МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

ИМЕНИ АКАДЕМИКА Д.Н. ПРЯНИШНИКОВА»

Факультет экономики и информационных технологий

Кафедра информационных технологий и программной инженерии

Направление подготовки 09.04.03 Прикладная информатика

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\* \*\* \*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*

Автор работы \_\_\_\_\_\_\_\_\_\_\_\_\_ \*.\*. \*\*\*\*\*\*\*\*

Группа ПИм-21 (подпись, дата)

Научный руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*.\* \*\*\*\*\*\*\*\*\*\*

(подпись, дата)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*.\*. \*\*\*\*\*\*\*\*\*

(подпись, дата)

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Муратова

(подпись, дата)

Пермь-20\_\_ г.

Министерство сельского хозяйства Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский государственный аграрно-технологический университет

имени академика Д.Н. Прянишникова»

**Факультет экономики и информационных технологий**

Кафедра информационных технологий и

программной инженерии

«УТВЕРЖДАЮ»

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_/Муратова Е.А./

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

Фамилия Имя

Отчество

Группа Пим-21 Направление Прикладная информатика 09.04.03

Научный руководитель  Беляков Андрей Юрьевич, кандидат технических наук, доцент, доцент кафедры информационных технологий и программной инженерии,

Рецензент(ы)

Срок выполнения диссертации:  с  \_\_.\_\_.\_\_ . по \_\_.\_\_.\_\_ ..

**1.** **Тема выпускной квалификационной работы**

Тема утверждена приказом по ПГАТУ № от « » 20 г.

**2. Содержание и объем работы, какие должны быть выполнены графические работы, расчеты, список рекомендуемой литературы**.

1. Проанализировать предметную область.
2. Подтвердить актуальность и определить новизну исследования.
3. Определить цель и задачи исследовательской работы.
4. Определить используемые методики исследования.
5. Провести анализ особенностей систем видеокомпресии.
6. Проанализировать существующие алгоритмы индексации видеофайлов.
7. Разработать алгоритм создания уникальных подписей ключевых кадров видеопоследовательности.
8. Разработать диаграмму потоков данных DFD информационной системы.
9. Разработать информационно-логическую модель базы данных и обосновать выбор базы данных.
10. Создать объекты базы данных, заполнить справочники, создать базовые запросы.
11. Обосновать выбор технологий, форматов хранения данных, сред(ы) разработки и используемых языков программирования.
12. Выполнить программную реализацию предложенной информационной системы.
13. Провести анализ эффективности алгоритма.
14. Оценить возможности реализованного приложения в задачах поиска видеофрагментов.
15. Разработать демонстрационный материал к ВКР, подготовить доклад.

**Рекомендованная литература:**

1. Выпускная квалификационная работа по направлению подготовки 09.03.03 Прикладная информатика: методические рекомендации /сост. А.Н. Козлов; М-во с.-х. РФ, «Пермский гос. аграрно-технологич. ун-т им. акад. Д.И. Прянишникова». – Пермь: ИПЦ «Прокростъ», 2019. – 46 с.
2. Белов, В.В., Чистякова В.И. Проектирование информационных систем: учебник для студ. учреждений высш. проф. образования – М.: Издательский центр «Академия», 2017. – 352с.
3. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – М.: Диалог-МИФИ, 2013. – 432 с.

**3. План выполнения выпускной квалификационной работы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Этапы выполнения диссертации | % этапа | Сроки | % выполнения | Подпись руководителя или консультанта |
| 1. Получение задания на выполнение диссертации | 5% | 10.06.\_\_ |  |  |
| 1. Анализ материалов и разработка структуры (содержания) диссертации | 5 % | 15.06.\_\_ |  |  |
| 1. Разработка и предоставление на проверку первого раздела | 20 % | 18.06.\_\_ |  |  |
| 1. Разработка и предоставление на проверку второго раздела | 30 % | 25.06.\_\_ |  |  |
| 1. Разработка и предоставление на проверку третьего раздела | 15 % | 04.07.\_\_ |  |  |
| 1. Оформление диссертации (введение, заключение, список литературы). Доработка диссертации в соответствии с замечаниями и предоставление ее на кафедру | 5 % | 05.07.\_\_ |  |  |
| 1. Проверка в системе "Антиплагиат" | 5% | 06.07.\_\_ |  |  |
| 1. Подготовка презентации и доклада. | 10% | 07.07.\_\_ |  |  |
| 1. Предзащита | 5 % | 08.07.\_\_ |  |  |

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Беляков А.Ю./

*Задание принял к исполнению* «\_\_\_\_» \_\_\_\_\_\_\_ 20\_\_ г.

Обучающийся *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* / /

(подпись)

Допустить к защите ВКР в государственной экзаменационной комиссии.

Протокол заседания кафедры № \_\_\_\_ от «\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_г.

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Муратова Е.А./

Защиту назначить на \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Декан факультета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Тупицына О. В./

Реферат

С. 113, рис. 25, табл. 6, лит. 24 источника

ИНФОРМАЦИОННАЯ СИСТЕМА, CBVIR, ИНДЕКСАЦИЯ, ОБРАБОТКА ВИДЕОФАЙЛОВ, БАЗА ДАННЫХ, DFD, ИНДЕКСАЦИЯ ВИДЕОФАЙЛОВ, ПОИСК ФАЙЛОВ

Разработан алгоритм создания уникальных подписей ключевых кадров видеопоследовательности и реализовано веб-приложение для задач поиска заимствования видеопотока. Сформулированы цель и задачи исследовательской работы. Найдена научная проблем и подтверждена актуальность исследовательской работы.

Дано описание существующих алгоритмов создания уникальных подписей для ключевых кадров, обоснован выбор методов индексации.

Построены модели «как будет» информационной системы в нотации DFD. Описаны средства для реализации информационной системы. Осуществлена реализация информационной системы с использованием средства разработки программного обеспечения «Node.js».

Выполнено тестирование информационной системы. Оценена эффективность алгоритма.

**Содержание**

[Введение 7](#_Toc108698018)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 9](#_Toc108698019)

[1.1. Анализ предметной области исследования 9](#_Toc108698020)

[1.2. Подтверждение актуальности и определение новизны исследования 12](#_Toc108698021)

[1.3. Определение цели исследовательской работы 16](#_Toc108698022)

[1.4. Используемые методики исследования 17](#_Toc108698023)

[Выводы по разделу 19](#_Toc108698024)

[2. РАЗРАБОТКА АЛГОРИТМА СОЗДАНИЯ УНИКАЛЬНЫХ ПОДПИСЕЙ КЛЮЧЕВЫХ КАДРОВ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ 20](#_Toc108698025)

[2.1. Анализ особенностей систем видеокомпрессии 20](#_Toc108698026)

[2.2. Алгоритмы обработки ключевых кадров видеопотока для оценки их схожести 26](#_Toc108698027)

[2.2.1. Метод цветовых гистограмм 26](#_Toc108698028)

[2.2.2. Метод матриц смежности 28](#_Toc108698029)

[2.2.3. Метод перцептивных хешей 30](#_Toc108698030)

[2.3. Выбор метода обработки пространственных характеристик ключевых кадров 31](#_Toc108698031)

[2.4. Разработка алгоритма создания уникальных подписей ключевых кадров видеопоследовательности 31](#_Toc108698032)

[2.5. Проектирование информационной системы с реализацией алгоритма создания уникальных подписей ключевых кадров видеопоследовательности 36](#_Toc108698033)

[2.5.1. Разработка модели TO-BE потоков данных DFD приложения 36](#_Toc108698034)

[2.5.2. Разработка базы данных информационной системы 38](#_Toc108698035)

[2.5.3. Обоснование выбора системы управления базами данных 39](#_Toc108698036)

[2.5.4. Разработка физической модели базы данных 43](#_Toc108698037)

[2.5.1. Критерии оценивания и показатели их вычисления 45](#_Toc108698038)

[Выводы по разделу 46](#_Toc108698039)

[3. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ И ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМА 47](#_Toc108698040)

[3.1. Обоснование выбора средств программной реализации ИС 47](#_Toc108698041)

[3.2. Описание программной реализации 49](#_Toc108698042)

[3.3. Описание процесса тестирования и сбора доказательной базы 53](#_Toc108698043)

[3.4. Интерпретация результатов и оценка эффективности 55](#_Toc108698044)

[Выводы по разделу 57](#_Toc108698045)

[Заключение 58](#_Toc108698046)

[Список использованных источников 59](#_Toc108698047)

[Приложение А 62](#_Toc108698048)

[Приложение Б 79](#_Toc108698049)

**Перечень условных обозначений, терминов и сокращений**

ИС – информационная система.

СУБД – система управления базами данных.

БД – база данных.

CBVIR – Content-based visual information retrieval (поиск визуальной информации на основе контента).

MPEG – Moving Picture Experts Group (Экспертная группа по движущимся изображениям).

ISO – International Organization for Standardization (Международная организация по стандартизации).

IEC – International Electrotechnical Commission (Международная электротехническая комиссия).

# **Введение**

В современном мире информационные технологии играют всё большую роль. Их область распространения затрагивает новые сферы человеческой жизни, что позволяет ставить и решать новые задачи. Широкое распространение гаджетов позволяет людям снимать огромное количество событий с самых разных ракурсов, увеличивая базу видеоконтента с невиданной скоростью. Благодаря столь стремительному росту появляются новые задачи, решение которых может внести очередной вклад в упрощение жизни человека. Происходит переход от простых задач сравнения видеофайлов к более сложным задачам распознавания каких-то конкретных образов и обработки видеопотоков.

Особый интерес представляет область обработки видеопотоков. Видеофайлы имеют некую двойственную структуру. С одной стороны, это лишь последовательность изображений, и для решения многих задач можно использовать алгоритмы, которые применяются к обычным изображениям. В то же время, видеопотоки имеют некоторые отличительные особенности, например, связность, которые позволяют решать задачи с помощью эффективных алгоритмов, основывающихся на этих особенностях. Индексирование в данной сфере подразумевает под собой описательную характеристику кадра – индекс.

Главный принцип использования индексирования видеопотоков заключается в том, что результат этого процесса может многократно использоваться после окончания программы. В результате обработки видеопотока каждый кадр получает свой индекс, который можно сохранить в некотором файле, и при запуске новой задачи обработки видеопотока, например, поиске определенного кадра, использовать имеющиеся индексы для быстрого решения задачи.

Объект исследования – процессы обработки и анализа видеопотоков в мультимедиа информационных технологиях.

Предмет исследования – методы создания уникальных подписей ключевых кадров для индексации и поиска в видеоколлекциях.

Положения, выносимые на защиту:

* В ходе исследования выявлено, что комбинированный метод создания уникальных подписей ключевых кадров видеопоследовательности на основе темпоральных и пространственных характеристиках позволяет эффективно решать задачу поиска заимствований видеофрагментов.

Практическая значимость состоит в разработке алгоритма сегментации видеопотоков, что создает предпосылки для многоаспектной индексации в задачах информационного поиска и обеспечивает повышение быстродействия поиска в системах CBVIR с запросами «по образцу».

