# ПОСТ-ОБРАБОТКА ВИДЕО С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА GSTREAMER

Курсовой проект по дисциплине

«Компьютерная графика»

Студент гр. 429-3

А.А. Бабец

« » 2022 г.

Руководитель

доцент кафедры АОИ, канд. техн. наук, доцент

Т.О. Перемитина

«\_ » 2022 г.

Томск 2022

## ЗАДАНИЕ

на курсовой проект по дисциплине «Компьютерная графика»

студенту Бабец Алексею Алексеевичу

группа 429-3, факультет систем управления

1. Тема работы: пост-обработка видео с использованием фреймворка GStreamer
2. Срок сдачи студентом законченного проекта: *4 июня 2022 г.*
3. Исходные данные к проекту: Разработать программу, осуществляющую возможность изменения разрешения, соотношения сторон и частоты кадров видео.
4. Содержание пояснительной записки: введение, анализ задачи, теоретические вопросы решаемой задачи, обзор аналогов, алгоритм программы, тестирование, руководство пользователя, заключение.
5. Дата выдачи задания: *19 марта 2022 г.*

Руководитель:

доцент кафедры АОИ, канд. техн. наук,

доцент, Перемитина Т.О.

Задание принял к исполнению 19 марта 2022 г.

(дата)

(подпись студента)

# Введение

Актуальность выбранной темы, а именно пост-обработка видео неоспорима, так как давно существуют различные видеосервисы, онлайн-кинотеатры, на которых можно смотреть видео в разном разрешении, качестве и соотношении сторон, подбирая в зависимости от скорости своего Интернет-соединения и монитора.

Для данного проекта выбрано изменение таких характеристик видео, как соотношение сторон, разрешение и частота кадров. Создание данного приложения может послужить полезным подспорьем при создании видеосервиса, а также очень полезный опыт в обработке видео, а также способ улучшить навыки программирования для разработчика.

Для разработки видео обработчика были поставлены следующие задачи:

* анализ предметной области и проектирование приложения;
* выбор инструментального программного обеспечения;
* изучение выбранных средств разработки;
* разработка приложения с использованием фреймворка GStreamer.

Таким образом, была обозначена актуальность выбранной темы и поставлены задачи.

3

# Анализ предметной области

Как говорилось ранее, задача – пост-обработка видео. Так как эта тема довольно обширная, решено было выбрать изменение таких характеристик, как соотношение сторон, разрешение и частота кадров. Выбор обусловлен тем, что во многих видеосервисах присутствует возможность выбрать эти характеристики видео при просмотре, следовательно, при загрузке на сервер, видео обрабатывается для придания соответствующих характеристик.

**Разрешение** — характеристика изображения, показывающая размер по горизонтали и вертикали в пикселях. Самыми распространёнными на данный момент являются разрешения при соотношении сторон 16:9, такие как SD (Standart Definition, 480p, 640x480), HD (High Definition, 720p, 1280x720), Full HD (Full High Definition, 1080p, 1920x1080), менее распространёнными – Quad HD (2k 2048x1080) и Ultra HD (4k, 3840x2160 и 8k, 7680x4320) [1].

**Частота кадров в секунду (англ. frame rate) –** характеристика, показывающая количество кадров, сменяющих друг друга за секунду. Единица измерения данной характеристики – кадры в секунду (англ. frames per second, или FPS). Чем их больше, тем плавнее изображение. 24 FPS – стандартная частота кадров в кинематографе, наиболее комфортная для зрителя, однако есть фильмы, снятые в 48 и даже 120 FPS [2].

# Обзор аналогов и выбор инструментов

## IDE Visual Studio

**Visual Studio –** полнофункциональная среда разработки от компании Microsoft, позволяющая работать с платформами Windows, Интернет и Android. Возможности данной IDE позволяют правильно и эффективно писать код, реорганизовывать, анализировать и исправлять проблемы с кодом [3].

Также Visual Studio имеет большой набор компонентов для создания графического интерфейса.

## IDE Qt Creator

**Qt Creator** – удобная и быстрая среда разработки C++. Данная кроссплатформенная среда имеет удобный современный редактор кода C++, встроенный удобный графический интерфейс, дизайнер форм, инструменты навигации и множество других полезных функций. Множество разработчиков выбирают его за API и библиотеки, поскольку они последовательны, подробны, удобны и хорошо документированы.

