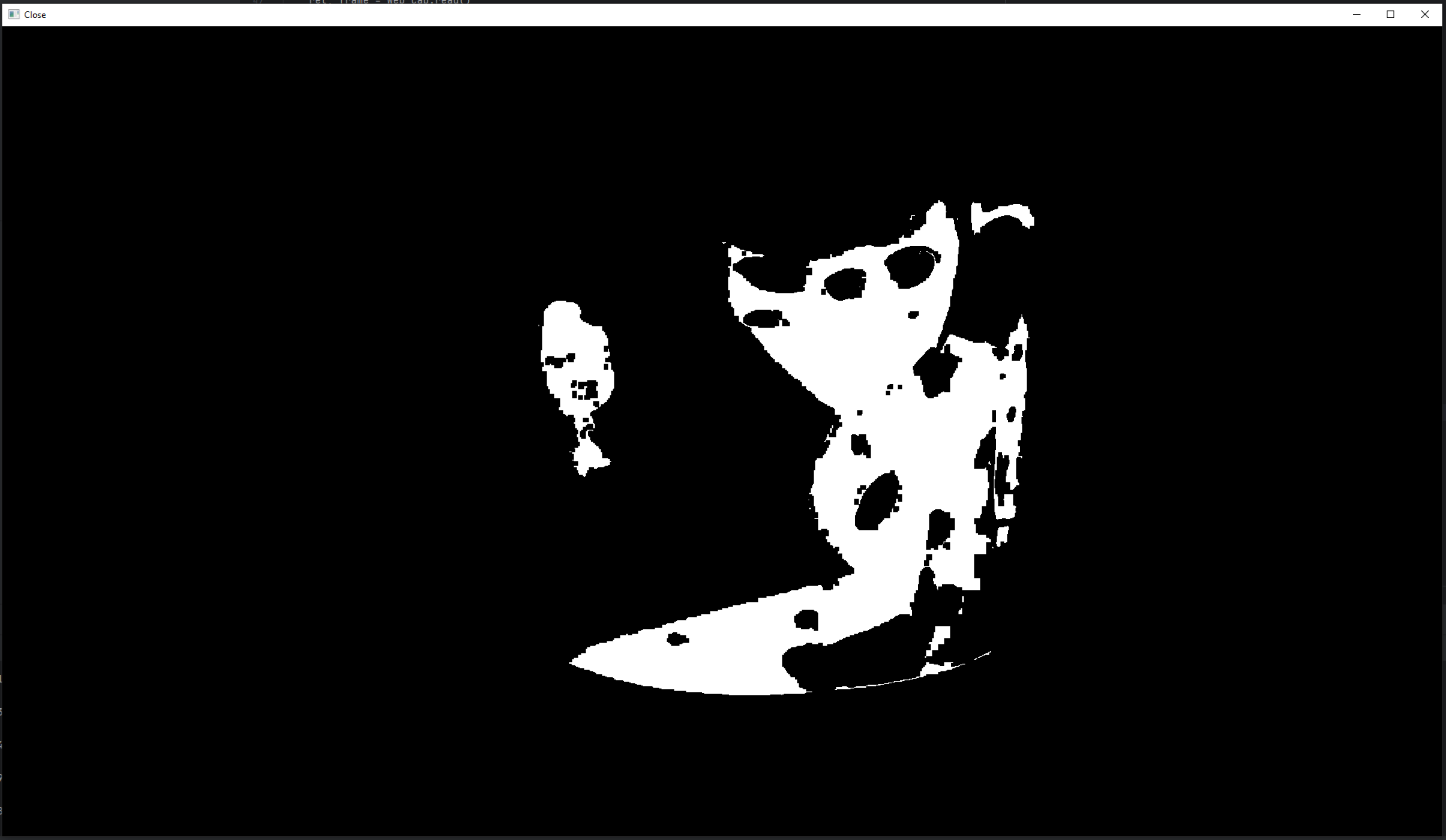


Рисунок 3 – операция close



Рисунок 4 – операции open

**Задание 5.** На основе анализа площади объекта найти его центр и построить черный прямоугольник вокруг объекта. Сделать так, чтобы на видео выводился полученный черный прямоугольник, причем на новом кадре.

Моменты изображения - это статистические показатели, которые описывают форму, структуру и распределение пикселей в изображении.

В нашем случае мы используем метод cv2.moments().

m00 - это один из моментов изображения, который называется нулевым моментом. Нулевой момент представляет собой интегральную яркость (или массу) всего изображения и используется для вычисления центра масс объекта.

По координатам m10 и m01 можно вычислить координаты центра масс объекта.

Далее рисуется черный прямоугольник вокруг центра масс объекта.

Рисунок 5 – трекинг красного цвета с прямоугольником

**Ответы на вопросы:**

1. Функция inRange используется для фильтрации пикселей в пределах заданного диапазона значений в изображениях. Функция принимает три параметра: входное изображение, нижнюю границу диапазона и верхнюю границу диапазона. Пикселям, попадающим в указанный диапазон, присваивается значение белого, в то время как пикселям, выходящим за пределы диапазона, присваивается значение черного
2. Методы размывания и расширения изменяют форму и структуру объектов на изображении с использованием математических преобразований.