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ
   1. Анализ предметной области исследования

В 2021 году российский рынок легальных онлайн-сервисов, предлагающих для просмотра профессиональный видеоконтент, вновь продемонстрировал высокие темпы роста [1]. Причинами роста, как сообщает «ТМТ Консалтинг», стали развитие цифровых экосистем, рост числа пользователей мультисервисных подписок, увеличение онлайн-кинотеатрами инвестиций в производство собственного кино и сериального контента, продолжившееся в 2021 году влияние пандемии на посещаемость кинотеатров и рост интереса к домашнему видеопросмотру. В то же время негативное влияние на рынок все еще продолжает оказывать достаточно высокий уровень видеопиратства.

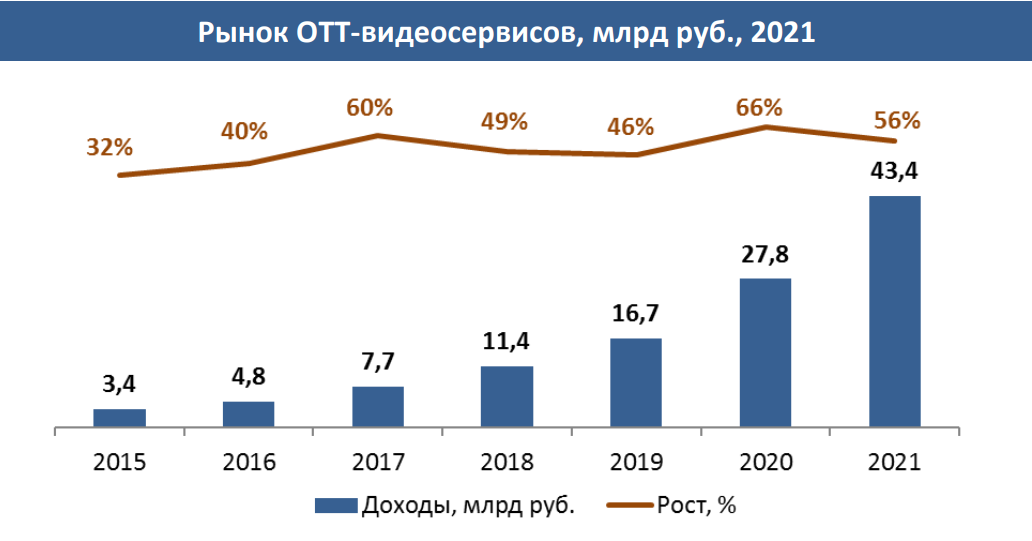


Рисунок 1.1 – Рынок видеосервисов в России (ТМТ Консалтинг)

За первый квартал 2022 года компания MUSO зафиксировал 52,5 миллиарда посещений пиратских веб-сайтов в первом квартале 2022 года, что на 29,3% больше по сравнению с первым кварталом 2021 года, при этом заметное увеличение в каждом из пяти медиа-секторов [2]. Рост пиратства телевизионного контента и фильмов составили в сумме 61,7 процента, по сравнению с первым кварталом 2021 года. В разбивке по отраслям 47,7% всех пиратских посещений приходится на телевизионный контент и 12.2% на фильмы (рисунок 1.2).

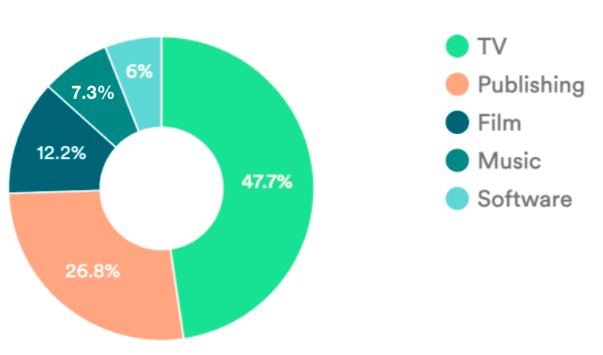


Рисунок 1.2 – Рейтинг пиратских сервисов по секторам (MUSO)

В рейтинге MUSO Соединенные Штаты демонстрируют самый высокий глобальный спрос на пиратство в первом квартале 2022 года. На втором месте находится Россия (таблица 1.1) с более чем 3 миллиардами посещений пиратских ресурсов.

Таблица 1.1 – Рейтинг стран по пиратству (по версии компании MUSO)

|  |  |  |
| --- | --- | --- |
| **Рейтинг** | **Страна** | **Посещений** |
|  | США | 5,662,541,831 |
|  | Российская федерация | 3,030,468,120 |
|  | Индия | 3,012,491,886 |
|  | Китай | 1,781,912,526 |
|  | Франция | 1,661,176,950 |
|  | Бразилия | 1,535,220,650 |
|  | Канада | 1,426,969,459 |
|  | Япония | 1,407,387,558 |
|  | Великобритания | 1,393,668,469 |
|  | Индонезия | 1,354,359,755 |
|  | Турция | 1,316,025,675 |
|  | Мексика | 1,310,955,577 |
|  | Вьетнам | 1,202,772,368 |
|  | Украина | 1,178,327,540 |
|  | Германия | 1,126,089,923 |

Экспоненциального рост приложений для социальных сетей и веб-сайтов для обмена видео, которые содержат огромное количество контента вызывают все больший интерес к средствам поиска и индексации видеофайлов последние годы. Отсутствие эффективных алгоритмов индексации является серьезной проблемой для правообладателей и введет к увеличению пиратского контента в сети. Проблема с видеофайлами усугубляется из-за его значительно большего объема, по сравнению с текстом и изображениями, что делает его большой проблемой для каждой видеоплатформы, а также для систем, которые анализируют и индексируют большие объемы веб-видеоконтента.

Таким образом, технологии, используемые для реализации функций индексирования и поиска видеофайлов, всё еще недостаточно развиты и не удовлетворяют требованиям большинства практических задач. Использование средств для индексации и сопоставления видеофайлов позволит эффективно бороться с нарушением авторских прав.

* 1. Подтверждение актуальности и определение новизны исследования

1) «Алгоритм поиска дубликатов в базе видеопоследовательностей на основе сопоставления иерархии смен сцен», Паршин, А. Е., Глазистов И. В. [3]

В статье приводится алгоритм для поиска дубликатов видеофрагментов. Он основывается на сопоставлении дерева сцен видеофрагмента-запроса и фильма в базе видео. Алгоритм состоит из трех основных этапов: разбиение видеопоследовательности на сцены, построение дерева сцен, сравнение деревьев сцен. Достоинствами метода являются устойчивость к большинству приемов создания искусственных дубликатов (т.к. используется только временная информация), малая сложность сравнения двух клипов и возможность создания иерархических индексов фильмов. Это позволяет отлавливать несовпадения, на начальных этапах проверки. Недостатком является то, что не учитываются характеристики самих сцен.

2) «Поиск нечетких дубликатов видео», Никитин И. К. [4]

В статье рассмотрена комбинация различных методов индексирования видеофайла. Видео рассматривается как последовательность фактов, развивающихся во времени. Авторы используют относительные длины сцен к длинам соседних сцен, а также внутренние характеристики сцен – характеристики начального и конечного кадров. Для сопоставления предлагается применять бинарные подписи. Авторы утверждают, что метод справляется с поиском в базе из тысячи видеофильмов за приемлемое время (менее секунды).

3) «Robust voting algorithm based on labels of behavior for video copy detection», Julien Law-To, Olivier Buisson, [Valerie Gouet-Brunet](https://www.researchgate.net/profile/Valerie-Gouet-Brunet), [Nozha Boujemaa](https://www.researchgate.net/profile/Nozha-Boujemaa). [5]

В работе выделяют особые точки кадра (с помощью детектора Харриса) и отслеживают их положение на протяжении всего видео. После чего формируют множество траекторий точек. Сопоставление с образцом происходит на основе нечеткого поиска. Подход облегчает локализацию нечетких дубликатов фрагментов. Однако, метод дорог из-за выделения особых точек кадров. А факт того, что траектории точек чувствительны к движению камеры, делают алгоритм применимым только для поиска точных копий видео.



Рисунок 1.3 – Пример работы метода с отслеживанием особых точек кадра

4) «Non-identical duplicate video detection using the SIFT method», Karthikeyan Vaiapury, Pradeep K. Atrey, Mohan S. Kankanhalli , Kalpathi Ramakrishnan [6]

Авторы выделяют особые точки ключевых кадров, и оценивают подобие кадров на основе масштабно-инвариантной трансформация признаков. Подобие кадров вычисляют как среднее арифметическое количества совпавших особых точек. Но для определения сходства видео, используется полная оценка соответствия как среднее значение подобия ключевых кадров по всему видео. Важно, что среднее значение вычисляется, не по всем возможным парам кадров, а только по некоторым из них. Это позволяет экономить вычислительные ресурсы.

5) «Inria-Learar's video copy detection system», Matthijs Douze, Adrien Gaidon, Herve Jegou, Marcin Marszałek, Cordelia Schmid. [7]

В данной статье используется метод, использующие визуальные слова, являются улучшенной версией прямого сравнения особых точек кадров. В их основе лежит квантование особых точек — формирование «слов». Сравнение кадров (и видео) целиком происходит по частотным словарям, как для текстов. Ключевые кадры представлены особенностями, которые получены с помощью масштабно-инвариантной трансформации признаков. Затем эти характеристики квантуются в визуальные слова. Из визуальных слов строится бинарная подпись. Но, для применения визуальных слов, должны быть построены частотные словари для заранее известной предметной области. Метод демонстрирует высокую производительность.

6) «Efficiently Matching Sets of Features with Random Histograms», Wei Dong, Zhe Wang, Moses Charikar, Kai Li. [8]

В работе авторы применяют локально-чувствительное хеширование. Его используют для отображения цветовой гистограммы каждого ключевого кадра на бинарный вектор. Характеристики кадра выделяются из локальных особенностей изображения. Эти характеристики представляются как множества точек в пространстве характеристик. При помощи локально-чувствительного хеширования точки отображаются на дискретные значения. По набору особенностей строится гистограмма кадра. Далее гистограммы сравниваются как обычные последовательности. Данный метод страдает от потенциальной проблемы большого потребления памяти. Временная информация никак не учитывается.