Данная среда разработки поставляется с полным набором инструментов разработчика для одновременного создания приложений и пользовательских интерфейсов, а также для дальнейшего их развёртывания в различных мобильных и настольных ОС [3].

## IDE CLion

**CLion** – мощная и кроссплатформенная IDE для C и C++ от компании JetBrains, включающая в себя современные стандарты C++, libs++ и Boost. Также она хорошо знает коды и упрощает рутину, позволяя сосредотачиваться на основных вещах разработки. Одной из основных возможностей этой среды разработки является рефакторинг, позволяющий переименовывать символы, сдвигать элементы вверх и вниз по ранжированию, изменять сигнатуру функции, а также гарантировать, что её автоматический рефакторинг будет правильно генерировать необходимые изменения в коде.

Интегрированный отладчик анализирует и решает проблемы с помощью дружественного пользовательского интерфейса IDE. Более того, он также удалённо связывается с нативными процессами или отладками [3].

## Формирование критериев сравнения

Для определения оптимального инструмента разработки приложения на C++ с использованием фреймворка GStreamer можно выделить следующие критерии:

1. Наличие подсветки синтаксиса;
2. Наличие инструментов для создания графического интерфейса;
3. Наличие документации;
4. Кроссплатформенность;
5. Возможность подключения дополнительных фреймворков.

## Сравнение аналогов и выбор среды разработки

Опираясь на техническую информацию, находящуюся в свободном доступе, а также на проводившиеся сравнения технических характеристик выбранных сред разработки, проведён анализ для определения оптимального и удобного инструмента. Результат приведён в таблице 2.1.

Таблица 2.1 – Сравнение аналогичных сред разработки

|  |  |  |  |
| --- | --- | --- | --- |
|  | Visual Studio | Qt Creator | CLion |
| Наличие подсветки синтаксиса | + | + | + |
| Наличие инструментов для создания графического интерфейса | + | + | - |
| Наличие документации | + | + | + |
| Кроссплатформенность | - | + | + |
| Возможность подключения дополнительных фреймворков | + | + | + |

По результатам проведённого анализа видно, что Qt Creator имеет преимущество перед другими представленными средами разработки и будет более удобен в использовании.

# Разработка эскизного проекта

# Логика программы

Программа будет использоваться для изменения параметров видео, имеющих числовые значения, поэтому необходимо разработать соответствующий интерфейс и классы, для работы с видео с фреймворком.

Основной алгоритм работы программы:

Шаг 1: выбор видеофайла и его анализ.

Шаг 2: установка необходимых значений параметров.

Шаг 3: обработка и сохранение видео.

Шаг 4: выбор другого видеофайла или выход из программы.

# Диаграмма классов

Для работы программы необходимо разработать 2 класса: класс для работы с самим фреймворком, класс для хранения информации о видео.

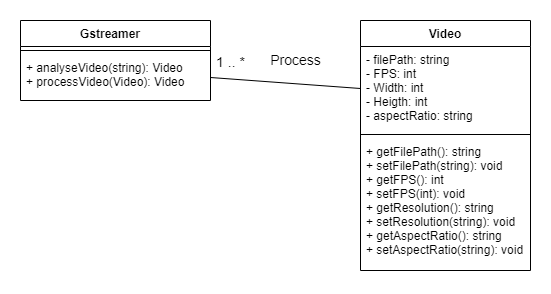


Рисунок 3.1 – Диаграмма классов

# Макет интерфейса

Далее был разработан макет интерфейса программы, состоящий из двух блоков: блок выбора видеофайла, в котором будет отображаться информация о нём, блок выбора значений, применяемых при обработке.

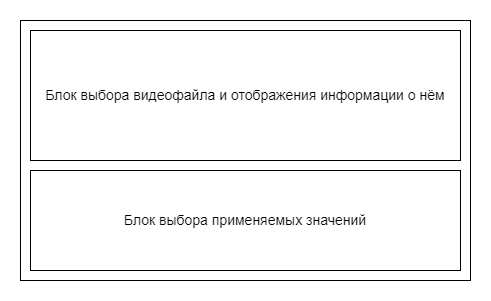


Рисунок 3.2 – Макет интерфейса программы

# Алгоритм обработки видео

# При использовании фреймворка GStreamer для обработки видео используются компоненты, собираемые в необходимой последовательности в структуру, называемую пайплайн.

