Санкт-Петербургский государственный университет
Направление: 01.03.02 Прикладная математика и информатика

ООП: Прикладная математика, фундаментальная информатика и программирование

ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

Тема работы: *автоматическое обнаружение мерцающих сцен в видеоряде*

Выполнил: Семенов К. Е. студент гр. 21.Б09
Научный руководитель: Коровкин М. В. доцент, к. ф.-м. н.

Санкт-Петербург
2023

Содержание

Введение....3
План работы:....4
Подготовительная часть....4
  Подбор критериев....4
  Выбор методов хранения и обработки видео....4
Описание алгоритма....6
Тестирование и оценка производительности....7
Вывод из проведенной работы....7
Ссылки и литература....8

Введение

Известно, что видеоряды с мерцающими последовательностями кадров способны вызывать приступы у людей, подверженных светочувствительной эпилепсии. Однако люди не страдающие от этого недуга также могут испытывать некоторый дискомфорт при просмотре роликов с таким содержанием.

Эпилепсия — довольно нечастое заболевание (<1% населения), а вариация при которой приступ происходит по причине мерцания света составляет 3-5% от всех случаев болезни. Несмотря на это, помимо помощи имеющим этот недуг, результаты работы могут пригодиться для большего количества людей, которые предпочитают видео с более плавным изменением контраста между кадрами или хотят обезопасить себя от заболевания светочувствительной эпилепсией, так как механизм развития этой болезни полностью не изучен.

Целью данной работы является программа, обрабатывающая в реальном времени видеофайл и сообщающая пользователю о наличии мерцающих сцен в видеофайле и/или их таймкоды/количество.

План работы:

1. Нахождение или синтез критериев для определения мерцающей последовательности
2. Ознакомление с библиотекой алгоритмов компьютерного зрения OpenCV
3. Реализация переборного алгоритма выявления мерцающей последовательности по выявленным критериям
4. Тестирование программы на нескольких видеофайлах и анализ производительности

Подготовительная часть

В отчете упор сделан на выделение критериев выявления мерцающих последовательностей, изучение методов обработки видеофайлов и реализацию первичного решения.

Подбор критериев

Международный союз электросвязи (ITU) разработал рекомендации (ITU-R) для проверки телепрограмм на индуцирование эпилептических приступов. Код рекомендации — ITU-R BT.1702-2.

В этой рекомендации определяются следующие понятия:

1. **Вспышка** — событие, при котором происходит изменение яркости изображения на >=20 нит, при условии, что более темный кадр имеет яркость <=160 нит
2. **"Опасная" последовательность вспышек**— набор кадров, для которых выполнены условия:
    1. площадь вспышек последовательности кадров составляет не менее 25% площади экрана
    2. происходит >3 вспышек (6 изменений яркости) в секунду

    Также отмечается, что если последовательность вспышек не удовлетворяет этим двум условиям, но длится >5 секунд, то также переходит в ранг "опасных"

Выбор методов хранения и обработки видео

Библиотека алгоритмов компьютерного зрения OpenCV предоставляет класс `cv::Mat`, являющийся контейнером для изображения. Работать с этим контейнером удобно, так как он является шаблонным и

динамическим. Массив, содержащий значения интенсивности каждого пикселя изображения (назовем *ресурсом*) реализован как массив из одного-трех элементов — матриц. Каждая из этих матриц называется *каналом изображения* и показывает интенсивность цвета каждого пикселя, согласно цветовой модели. Один канал для изображения в оттенках серого, а три канала для цветовой схемы, к примеру BGR (OpenCV использует эту схему для цветных изображений по умолчанию).

	Column 0	Column 1	Column ...	Column m
Row 0	0,0	0,1	...	0, m
Row 1	1,0	1,1	...	1, m
Row ...	...,0	...,1	...	..., m
Row n	n,0	n,1	n,...	n, m

Рисунок 1: одноканальный режим

	Column 0			Column 1			Column ...			Column m		
Row 0	0,0	0,0	0,0	0,1	0,1	0,1	...	...	...	0, m	0, m	0, m
Row 1	1,0	1,0	1,0	1,1	1,1	1,1	...	...	...	1, m	1, m	1, m
Row ...	...,0	...,0	...,0	...,1	...,1	...,1	...	...	...	..., m	..., m	..., m
Row n	n,0	n,0	n,0	n,1	n,1	n,1	n,...	n,...	n,...	n, m	n, m	n, m

Рисунок 2: трехканальный режим (BGR)

Матрицы могут иметь разные типы данных: знаковые или беззнаковые целые, знаковые или беззнаковые числа с плавающей точкой. Каждый тип нужен для своих форматов выходных файлов, в данной работе ограничимся `uint8`. Также `cv::Mat` использует в своей имплементации умные указатели, так что экземпляры этого класса могут использовать один ресурс без копирования, для копирования есть отдельный метод.