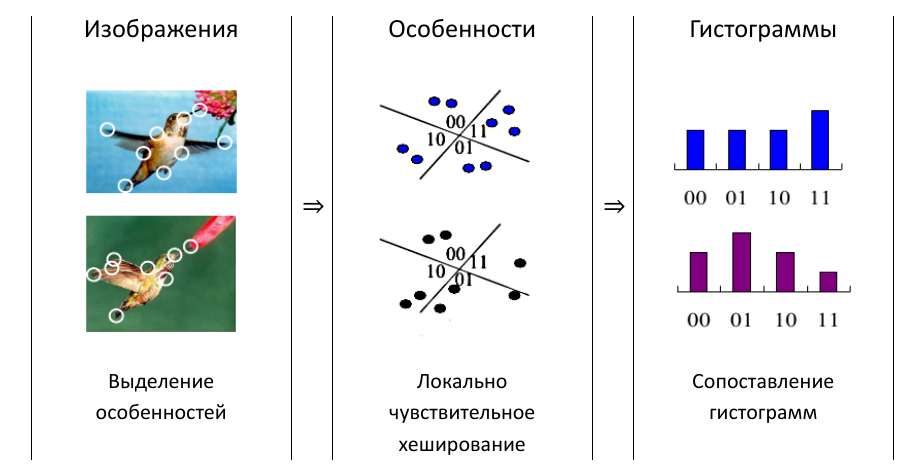


Рисунок 1.4 – Пример работы метода с локально-чувствительным хешированием

7) «Robust video signature based on ordinal measure», [Xian-Sheng Hua](https://www.researchgate.net/profile/Xian-Sheng-Hua), [Xian Chen](https://www.researchgate.net/scientific-contributions/Xian-Chen-2031444407), [Hong-Jiang Zhang](https://www.researchgate.net/scientific-contributions/Hong-Jiang-Zhang-8023063). [9]

В работе используются порядковые подписи для моделирования относительного распределения интенсивности в кадре. Расстояние между двумя фрагментами измеряется с помощью временного сходства подписей. Подход позволяет искать нечеткие дубликаты видео с разными разрешением, частотой кадров, с незначительными пространственными изменениями кадров. Плюсом алгоритма является возможность работы в режиме реального времени. К минусам можно отнести неустойчивость к большим вставкам лишних кадров. Метод плохо применим для поиска естественных нечетких дубликатов, например, если объект снимался при разной освещенности.

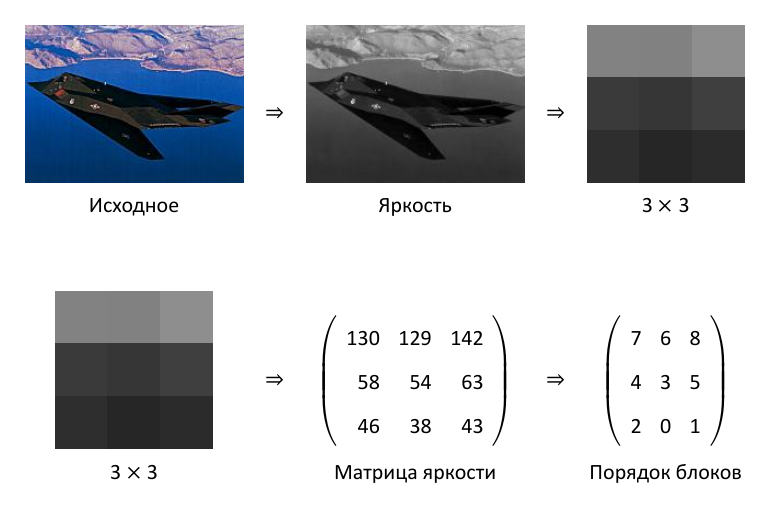


Рисунок 1.5 – Пример работы метода с порядковыми подписями

8) «Near-Duplicate Video Retrieval Based on Clustering by Multiple Sequence Alignment», Yandan Wang, Mohammed Belkhatir, Bashar Tahayna. [10]

Предложенный в работе метод основан на множественном выравнивании последовательностей (MSA). Подобный подход используется в биоинформатике для поиска выравнивания последовательностей ДНК. Авторы используют эвристическое выравнивание и итеративные методы. Для представления видео как геномов выделяют ключевые кадры, переводят в полутоновое цветовое пространство и разбивают кадры на блоки 2 × 2. Возможно всего 24 пространственных порядковых шаблона. Каждому такому шаблону можно поставить в соответствие букву латинского алфавита. Метод обладает высокой точностью и полнотой и не требует особенных вычислительных затрат. Минусы подхода — метод никак не учитывает временную информацию видео.

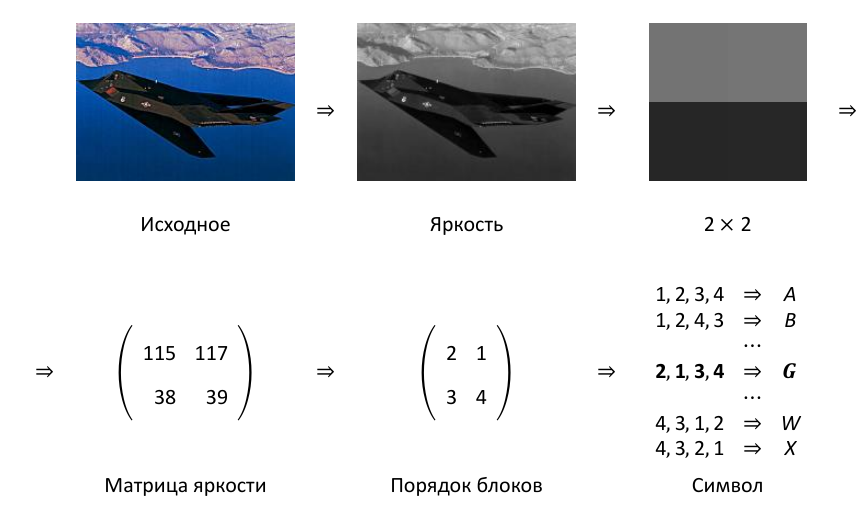


Рисунок 1.5 – Пример работы метода с ДНК-представлениями

В результате эффективные средства индексации и поиска видео в настоящее время является незаменимым компонентом во многих приложениях для поиска видео, управление рекомендациями, обнаружения копий и защиты авторских прав.

Таким образом, задача разработки алгоритмов индексирования и сопоставления видеофайлов по ключевым кадрам требует проведения интенсивных исследований и является актуальной.

* 1. Определение цели исследовательской работы

На рынке представлены несколько онлайн платформ, а также систем, реализующие автоматизированную проверку программного кода.

Цель исследования – разработать алгоритм создания уникальных подписей ключевых кадров видеопоследовательности для применения его в задачах поиска видеофрагментов.

Для достижения данной цели необходимо выполнить следующие частные задачи:

1. Проанализировать особенности систем видеокомпрессия.
2. Проанализировать существующие алгоритмы создания уникальных подписей ключевых кадров видеопоследовательности.
3. Разработать алгоритм создания уникальных подписей ключевых кадров видеопоследовательности.
4. Спроектировать и реализовать приложение с реализацией разработанного алгоритма
5. Оценить эффективность разработанного алгоритма.
6. Оценить возможности реализованного приложения в задачах поиска видеофрагментов.

Гипотеза: Алгоритм создания уникальных подписей ключевых кадров видеопоследовательности на основе темпоральных и пространственных характеристиках позволит решить задачу поиска заимствований видеофрагментов.

* 1. Используемые методики исследования

В соответствии с намеченной целью и задачами исследования были определены следующие методы:

* Теоретический анализ литературы и различных источников по теме исследования.
* Измерение.
* Сравнение.
* Практическое моделирование.
* Эксперимент.

При анализе источников будут использованы научно-исследовательские статьи и интернет-источники для сбора информации. Это позволит сформировать представление о существующих методах индексации видеофайлов.

Для выявление наиболее эффективных способов индексации и сопоставления видеофайлов необходимо использовать методы измерения. Ключевой характеристикой при измерении будет эффективность алгоритма как индексации, так и сопоставления сигнатур видеофайла.

Сравнение позволит определить наиболее эффективные методы для достижения поставленной цели исследования.

При практическом моделировании будет сформирована модель информационной системы с реализацией алгоритмов индексации и сопоставления видеофайлов.

Эксперимент позволит воспроизвести ожидаемые результаты работы алгоритма и оценить эффективность использованных алгоритмов.

Эффективность алгоритма индексирования определяется следующими аспектами:

* Точность.
* Достоверность.
* Временная сложность алгоритма.

Точность системы индексирования определяет вероятность правильной идентификации видеофайла или сигнатур. Этот показатель является критичным и влияет на работу алгоритма в целом.

Под достоверностью подразумевается актуальность результатов работы алгоритма индексирования. Это связанно с абстрагированием сигнатур видеофайла. При использовании неэффективных методов индексирования, возможна ситуация, когда разные видеофайлы определятся как идентичные.

Временная сложность алгоритма означает какое время будет затрачено, для решения алгоритма. Использование алгоритма будет нецелесообразной, при высоких временных затратах.

Гипотеза: возможно разработать эффективный алгоритм, который позволяет индексировать видеозаписи по их ключевым кадрам, а также сопоставить их для поиска общих кадров.

Выводы по разделу

В первом разделе был проведен анализ предметной области исследования, в результате чего была выявлена научная проблема. Далее на основе научных публикаций была подтверждена актуальность темы исследования. В результате была определена цель исследовательской работы, сформулированы частные задачи исследования и гипотеза. Определены методы исследования.

1. РАЗРАБОТКА АЛГОРИТМА СОЗДАНИЯ УНИКАЛЬНЫХ ПОДПИСЕЙ КЛЮЧЕВЫХ КАДРОВ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ
   1. Анализ особенностей систем видеокомпрессии

Одной из важнейших практических задач в области цифрового телевидения является задача сокращения скорости передачи двоичных символов и, соответственно, требуемой полосы частот канала связи. Эта задача может быть решена путем уменьшения избыточности информации, передаваемой в телевизионном сигнале без заметного для глаза ухудшения качества воспроизводимого телевизионного изображения. Уменьшение избыточности обеспечивает также сокращение требуемого объема памяти запоминающих устройств при записи телевизионных программ, видеофрагментов или отдельных изображений.

Сигнал, несущий определенную информацию, можно сжать путем удаления из него имеющейся избыточности. Избыточность – это компоненты данных, без которых можно обойтись для верного изображения исходной информации. Многие типы данных содержат в себе статистическую избыточность. Такие данные можно эффективно сжимать, используя компрессию без потерь. К сожалению, сжатие без потерь применительно к видео дает относительно небольшой выигрыш. Поэтому для достижения высокой эффективности сжатия приходится применять сжатие с потерями. При сжатии видео с потерями используется несколько типов избыточности:

* когерентность областей изображения - малое изменение цвета изображения в соседних пикселях (свойство, которое эксплуатируют все алгоритмы сжатия изображений с потерями);
* избыточность в цветовых плоскостях - используется большая важность яркости изображения для восприятия;
* подобие между кадрами - использование того факта, что на скорости 25 кадров в секунду, как правило, соседние кадры изменяются незначительно

В соответствии с установившейся в технической литературе терминологией уменьшение объема передаваемой видеоинформации в большинстве случаев называется сжатием видеоинформации или видеокопрессией.

Принципы видеокомпрессии во всех кодерах примерно одинаковы и отличаются только реализацией. Каждый способ реализации имеет свои особенности. Эти особенности влияют на количество артефактов компрессии и степень сжатия, производимой разными кодерами. В стандартах описаны только принципы декодирования, а не принципы кодирования. Поэтому кодеры одного и того же стандарта могут давать разное качество изображения или звука.

Алгоритмы сжатия (компрессии) делятся на два больших класса [11]:

* внутрикадровое кодирование (осуществляется пространственное сжатие информации);
* межкадровое кодирование (осуществляется временное сжатие).

Внутрикадровое кодирование осуществляет компрессию изображения с использованием только информации о текущем кадре.