# Первым активным элементом в данной структуре является GstQTDemux – модуль, разделяющий видеопоток на видео и аудиодорожки.

# 

# Рисунок 4.1 – Первый элемент пайплайн

# Далее следует GstQueue – очередь, обеспечивающая правильную передачу данных далее. Затем следует GstH264Parse - вспомогательный элемент для декодирования H264 формата видео. После чего следует сам декодер этого формата – avdec\_h264. После чего данные передаются в GstVideoConvert – видео конвертер, который, с помощью элементов, установленных далее конвертирует видео к необходимым характеристикам.

# 

# Рисунок 4.2 – Продолжение пайплайна

# Затем присоединяется элемент для изменения соотношения сторон пикселя – GstAspectRatioCrop.

# 

# Рисунок 4.5 – Продолжение пайплайна

# Далее видеоданные попадают в элемент GstVideoScale, используемый для изменения соотношения и размера сторон картинки. Параметры для изменеия этим элементом задаются с помощью элемента GstCapsSetter, следующего далее.

# 

# Рисунок 4.6 – Продолжение пайплайна

# Затем видеоданные попадают в элемент GstVideoRate, который используется для изменения частоты кадров. Параметры изменения этим элементом также задаются с помощью GstCapSetter.

# 

# Рисунок 4.7 – Продолжение пайплайна

# Далее следуют последние элементы пайплайна:

# Gst\_NvH264Enc – енкодер, кодирующий видеоданные в формат H264;

# GstH264Parse – вспомогательный элемент для енкодера;

# GstQueue – очередь для предотвращения потери данных в случаях, когда последующий элемент не успевает всё обрабатывать до поступления следующих данных;

# GstAviMux – муксер, собирающий аудио- и видеодорожки вместе.

# 

# Рисунок 4.8 – Последние элементы пайплайна

# Таким образом мы получили структуру, при прохождении которой видео приобретает необходимые нам параметры.

# Программирование и отладка.

# Интерфейс

# На рисунке 5.1 показан интерфейс программы, состоящий из двух блоков.

# Первый блок содержит кнопку, открывающую диалоговое окно для выбора файла, элемент, показывающий путь к выбранному файлу, под которым находится блок элементов, показывающий текущие характеристики выбранного видеофайла:

# Heigth – высота картинки в пикселях;

# Width – ширина картинки в пикселях;

# FPS – количество отображаемых кадров в секунду;

# Aspect Ratio – соотношение сторон пикселя.

# Далее находится блок для выбора применяемых к видео характеристик:

# Aspect Ratio – соотношение сторон видео;

# Width – ширина картинки в пикселях;

# Heigth – высота картинки в пикселях;

# FPS – количество отображаемых кадров в секунду.

# И кнопка «Process» для запуска процесса обработки.

# 

# Рисунок 5.1 – Интерфейс программы

## Логика работы

# При нажатии на кнопку «Select File» открывается диалоговое окно для выбора файла, после выбора происходит анализ этого файла с помощью утилиты gst-discoverer-1.0, после чего результаты отображаются в программе.

# Далее необходимо выбрать нужные значения характеристик и нажать на кнопку «Process».

# Тестирование и отладка

Для выполнения отладки в процессе разработки были использованы средства отладки, интегрированные в среду разработки QTCreator, а также средства отладки фреймворка GStreamer. Средства отладки среды разработки позволяют останавливать выполнение программы в любой момент и смотреть текущие значения всех задействованных переменных. Средства отладки фреймворка позволяют выводит граф пайплайна на промежуточных этапах работы программы, что позволяет отследить правильность сборки и работы этой структуры. Благодаря вышеперечисленным средствам отладки были исправлены ошибки как самой программы, так и средств фреймворка.

После завершения разработки были проведены необходимые тесты программы. Основной задачей тестирования было исследование работы программы при различных изменениях характеристик видео. Таким образом было выяснено, что изменение характеристик в сторону увеличения вызывает более длительную работу программы при больших изменениях. Уменьшение или небольшие изменения характеристик требуют не так много времени.

## Результат работы

Результатом работы стала программа, выполняющая изменение таких характеристик видео как соотношение сторон, высота и ширина (в пикселях), количество кадров в секунду.

На рисунках 6.1 и 6.2 представлены характеристики видео до и после изменения соответственно. Были изменены соотношение сторон, высота и ширина кадра, количество кадров в секунду. Для проверки характеристик была использована утилита фреймворка GStreamer под названием gst-discoverer-1.0.