Чтение кадров видео реализовано в OpenCV с помощью класса `cv::VideoCapture`. Метод `read()` осуществляет захват, декодинг и возвращение в выходной массив следующего изображения.

Описание алгоритма

Шаг 0. Инициализация захвата видео.

Создаем экземпляр `cv::VideoCapture`, проверяем инициализацию экземпляра методом `isOpened()`.

K-я итерация цикла по кадрам:

Шаг 1. Захват k-го кадра из видео.

Проверяем флаг метода `read(frameCurr)`, если `true`, то продолжаем выполнение алгоритма, иначе передаем исключение и завершаем работу программы.

Шаг 2. Подсчет изменений яркости по пикселям.

В выбранных критериях вспышки рассматривается разница яркостей изображения. Изображение хранится изначально в трех каналах по цветовой схеме BGR. Чтобы получить яркость, можно воспользоваться конвертацией в цветовую схему YUV или формулой нахождения яркости (компоненты Y) этой цветовой модели

$$Y = \frac{1}{256}(65.738 \cdot R + 129.057 \cdot G + 25.064 \cdot B)$$

где $R \in [0,255], G \in [0,255], B \in [0,255]$ — интенсивности цветов пикселя

Заполняем матрицу $\mathbf{L_k}$ яркостей значениями Y для каждого пикселя. Создаем матрицу $\mathbf{L_{diff} = L_k - L_{k-1}}$. Разделяем матрицу $\mathbf{L_{diff}}$: создаем матрицы $\mathbf{L_+}$ и $\mathbf{L_-}$, состоящие из абсолютных значений $\mathbf{L_{diff}}$, так что $\mathbf{L_+}$ содержит положительные разности, $\mathbf{L_-}$ — отрицательные.

Шаг 3. Подсчет среднего изменения яркости по кадру.

Преобразуем $[m \times n]$ матрицы $\mathbf{L_+}$ и $\mathbf{L_-}$ в массивы $\mathbf{h_+}$ и $\mathbf{h_-}$ $[[1 \times m \cdot n]]$.

Сортируем в порядке убывания, и определяем, есть ли ненулевой элемент на индексе $\frac{1}{4} n \cdot m$.

Если так для *обоих* массивов, то считаем среднее у каждого и среднее изменение яркости `lumDiffCurr` будет наибольшим из средних значений по каждому массиву.

Если есть нулевой элемент на индексе $\frac{1}{4} n \cdot m$ *у одного* из массивов, то возвращаем среднее того, у которого элемент на этом индексе ненулевой.

Иначе возвращаем 0.

Шаг 4. Сравнение k-го и (k-1)-го средних значений яркости.

Считаем среднее `lumMeanCurr` яркости кадра. Если минимальная средняя яркость на (k-1)-м и k-м шагах (`lumMeanCurr` на k-м, `lumMeanPrev` на (k-1)-м шаге) <=160 нит и `lumDiffCurr` >=20 нит, то получена вспышка, в конец вектора индексов вспышек добавляем номер кадра.

Иначе получено приемлемое изменение яркости и переходим к (k+1)-му шагу.

Тестирование и оценка производительности.

№	Описание видео	$T_{видео}$, с	$n \times m$	FPS	$T_{прог}$, с	$N_{вспышек}$
1	Контраст не меняется, преобладают темные цвета	30	1920x1080	30	~123 с	0
2	Сильное мерцание на протяжении всего видео, преобладают яркие цвета	79	480x360	30	~31 с	1422
3	Выраженное короткое мерцание после половины видео, преобладают яркие цвета	31	480x360	25	~8 с	12
4	Выраженное мерцание в нескольких местах видео, преобладают темные цвета	12	1280x720	30	~20 с	93

Видео 1 (только скачивание), Видео 2 (youtube), Видео 3 (youtube), Видео 3 (youtube)

Вывод из проведенной работы

Простой переборный алгоритм, реализующий проверку критериев ITU-R BT.1702-2, показал, что эти критерии действительно определяют, есть ли мерцание в видео, а формула перехода к компоненте Luma (Y) в модели YUV подходит для подсчета яркости в этом случае. То есть проблему поиска критериев мерцания в видео можно считать решенной.

Однако производительность такого алгоритма оставляет желать лучшего при анализе видео в HD разрешении и выше. Дальнейшие шаги по улучшению алгоритма могут включать в себя уменьшение области поиска на кадре с помощью функционала OpenCV, такого как установление пороговых значений цветов (выделение светлых областей и запуск алгоритма там), векторизации SIMD-инструкциями (параллелизм на уровне команд) и параллелизма на уровне задач.

Ссылки и литература

- Документация OpenCV: https://docs.opencv.org/4.x/
- «Guidance for the reduction of photosensitive epileptic seizures caused by television», ITU-R BT.1702-3, 2023
- Статья «Automatic detection of flashing video content», Lúcia Carreira, Nelson Rodrigues, Bruno Roque, Maria Paula Queluz, 2015