Межкадровое кодирование осуществляет компрессию изображения с использованием информации о других кадрах видеопоследовательности. Основная идея компрессии состоит в том, что передается разница изображений между двумя кадрами, а не сами эти изображения. Один кадр видеопоследовательности является ключевым (keyframe), а все остальные кадры передаются как разница между передаваемым кадром и ключевым кадром. Обычно передается несколько десятков ключевых кадров в минуту.

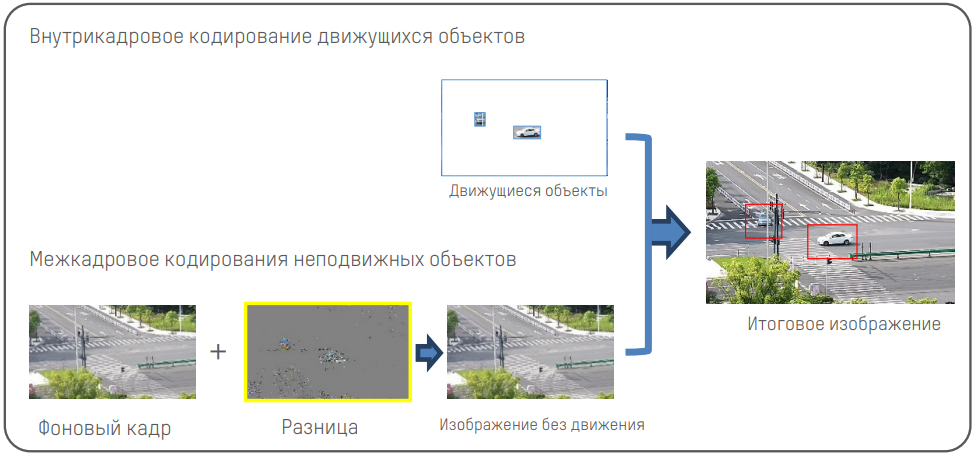


Рисунок 2.1 – Процесс формирования кадра с использованием межкадрового и внутрикадрового кодирования

Практическими вопросами сжатия аудиовизуальной информации и выпуском соответствующих стандартов и рекомендаций занимается MPEG— созданная в 1988 г. организация, объединяющая представителей фирм-производитслей оборудования и научных институтов разных стран. MPEG представляет собой подкомитет двух международных организаций — ISO и Международной электротехнической комиссии (IEC). Одна из основных задач группы MPEG состояла в изучении проблемы и разработке стандарта на компрессию цифрового телевизионного сигнала, что позволило предложить способы записи или передачи сигналов изображения и звука посредством возможно меньшего числа данных возможно лучшим качеством. [12]

Для поддержки межкадрового и внутрикадрового кодирования поток данных MPEG содержит три типа закодированных кадров:

* I-кадры (внутрикадровое кодирование);
* Р-кадры (кодирование с предсказанием);
* В-кадры (двунаправленное кодирование).

В I-кадре записан один кадр видеоданных, который никак не связан с информацией любого другого кадра. Все остальные кадры анализируются процессором, который сравнивает их с опорными, а также между собой. Данный кадр является ключевым.

Р-кадр содержит различия между текущим кадром и предыдущим I- или Р-кадром. Кодирование Р-кадров выполняют с использованием алгоритмов компенсации движения и предсказания «вперёд» по предшествующим I- и Р-кадрам. Они сжаты в три раза сильнее, чем I-кадры, и служат опорными для последующих Р- и В-кадров. Компенсация движения, применяемая к микроблокам Р-кадров, вырабатывает два вида информации: векторы движения (разница между базовыми и кодированными макроблоками) и значения ошибок (разница между предсказанными и действительными результатами)

В-кадр состоит из различий между текущим кадром и двумя (предыдущим и последующим) I- или Р-кадрами, то есть с двунаправленным предсказанием и компенсацией движения. В-кадры имеют наибольшее сжатие.

Проще говоря, Р-кадры «предсказываются» на основании разницы с предыдущим по времени ключевым кадром или предыдущим Р-кадром. В- кадры «предсказываются» на основании различий с предыдущим и последующим ключевыми кадрами или соседними Р-кадрами.

Соседние кадры, как правило, мало отличаются друг от друга. Если вычесть один кадр из другого, получится остаточный кадр, который будет содержать незначительное количество информации. Он будет заполнен преимущественно нулевыми значениями. Такие данные легко поддаются компрессии. Количество нулевых значений можно еще увеличить, если до вычитания произвести компенсацию движения. Компенсация движения — это последовательность действий, которая позволяет описать перемещающиеся участки изображений в двух последовательных кадрах. Для компенсации движения передается информация об участке изображения, который движется, и информация о направлении движения. В тех местах, где перемещался участок изображения, получатся нулевые значения в остаточном кадре (рисунок 2.2).

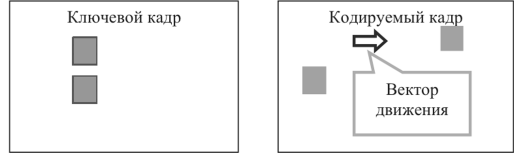


Рисунок 2.2 – Ключевой и кодируемый кадры при компенсации движения

На рисунке 2.3 показана группа кадров и входящие в нее отдельные кадры различных типов. Группа кадров ограничена двумя I-кадрами, между которыми расположены в определенном порядке P и B-кадры.

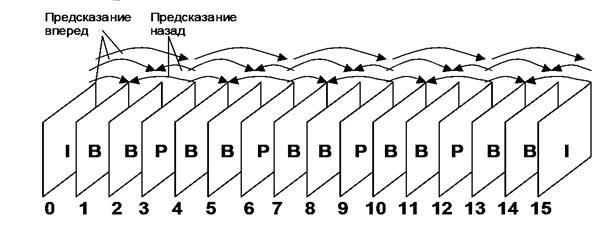


Рисунок 2.3 – Пример структурной группы кадров

Закодированный видеофайл хранится в медиаконтейнерах (рисунок 2.4). Медиаконтейнер - специальный формат файла, в спецификациях которого определен способ хранения данных в файле, без указания алгоритма его кодирования.



Рисунок 2.4 – Пример структурной группы кадров

То есть медиаконтейнер определяет количество сохраненных в файле данных, но не может определить никакой кодификации этих данных. В сущности, медиаконтейнер представляет собой метаформат, поскольку в нем хранится информация, определяющая, каким именно способом данные будут сохранены внутри файла. Несмотря на то, что большинство контейнеров привязаны к определенному формату, в некоторых из них может храниться видео в совершенно разных стандартах. Например, файл с расширением AVI способен содержать ролики как в формате MPEG-1, так и в MPEG-2 или в MPEG-4. Также, в медиаконтейнере может сохраняться несколько однотипных потоков, например фильм (видеопоток) с несколькими звуковыми дорожками (аудиопотоками) и субтитрами (текстовыми потоками).

Таким образом ключевые кадры являются главным источником визуального содержания видеопотока. Для индексирования всего видеофайла целесообразно создавать уникальные подписи именно ключевых кадров.

* 1. Алгоритмы обработки ключевых кадров видеопотока для оценки их схожести

Так как ключевые кадры являются изображениями, для них справедливо использовать методы пространственной обработки изображений. На данный момент разработано много методов, и чтобы выбрать, какие из них нужно использовать для достижения поставленной цели, нужно проанализировать особенности данных методов.

Видеопоток – это последовательность большого количества изображений. Если видеофайл имеет длинную продолжительность, то общее количество изображений, которое требуется проиндексировать, становится огромным. В первую очередь стоит обратить внимание на методы, которые работают быстро и используют простую информацию, которую можно извлечь из изображения даже плохого качества.

Из, методов, которые удовлетворяют данным условиям, были выбраны три: метод цветовых гистограмм, методы матрицы переходов и метод перцептивного хэша. В следующих параграфах производится обзор этих методов.

* + 1. Метод цветовых гистограмм

Первым среди методов индексирования изображений рассматривается алгоритм построения цветовых гистограмм. Принцип действия такого алгоритма основан на подсчете попаданий пикселей в определенные диапазоны цветов, что позволяет проводить сравнение изображений путем нахождения расстояний между найденными гистограммами. Далее будет дано краткое описание алгоритма на основе материалов [13].

Цветовое RGB-пространство рассматривается как трехмерный куб, каждая ось которого соответствует одному из трех основных цветов (красному, зеленому или синему), деления на осях пронумерованы от 0 до 255 (большее значение соответствует большей интенсивности цвета). При таком рассмотрении любой цвет RGB-изображения может быть представлен точкой куба. Для построения цветовой гистограммы каждая сторона делится на n равных интервалов, соответственно RGB-куб делится на N прямоугольных параллелепипедов. Гистограмма изображения отражает распределение точек RGB-пространства, соответствующих цветам пикселей изображения, по параллелепипедам.

Для работы алгоритма требуется обработать каждый канал пикселя и проинкрементировать соответствующее значение вектора. Соответственно, для обработки всего пикселя компьютеру нужно произвести 4 арифметических операций, что в итоге дает в сумме операций и оценку временной сложности .

Для определения степени схожести гистограмм существуют несколько методов [14]. В стандартном случае после нахождения гистограммы производится её нормализация: значение каждой ячейки гистограммы делится на сумму всех ячеек, что не что иное, как количество пикселей изображения.

Рассматривается три формулы вычисления схожести гистограмм. Стоит отметить, что наиболее простой является первая, однако при распознавании изображений может проигрывать другим метрикам в качестве.

1) Пересечение гистограмм. Расстояние между гистограммами рассчитывается как: (1)

(1)

где *Hi* – гистограмма изображения.

Возвращаемое значение лежит в диапазоне [0, 1]. Чем больше значение, тем более схожи изображения. Преимуществом этой метрики считается её простота и скорость, так как для каждой пары элементов производится только одна операция сравнения, поэтому подсчет расстояния происходит достаточно быстро. Благодаря этому преимуществу этот метод получил широкое распространение.

2) Расстояние Бхатачария. Формула вычисления расстояние выглядит следующим образом: (2)

(2)

где *Hi* – гистограмма изображения.

Функция принимает значение из отрезка [0; 1]. Чем меньше получившееся значение, тем более схожи изображения. 0 соответствует полному сходству, 1 – наиболее сильному различию.

3) Метрика хи-квадрат. Формула расчета расстояния между гистограммами принимает следующий вид: (3)

(3)

где *Hi* – гистограмма изображения.

Функция возвращает значение из диапазона [0; ∞]. Меньшее значение функции соответствует большей схожести изображений.

* + 1. Метод матриц смежности

Помимо методов, основывающихся на цветовом сравнении изображений, есть и иные идейные подходы поиска схожих изображений. В частности, характеристикой изображения может служить её текстура. Матрицы переходов – один из способов описания текстуры изображений, который позволяет представить текстуру изображения в виде многомерного вектора [15]. Алгоритм построения матрицы переходов описывается следующими шагами.

**Шаг 1.** Пусть имеется изображение размером N на M. Сначала строится матрица той же размерности, которая задает уровень яркости каждого пикселя.

**Шаг 2.** Происходит квантование яркости на несколько уровней. Например, значения яркости для пикселя равно 255. Яркость квантуется на 4 области, получается 4 новых уровня яркости с индексами 0, 1, 2 и 3. Значение 255 попадает в область 192-255, поэтому в новой матрице для данного пикселя запишется число 3.