Стирание: метод стирания уменьшает размер объекта переднего плана и стирает границы пикселей объекта. Он удаляет пиксели на границах объекта и уменьшает изображение. Эрозия полезна для таких задач, как подавление шума и удаление мелких объектов или деталей с изображения.

Расширить: метод расширения увеличивает размер объекта переднего плана и расширяет границы пикселей объекта. Это добавляет пиксели к границам объекта и увеличивает его размер. Расширение полезно для таких задач, как заполнение отверстий, соединение разорванных частей объекта или утолщение линий на изображении.

1. Морфологическое открытие и закрытие - это сочетание эрозии и расширения, которые используются для удаления шума и мелких объектов, а также для заделки небольших отверстий или зазоров.

Opening: Морфологическое вскрытие - это сочетание эрозии с последующим расширением. Открытие помогает сгладить границы объектов и разделить связанные объекты.

Closing: Морфологическое закрытие - это сочетание расширения с последующей эрозией. Закрытие помогает завершить форму объекта и устранить небольшие пробелы.

4. Моменты изображения - числовые характеристики, описывающие распределение интенсивности пикселей в изображении. Различают несколько типов моментов:

* Геометрические моменты - описывают форму и размер объекта. Нулевой момент равен площади объекта, первые моменты - координатам центра масс (центроида), центральные моменты инвариантны к сдвигу.
* Физические моменты - связаны с физическими характеристиками объекта, например, массой.
* Алгебраические моменты - используются для построения инвариантных признаков объекта, например, для распознавания.

Моменты широко применяются в задачах компьютерного зрения для анализа формы объектов, вычисления их характеристик, распознавания и классификации.

m00 - это один из моментов изображения, который называется нулевым моментом. Нулевой момент представляет собой интегральную яркость (или массу) всего изображения и используется для вычисления центра масс объекта.

area представляет собой значение нулевого момента, которое соответствует площади объекта на изображении. Это значение используется для определения площади объекта.

5. Центроид (центр масс) объекта - точка, определяющая геометрический центр объекта. Она вычисляется как среднее арифметическое координат всех пикселей объекта:xc=∑i=1nxin,yc=∑i=1nyin*xc*​=*n*∑*i*=1*n*​*xi*​​,*yc*​=*n*∑*i*=1*n*​*yi*​​где $(x\_i, y\_i)$ - координаты $i$-го пикселя объекта, $n$ - число пикселей объекта.Центроид используется для определения местоположения объекта на изображении. Он может служить опорной точкой для дальнейшего анализа, например, для отслеживания движения объекта, вычисления расстояния до него и т.д.

**Листинг программы**

from fileinput import filename  
from sys import flags  
import numpy as np  
import cv2  
  
  
# Задание 1  
web\_cap = cv2.VideoCapture("http://192.168.1.68:8080/video")  
  
while True:  
 ret, frame = web\_cap.read()  
 if not ret:  
 print("Не удалось получить кадр")  
 break  
 frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)  
 cv2.imshow('Rec', frame)  
 if cv2.waitKey(1) & 0xFF == 27:  
 break  
  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
  
  
#Задание 2  
web\_cap = cv2.VideoCapture("http://192.168.1.68:8080/video")  
  
while True:  
 ret, frame = web\_cap.read()  
 if not ret:  
 print("Не удалось получить кадр")  
 break  
 hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)  
 red\_mask = cv2.inRange(hsv, (0, 100, 100), (30, 255, 255))  
 red\_only = cv2.bitwise\_and(frame, frame, mask=red\_mask)  
 cv2.imshow('Rec', red\_only)  
 if cv2.waitKey(1) & 0xFF == 27:  
 break  
  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
  
  
#Задание 3,4,5  
web\_cap = cv2.VideoCapture(r'http://192.168.1.68:8080/video')  
  
while True:  
 ret, frame = web\_cap.read()  
 if not ret:  
 print("Не удалось получить кадр")  
 break  
  
 hsv\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)  
 red\_frame = cv2.inRange(hsv\_frame, (0, 100, 100), (30, 255, 255))  
  
 kernel = np.ones((5, 5), np.uint8)  
 dilate = cv2.dilate(red\_frame, kernel, iterations=1)  
 erode = cv2.erode(dilate, kernel, iterations=1)  
 cv2.imshow('Close', erode)  
  
 erode = cv2.erode(red\_frame, kernel, iterations=1)  
 dilate = cv2.dilate(erode, kernel, iterations=1)  
 cv2.imshow('Open', erode)  
  
  
 moment = cv2.moments(erode)  
 if (moment["m00"] != 0):  
 print(f"Площадь: {moment['m00']}")  
 print(f"Моменты 1 порядка: {moment['m01']}, {moment['m10']}")  
 xc = int(moment['m10'] / moment['m00'])  
 yc = int(moment['m01'] / moment['m00'])  
 (contours, \_) = cv2.findContours(erode.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)  
 for countour in contours:  
 (x, y, w, h) = cv2.boundingRect(countour)  
 cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 4)  
  
 cv2.imshow('Moments', frame)  
  
 if cv2.waitKey(1) & 0xFF == 27:  
 break  
  
cv2.waitKey(0)  
cv2.destroyAllWindows()