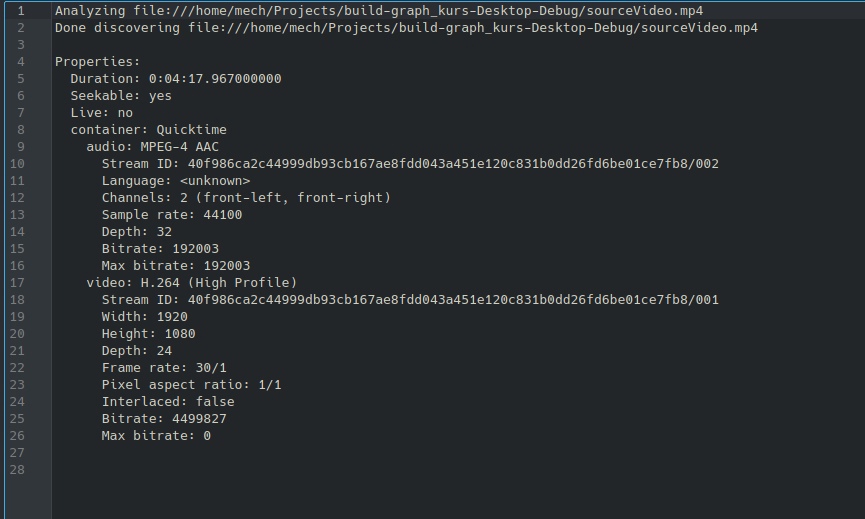


Рисунок 6.1 – Характеристики видео до изменения

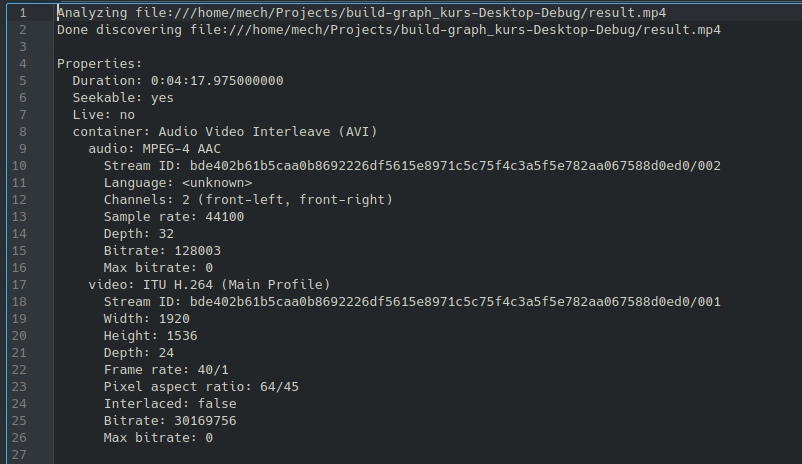


Рисунок 6.2 – Характеристики видео после изменения

# Заключение

GStreamer позволяет обрабатывать различные форматы видеофайлов и видеопотоков, с помощью определённых наборов компонентов структуры пайплайн можно добиться практически любого результата обработки видеоданных, начиная от простого перекодирования для изменения формата, заканчивая изменением практически любых характеристик видеоданных. Это может быть применено, например, в сервисе показа онлайн телеканалов для приведения видеопотока в более компактный формат с необходимыми характеристиками.

Исследование предметной области дало опыт и улучшило навыки научно-исследовательской работы, анализа, систематизации материала. В процессе разработки программы был изучен используемый фреймворк, улучшены навыки создания алгоритмов, интерфейса программы. Также закреплены навыки самостоятельной работы над проектом и оформления пояснительной записки.

# Список литературы

1. Sravnismart.ru портал о смартфонах [электронный ресурс] – Режим доступа: <https://sravnismart.ru/video_recording/> (дата обращения 28.03.2022)
2. Skillbox Media – журнал про бизнес, дизайн, разработку игр и программирование [электронный ресурс] – Режим доступа: <https://skillbox.ru/media/gamedev/chto_takoe_chastota_kadrov_v_sekundu_24_30_i_60_fps/> (дата обращения 28.03.2022)
3. ITVDN Видеокурсы по программированию [электронный ресурс] – Режим доступа: <https://itvdn.com/ru/blog/article/cplspls-top7> (дата обращения: 08.04.2022)