**Шаг 3.** Вычисляются матрицы переходов. Их размерности равны k × k, где k- уровни квантования (для предыдущего примера k = 4). У каждой матрицы смежности есть направление. Например, если направление матрицы равно *(x,y) = (1,0),* то она характеризует переходы пикселей вдоль оси x (направо). Значение i-ой строки и j-ого столбца матрицы перехода равно количеству переходов значения уровня яркости (i - 1) в значение уровня яркости (j - 1) по направлению (x,y). Например, значение элемента первой строки первого столбца матрицы переходов по направлению (0, 1) показывает, сколько пикселей в квантованной матрице со значением 0 имеют справа пиксель с тем же значением 0.

Матрицы переходов вычисляются по нескольким направлениям. В стандартном случае вычисляют 4 матрицы: по направлениям вправо, влево, вверх, вниз.

**Шаг 4.** Полученные матрицы преобразуются в вектор размерности 128. Наглядным образом работу алгоритма на примере демонстрирует рисунок ниже (рис. 1).

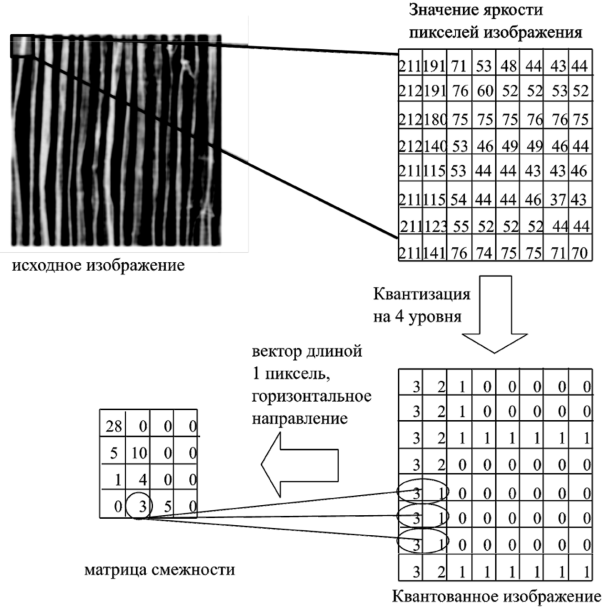


Рисунок 2.5 – Пример формирования матрицы смежности

Для работы алгоритма требуется перевести изображение в градации серого, что потребует операций. Для получения квантованной матрицы яркости потребуется обработать каждый пиксель и произвести одну арифметическую операцию, что дает операций. После этого строится 4 матрицы переходов, для которых должны обработаться все элементы квантованной матрицы 4 раза, при каждой обработке инкрементируя соответствующий элемент матрицы переходов. Получается операций.

Суммируя полученные данные, получаем порядка операций и оценку сложности алгоритма . Она аналогична оценке сложности метода цветовых гистограмм, однако коэффициент при старшем члене в данном случае выше, поэтому скорость алгоритма должна быть ниже.

* + 1. Метод перцептивных хэшей

Алгоритм перцептивного хэша используется для разных задач обработки изображений. Перцептивный хэш – это такая функция, генерирующая некоторый отпечаток изображения, который удобно сравнивать с отпечатками других изображений. Существует много вариантов реализации этой функции, автором, на основе изученного материала [16] рассмотрен один вариант функции, оптимальный с точки зрения скорости, описание которого представлено ниже.

Алгоритм построения простого перцептивного хэша (Simple Hash) изображения состоит из следующих шагов.

**Шаг 1.** Уменьшение размера исходного изображения до пикселей размерности N на N.

**Шаг 2.** Перевод уменьшенного изображения в градации серого.

**Шаг 3.** Вычисление среднего значения цвета всех пикселей.

**Шаг 4.** Бинаризация всех пикселей изображения по найденному среднему значению: если значение пикселя больше среднего, то новое значение равно 1, иначе – 0;

**Шаг 5**. Перевод массива битов в одно значение 16-ричной системы счисления.

При реализации алгоритма в сумме получается операций . Получается оценка сложности .

Для сравнения полученных хэшей используется расстояние Хэмминга: оно равно количеству битов различающихся битов в сравниваемых цепочках. Чем меньше это значение, тем более похожи изображения.

* 1. Выбор метода обработки пространственных характеристик ключевых кадров

Рассмотренные методы из предыдущего раздела были оценены в работе [17]. В итоге метод цветовых гистограмм показал результаты в среднем хуже метода матриц смежности. Метод цветовых гистограмм и метод матриц смежности показали примерно равные результаты в плане качества, в плане скорости лучше показал себя метод матриц переходов. Однако не стоит забывать, что индексация у этого метода намного дольше, чем у метода цветовых гистограмм, поэтому в суммарном времени они проигрывает.

Алгоритм сравнения кадров внутри видеопотоков по качеству показал наилучшие результаты, а если быть точнее, то вариант его реализации с использованием хешей. Метод цветовых гистограмм ненамного отстает от метода перцептивных хешей.

Таким образом, наилучший результат показал алгоритм сравнения кадров внутри видеопотоков с использованием перцептивного хеша. Поэтому принято решение использовать данный метод.

* 1. Разработка алгоритма создания уникальных подписей ключевых кадров видеопоследовательности

Использование перцептивного хэша позволит получить уникальные подписи ключевых кадров для индексации пространственных характеристик. Исходя из того, что видеофайл обладает темпоральными характеристиками, было принято решение использовать данное свойство при индексации видеофайла. Это позволит сделать индексы уникальными, что улучшит точность нахождения заимствований. Предлагается использовать в качестве индексов перцептивные хэши ключевых кадров в точках смен сцен (склейки), а в качестве темпоральных характеристик – использовать длину сцены.

Алгоритм создания уникальных подписей ключевых кадров видеопоследовательности состоит из следующих шагов.

**Шаг 1.** Извлечение всех ключевых (I-ых) кадров из видеофайла в разрешении *N×N*. По умолчанию предлагается использовать разрешение 8*×*8.

**Шаг 2.** Получение перцептивных хэшей всех извлеченных ключевых кадров. Описание алгоритма формирования перцептивных хэшей описан в первом разделе данной главы.

**Шаг 3.** Сравнение полученных хэшей с помощью расстояния Хэмминга.

**Шаг 4.** Удаление хэшей с меньшим расстоянием Хэмминга. Таким образом остаются хэши только в точках смен сцен. Расстояние Хэмминга принимает значение от 0 до 16. Нулевое расстояние означает, что это, скорее всего, одинаковые картинки. Дистанция 5 означает, что картинки в чём-то отличаются, но в целом всё равно довольно близки друг к другу. Если дистанция 10 или больше, то это, вероятно, совершенно разные изображения. По умолчанию предлагается использовать значение 9.

**Шаг 5.** Расчет длительности сцены от одного ключевого кадра до следующего.

Наглядный пример представлен на рисунке 2.6.

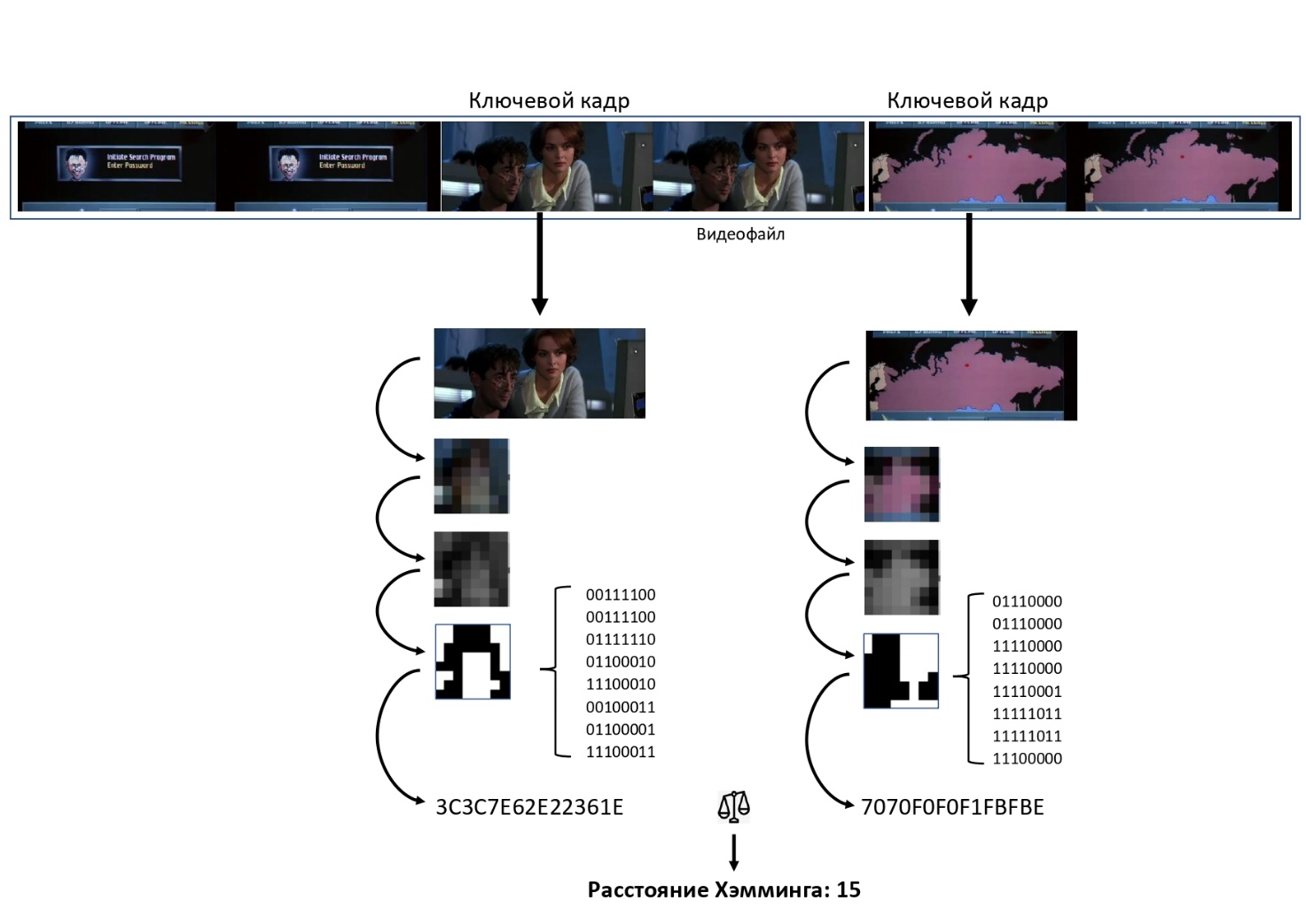


Рисунок 2.6 – Процесс формирования уникальных подписей

Расстояние Хэмминга – это число позиций, в которых соответствующие символы двух слов одинаковой длины различны [18]. Пример расчета длины изображен на рисунке 2.7.

Расстояние Хэмминга рассчитывается как: (4)

(4)

где , – последовательности длиной .

