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 №5**

**Дисциплина: Алгоритмы цифровой обработки мультимедиа**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Р. Гаджаев

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Андрей Александрович Крамаренко

**Тема работы:** Выявление движения на видео.

**Ход работы:**

* начать чтение из файла, прочитать первый кадр, перевести в черно-белый цвет и применить размытие Гаусса;
* подготовить файл для записи;
* далее начать цикл, который завершиться по завершению файла, внутри этого цикла:
* скопировать старый кадр;
* прочитать новый кадр, перевести в черно-белый цвет, применить размытие Гаусса;
* если чтение неуспешно, остановить цикл;
* найти разницу между двумя кадрами в отдельный фрейм (frame\_diff) - cv2.absdiff;
* провести операцию двоичного разделения для фрейма (frame\_diff) - cv2.threshold;
* найти контуры объектов для фрейма (frame\_diff) - cv2.findContours;
* пройтись по контурам объектов для фрейма (frame\_diff) и найти контур площадью большей, чем наперед заданный параметр -cv2.contourArea;
* если такой контур найден, значит было движение, записать кадр в файл;
* отобразить видео.

Сначала подготовим кадр для дальнейшей обработки, а именно переведем в черно-белое представление и применим Гауссовское размытие. (рисунок 1)



Рисунок 1 – подготовка кадра к обработке

Далее запустим цикл, где будут происходить все преобразования. Вычислим разницу между искомым изображением и новым с помощью функции cv2.absdiff(). Бинаризируем её, используя функцию cv2.threshold(), превращая пиксели, превышающие порог delta\_tresh, в белый цвет, а остальные - в черные (сохраняем только пороговое значение). Дальше найдем контуры (cv2.findContours()), используя пороговое значение. Все эти манипуляции изображены на рисунке 2.



Рисунок 2 – Разница, бинаризация и нахождение контуров

И наконец, будем проходится по каждому контуру и сравнить его площадь с общей площадью кадра. Если она больше или равна значению min\_area, то записываем текущий кадр в выходное видео. Если же нет, то продолжаем проход.

Были проведены тесты, которые показали более оптимальные значения параметров для выбранного видео, а именно:

* Размер ядра = 11
* Стандартное отклонение = 70
* Пороговое значение = 60
* Минимальная площадь = 20

На рисунке 3 показан результат работы алгоритма при самых оптимальных значениях.



Рисунок 3 - Результат работы алгоритма при самых оптимальных значениях.

Также тестирование было проведено со значениями:

* ядра 3, отклонением 50, пороговым значением 60 и площадью 20 (рисунок 4).



Рисунок 4 - Результат работы алгоритма при значениях 1.

* ядра 3, отклонением 50, пороговым значением 20 и площадью 20 (рисунок 5).



Рисунок 5 - Результат работы алгоритма при значениях 2.

* ядра 3, отклонением 50, пороговым значением 60 и площадью 10 (рисунок 6).



Рисунок 6 - Результат работы алгоритма при значениях 3.