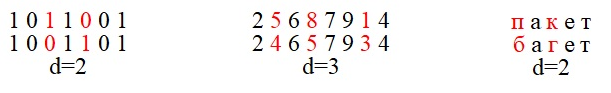


Рисунок 2.7 – Пример расстояния Хэмминга

Полученные значения образуют массив индексов видеофайла, где каждый элемент массива — это совокупность пространственной и темпоральной характеристики ключевого кадра. Данный элемент массива имеет следующий вид: {хэш, длительность сцены}.

Блок схема алгоритма получения хэша представлена на рисунке 2.8. Входными данными являются массив из пикселей изображения в градации серого и переменная со значением среднего цвета.

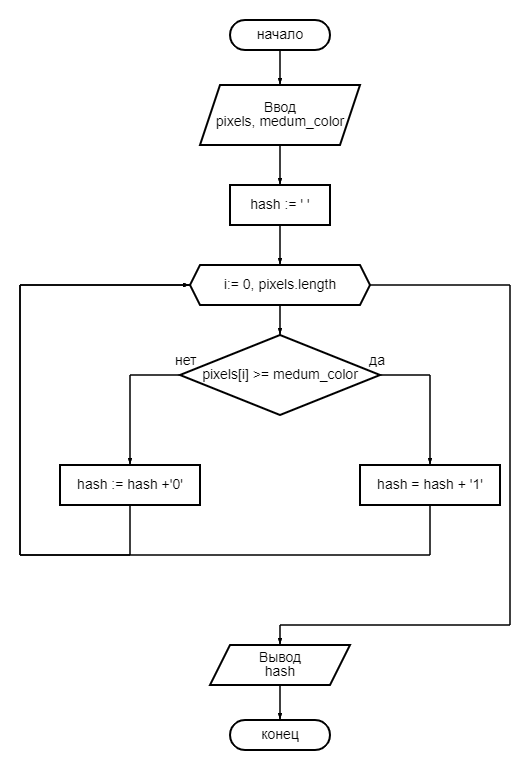


Рисунок 2.8 – Блок-схема формирования перцептивного хэша

Блок схема алгоритма получения дистанции Хэмминга представлена на рисунке 2.9. Входными данными являются хэш текущего кадра и хэш следующего кадра. Результатом работы является значение дистанции длины Хэмминга.

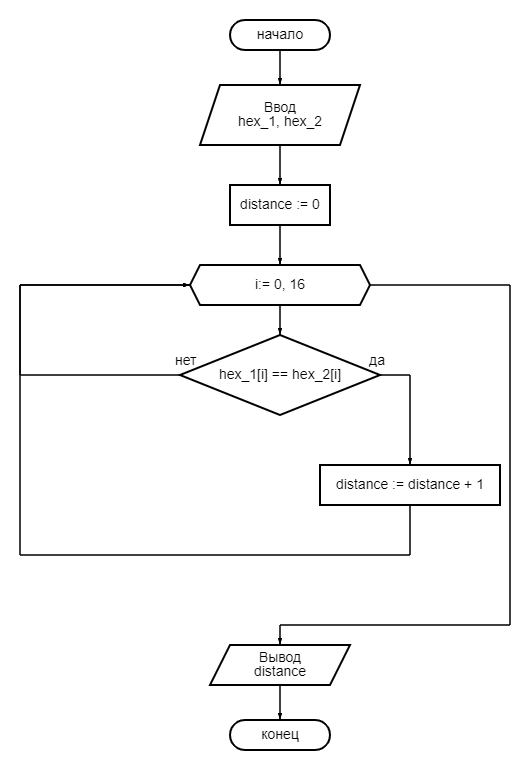


Рисунок 2.9 – Блок-схема формирования расстояния Хэмминга

Сравнение индексов видеофайлов между собой будет включать в себя попарное сравнение каждого индекса одного видеофайла с каждым индексом второго видеофайла. Исходя из того, что элементом индекса являются два значения, предлагается использовать алгоритм расстояния Хэмминга (например, 1) для сравнения хэшей и модуль разности длительности сцен с определением точности (например, 0.1 секунд). Определение точности для темпоральной характеристики обусловлен тем, что временные метки ключевых кадров в видеофайлах определяются с некоторой неточностью. Это связано с разной частотой кадров в секунду в видеофайлах.

* 1. Проектирование информационной системы с реализацией алгоритма создания уникальных подписей ключевых кадров видеопоследовательности
     1. Разработка модели TO-BE потоков данных DFD приложения

Диаграмма потоков данных показывает внешние источники и адресаты

данных, логические функции, потоки данных и хранилища данных.

На рисунке 2.10 представлена контекстная диаграмма информационной системы.

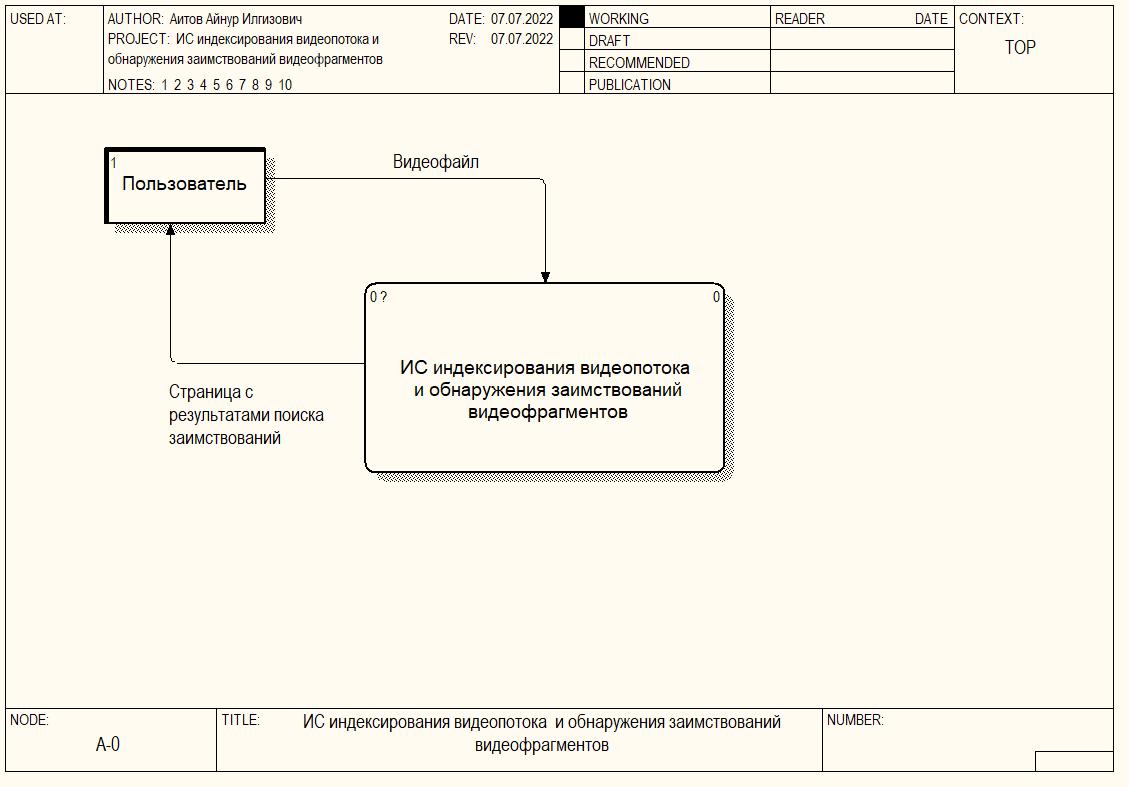


Рисунок 2.10 – Контекстная диаграмма «Информационная система индексирования видеопотока и обнаружения заимствований видеофрагментов» в нотации DFD

В данной диаграмме выделена единственная внешняя сущность – пользователь.

Выделены следующие потоки данных:

* Видеофайл. Пользователь загружает видеофайл в информационную систему для поиска «по образцу».
* Страница с результатами поиска заимствований. Пользователю возвращается страница с результатами поиска заимствований среди видеофайлов, что находятся в базе данных информационной системы.

Далее произведена декомпозицию системы. На рисунке 2.11 проиллюстрированы основные процессы внутри системы. Данная диаграмма показывает какие процессы и связи устроены внутри системы.

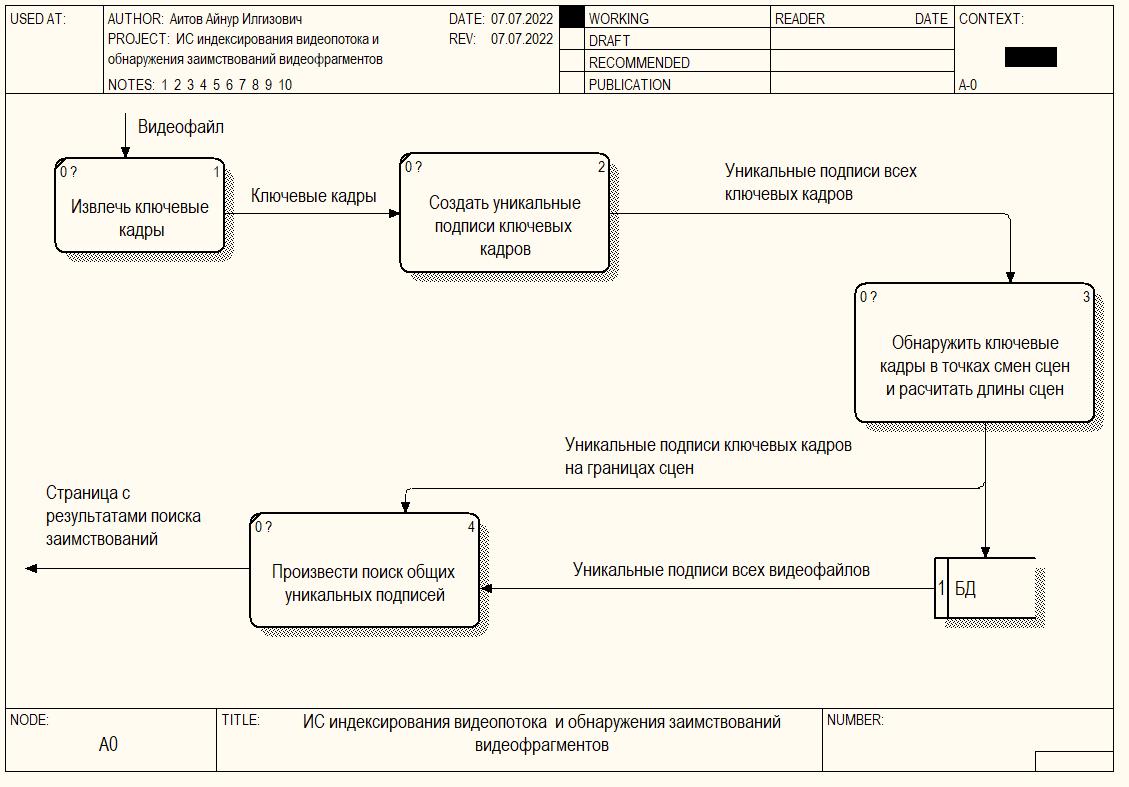


Рисунок 2.11 – Декомпозиция контекстной диаграммы «Информационная система проверки программных решений» в нотации DFD

Процессы, рассмотренные в диаграмме:

* Извлечь ключевые кадры. На данном этапе информационная система извлекает все ключевые кадры из загруженного пользователем видеофайла.
* Создать уникальные подписи ключевых кадров. На основе ключевых кадров строится перцептивный хэш для каждого кадра.
* Обнаружить ключевые кадры в точках смен сцен и рассчитать длительность сцен. В зависимости от расстояния Хэмминга, перцептивные хэши соседних ключевых кадров сравниваются и удаляются индексы с низким расстоянием. Таким образом формируются ключевые кадры в точках смен сцен. Так же рассчитывается длительность сцены на основе уже отфильтрованных индексов. Индексы загруженного пользователем файла отправляются в базу данных.
* Произвести поиск общих уникальных подписей. Информационная система получает из базы данных индексы уже проиндексированных видеофайлов и сравнивает их с индексами загруженного файла.
  + 1. Разработка базы данных информационной системы

После изучения предметной области и моделирования основных процессов в системы были выделены основные сущности. После выявления основных сущностей, были выявлены атрибуты этих сущностей и связи между ними.

На рисунке 2.12 отображена диаграмма логической модели данных, на которой отображены сущности, и их атрибуты.



Рисунок 2.12 – Логическая модель базы данных

Описание использование каждой сущности представлена ниже.

*Сущность «Видеофайл».* Используется для хранения данных о ранее проиндексированных видеофайлах. Атрибуты сущности позволяют хранить название, длительность видеофайла и другие данные для анализа эффективности работы информационной системы.

*Сущность «Индексы».* Используется для хранения индексов видеофайлов. Атрибуты сущности позволяют хранить темпоральные и пространственные индексы, а также время ключевого кадра, которая затем будет использована при формировании страницы с результатами проверки.

* + 1. Обоснование выбора системы управления базами данных

Так как индексирование видеофайлов подразумевает работу с информацией, нужно определиться, в каком виде ее хранить. Наиболее наглядно и правильно это реализуется в виде таблиц базы данных. Ниже приведены описания популярных на сегодняшний день систем управления базами данных.

**Система управления базами данных MySQL**

MySQL – это свободная система управления базами данных. Официально разработчиком системы являлась компания MySQL AB, сейчас поддержку и разработку системы осуществляет известная корпорация Oracle [19].

СУБД MySQL первоначально создавалась для работы с большими массивами данных [19]. Однако, благодаря доступности и быстродействию этой СУБД, она весьма быстро захватила интернет. Существуют множество других системы управления базами данных, но сравниться по быстродействию с MySQL могут не многие. Помимо скорости работы, важным отличием от других СУБД является поддержка SQL, что автоматически обеспечивает высокий уровень кроссплатформенности кода и данных, созданных на базе MySQL. Благодаря этому отличительному признаку любой разработчик с легкостью может перенести данные своей базы данных в любую другую СУБД, а весь сохраненный код можно легко применять на любой из этих платформ.

Также система привилегий позволяет ограничивать каждую учетную запись правами на осуществление различных операций с данными, причем не только на серверном уровне, но и на уровне таблиц. Хэширование паролей обеспечивает высокий уровень безопасности.

Помимо универсальности и распространенности СУБД MySQL обладает целым комплексом важных преимуществ перед другими системами. В частности, следует отметить такие качества как:

* Простота в использовании. MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных.
* Обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта.
* Безопасность. Система изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию.
* Масштабируемость. Являясь весьма универсальной СУБД, MySQL в равной степени легко может быть использована для работы и с малыми, и с большими объемами данных.
* Скорость. Высокая производительность системы обеспечивается за счет упрощения некоторых используемых в ней стандартов.

К недостаткам этой СУБД относятся:

* Недостаточная надежность. В вопросах надежности некоторых процессов по работе с данными (например, связь, транзакции, аудит) MySQL уступает некоторым другим СУБД.
* Низкая скорость разработки. Как и многим другим программным продуктам с открытым кодом, MySQL не достает некоторого технического совершенства, что порой сказывается на эффективности процессов разработки.

**Система управления базами данных SQLite**

SQLite - легко встраиваемая кроссплатформенная база данных, поддерживающая полный набор команд SQL. Разработчик – Ричард Хипп. На данный момент СУБД распространяется свободно, имеет открытый код.

Благодаря тому, что система базируется на файлах, предоставляется большой набор инструментов для работы с ней в сравнении с сетевыми системами, так как реализованы прямые обращения к файлам, вместо портов сокетов и т.д. Все базы хранятся в файлах, благодаря чему количество баз и таблиц не ограничивается количеством, а ограничивается лишь местом на диске. Максимальный объем одной базы не может превышать 2ТБ.

Связь приложений с базой данных организуется с помощью прямых вызовов файлов, которые содержат в себе данные. На сегодняшний день SQLite пользуется большой популярностью среди разработчиков. В PHP5 поддержка СУБД SQLite добавлена по умолчанию.

Достоинства SQLite:

* Файловая система – вся база данных хранится в одном файле.
* Стандартизация – SQLite использует функции SQL.
* Поддержка – в сети большое количество документации и примеров работы.

Недостатки SQLite:

* Отсутствие пользовательского управления – нет возможности разграничивать права пользователей БД.
* Однопоточность – допускается единовременное исполнение только одной операции записи.

**Система управления базами данных PostgreSQL**

PostgreSQL – на сегодняшний день самая продвинутая система управления базами данных, ориентир которой направлен на расширяемость и соответствие стандартам. История PostgreSQL начинается в далеком 1986 году, когда Майкл Стоунбрейкер и его студенты разрабатывали новую СУБД. С 1986 до 1994 в синтаксис ввели процедуры, правила, типа пользователей и другие компоненты. Но в 1995 году разработка разделилась – студенты стали разрабатывать новую версию СУБД, названную Postgres, затем Postgres95. После того как разработчики покинули пределы университета, наработки были переданы команде энтузиастов, благодаря которым и появилось название PostgreSQL. Основное и самое важное отличие PostgreSQL от остальных систем управления базами данных – объектно-ориентированный функционал. СУБД легко справляется с одновременной обработкой сразу нескольких заданий.

Достоинства PostgreSQL:

* Полная совместимость с SQL.
* Объектно-ориентированный принцип.

Недостатки PostgreSQL:

* Сложность разработки.
* Низкая скорость чтения данных.

**Система управления базами данных MS SQL Server**

Ещё одной из популярных СУБД является программный продукт Microsoft SQL-сервер. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно. Вскоре после выпуска Microsoft SQL сервер 2016, Microsoft адаптировала продукт для операционной системы Linux, а на Windows-платформе он работал изначально.

Одной из уникальных особенностей версии 2016 года является temporal data support (временная поддержка данных), которая позволяет отслеживать изменения данных с течением времени. Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные. Идеально подходит для: крупных организаций, которые уже используют ряд продуктов Microsoft.

Достоинства:

* Простота в использовании.
* Стабильность и быстродействие системы.
* Отслеживание уровня производительности.
* Хорошая интеграция с другими продуктами Microsoft.

Недостатки:

* Высокая цена для юридических лиц.
* Высокая нагрузка ресурсов системы.

Из всего вышеперечисленного и принимая во внимание анализ СУБД методом экспертного оценивания, подходящим средством проектирования базы данных выбрано My SQL. Другие СУБД по методу оценок и по объективным показателям программирования не подходит, из-за высокой цены, сложности разработки, однопоточности и низкой популярности систем, который гарантирует длительную поддержку СУБД. My SQL удовлетворяет абсолютно всем потребностям проектирования физической модели данных, он свободно распространяется, легко интегрируется и масштабируется.

* + 1. Разработка физической модели базы данных

Физическая модель данных – это модель данных, описанная с помощью средств конкретной СУБД, которая строится на на основе логической с учетом ограничений, накладываемых особенностей выбранной СУБД (в данном случае – MySQL 5). К таким особенностям могут относиться поддерживаемые СУБД типы данных, соглашения о присвоении имен таблицам, атрибутам и т.д. Физическая модель данных фактически является готовым заданием на создание БД, имея которое можно реализовать БД в выбранной СУБД.

Каждая сущность и каждое отношение логической модели базы данных трансформируется в таблицу. Имена сущностей и отношений становятся именами таблиц, а имена атрибутов– именами колонок. Физическая модель базы данных представлена на рисунке 2.13.

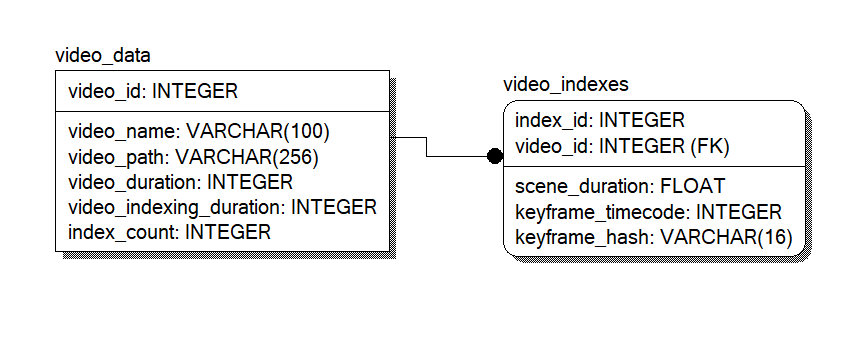


Рисунок 2.13 – Физическая модель базы данных

Описание физической модели данных представлено в таблицах с 2.1 и 2.2.

Таблица 2.1 – Описание таблицы «video\_id»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Краткое описание** |
| 1 | video\_id | INT | Уникальный  идентификатор видеофайла |
| 2 | video\_name | VARCHAR(100) | Наименование видео |
| 3 | video\_path | VARCHAR(256) | Наименование файла с видеопотоком |
| 4 | video\_duration | INT | Длительность видео |
| 5 | video\_indexing\_duration | INT | Длительность индексирования видеофайла |
| 6 | index\_count | INT | Количество индексов видеофайла |

Таблица 2.2 – Описание таблицы «video\_indexes»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Краткое описание** |
| 1 | index\_id | INT | Уникальный  идентификатор индекса |
| 2 | video\_id | INT | Уникальный  идентификатор видеофайла, которому данный индекс относится |

Продолжение таблицы 2.2

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | scene\_duration | FLOAT | Длительность сцены до следующего ключевого кадра |
| 4 | keyframe\_timecode | INT | Временная метка ключевого кадра |
| 5 | keyframe\_hash | VARCHAR(16) | Перцептивный хэш ключевого кадра |

* + 1. Критерии оценивания и показатели их вычисления

Необходимо оценить, насколько предложенный алгоритм индексации видеопотока эффективен по сравнению с существующими методами и позволяют работать с видеопотоками. Для этого существует ряд формальных критериев, по которым можно утверждать, насколько тот или иной метод применим для того или иного класса видеофайлов. В таком случае к основным количественным характеристикам по оценке качества изображений в целом относятся критерии, описанные ниже.

Средняя скорость построения индекса одной секунды видеофайла (AS): (5)

(5)

где – продолжительность видеофайла в секундах,

– время индексации видеофайла в миллисекундах.

Данная метрика обозначает с какой скоростью (в миллисекундах) создавались индексы для одной секунды видеофайла. Чем ниже значение, тем эффективнее работает алгоритм.

Процент заимствования видеофайла (BP): (6)

(6)

где – количество общих индексов,

– продолжительность сцены кадра,

– продолжительность видеофайла в секундах.

Данная метрика отражает найденные алгоритмом проценты заимствований между видеофайлом – образцом и найденного видеофайла с заимствованиями.

Достоверность индексирования (EBP): (7)

*\*100* (7)

где – процент найденного заимствования видеофайла,

– действительный процент заимствований видеофайла.

Данная метрика демонстрирует насколько достоверно алгоритм находит заимствования.

Выводы по разделу

Проанализированы особенности систем видеокомпрессии и рассмотрены алгоритмы обработки ключевых кадров видеопотока для оценки их схожести.

Разработан алгоритм индексации видеофайлов на основе темпоральных и пространственных характеристиках ключевых кадров.

Были построены и описаны модели «TO-BE» информационной системы в нотации DFD для описания движения потоков данных.

Была построена логическая модель базы данных, обоснован выбор системы управления базами данных, построена и описана физическая модель база данных.

Определены критерии оценивания разработанного алгоритма и показатели их вычисления.

1. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ И ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМА
   1. Обоснование выбора средств программной реализации ИС

В качестве платформы разработки был выбран Node.js. Данная программная платформа, основанная на движке V8 (транслирующем Javascript в машинный код), разработанный компанией Google, превращает Javascript из узкоспециализированного языка в язык общего назначения. NodeJS добавляет возможность Javascript взаимодействовать с устройствами ввода-вывода через свой API, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из Javascript-кода. В основе NodeJS лежит событийно ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом.

Node.js был разработан специально для того, чтобы обеспечить разработку сетевых приложений. В связи с этим в стандартной поставке включены инструменты для лёгкой работы с сетью, в частности, Node.js позволяет запустить http сервер с использованием всего одной стандартной функции [20].

Ниже представлены причины выбора платформы Node.js.

Простота. В основном, библиотеки в Node имеют простые API с которыми легко разобраться и которые работают интуитивно понятным образом. Если одна из библиотек по какой-либо из причин не подойдёт для разработки, то ей можно будет найти замену, так как их существует огромное множество.

Контроль. Программист сам строит инфраструктуру проекта, выбирая и объединяя небольшие модули для конкретных задач. Полученный опыт легко применять в дальнейшем.

Легкое развертывание. Node очень легко разворачивать на сервере: как на Linux, так и на Windows.

Межплатформенность. На основе Node будет реализовано веб приложение, которое будет доступно во всех устройствах при наличии браузера и выхода в сеть интернет.

Производительность. Node основан на высокопроизводительном движке V8, он асинхронен и не блокирует процесс выполнения во время длительных операций, таких как вычитка файла или обращение к базе данных. Это позволяет достичь высокого уровня производительности при использовании единственного потока.

Использование данной платформы позволит реализовать генерацию интерфейса пользователя с помощью шаблонов страниц.

Так, как разрабатываемая система будет веб приложением, то при организации интерфейса пользователя будут использованы следующие языки:

* HTML, для разметки интерфейса.
* CSS, для оформления интерфейса.
* JavaScript, для определения поведения при взаимодействии с элементами интерфейса.

HTML (HyperText Markup Language, «язык разметки гипертекста») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). В разработке данной информационной системы будет использоваться стандарт HTML 5.

CSS - язык таблиц каскадных стилей. Он разработан для того, чтобы расширить возможности по оформлению Web-страниц. CSS используется веб-разработчиками для задания внешнего вида (шрифтов, цветов, отступов, расположения и др.) веб-страниц. CSS разработан для отделения основного содержимого документа (написанного на языке разметки, например HTML) от оформления этого содержимого (написанного на CSS). Такое отделение предоставляет веб-разработчикам большую гибкость, упрощает задание внешнего вида документов и оформление повторяющихся элементов разметки. В разработке данной информационной системы будет использоваться стандарт CSS 3.

JavaScript — это язык программирования, который служит для управления сценариями просмотра Web-страниц. Главная особенность данного языка заключается в том, что при его использовании есть возможность изменять свойства среды отображения при просмотре Web-сайта и при этом не будет происходить перезагрузка Web-страниц. К примеру, с помощью языка JavaScript можно заменить цвет фона Web-страницы или заменить изображение, интегрированное в Web-страницу, также есть возможность создать новое окно отображения либо вывести различные уведомления.

**Выбор библиотек для разработки**

Для генерации интерфейса пользователя использовалась библиотека представлений, шаблонизатор для Node.js — EJS (Embedded JavaScript templates). Embedded JavaScript предлагает совершенно прямолинейный подход к шаблонизации, который покажется очень близким тому, кто использует шаблонизаторы в других языках. В этом шаблонизаторе теги внедряются в НТМL код в качестве местозаполнителей для данных. Кроме того, EJS позволяет выполнять в шаблонах исходную JavaScript-логику для таких операций, как условное ветвление и итерация, как это делается в РНР [21].

Для быстрой разработки интерфейса пользователя, применялась популярная библиотека Bootstrap. Включает в себя HTML и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения [22]. В данной работе библиотека Bootstrap 4.5 позволила использовать заранее созданные CSS стили для оформления интерфейса, а также для поддержки различных типов устройств, создав адаптивную систему сеток.

Для обработки видеофайлов использована библиотека с открытым исходным кодом FFmpeg. FFmpeg — набор свободных библиотек с открытым исходным кодом, которые позволяют записывать, конвертировать и обрабатывать видео в различных форматах. Благодаря данной библиотеки возможно извлечь ключевые кадры, получить длительность видеофайла, а также получить временные метки ключевых кадров.

* 1. Описание программной реализации

Для реализации проекта разработаны экранные формы, состоящая из 5 представлений (шаблонов).

Главная страница является стартовой (рисунок 3.1), содержит экранную форму с краткой информацией о возможностях информационной системы. В форме представлены несколько ссылок для перехода на страницу выполнения запроса.

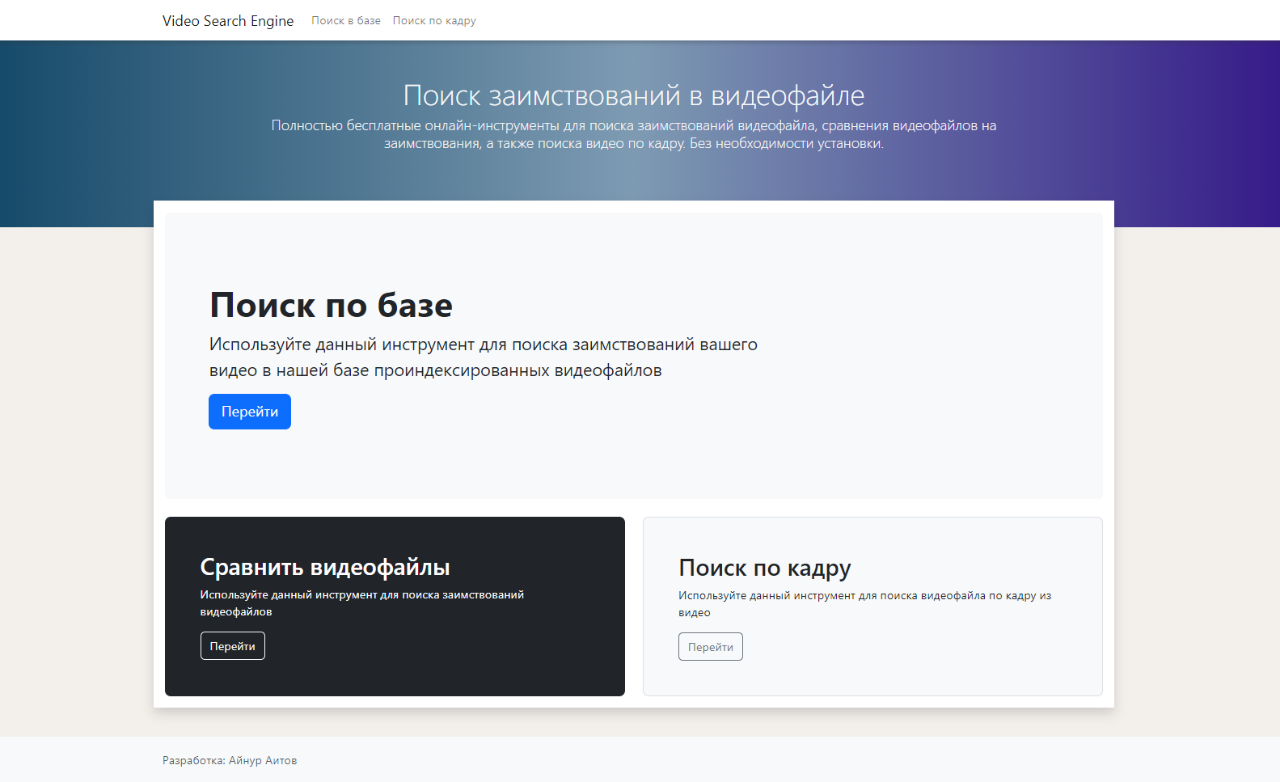


Рисунок 3.1 – Экранная форма главной страницы

Благодаря использованию систему сеток библиотеки Bootstrap, интерфейс пользователя адаптируется под разрешение экрана пользователя. Результат адаптивности интерфейса главной страницы под мобильные устройства представлена на рисунке 3.2.

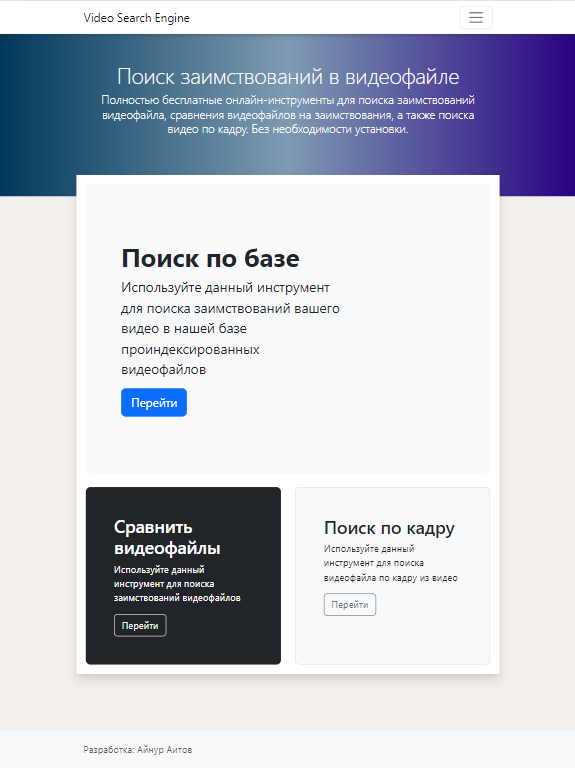


Рисунок 3.2 – Экранная форма главной страницы на мобильных устройствах

Процесс поиска заимствований видеопотока происходит после загрузки пользователем видеофайла на странице (рисунок 3.3). Форма страницы содержит вертикальное меню и форму для загрузки файла.

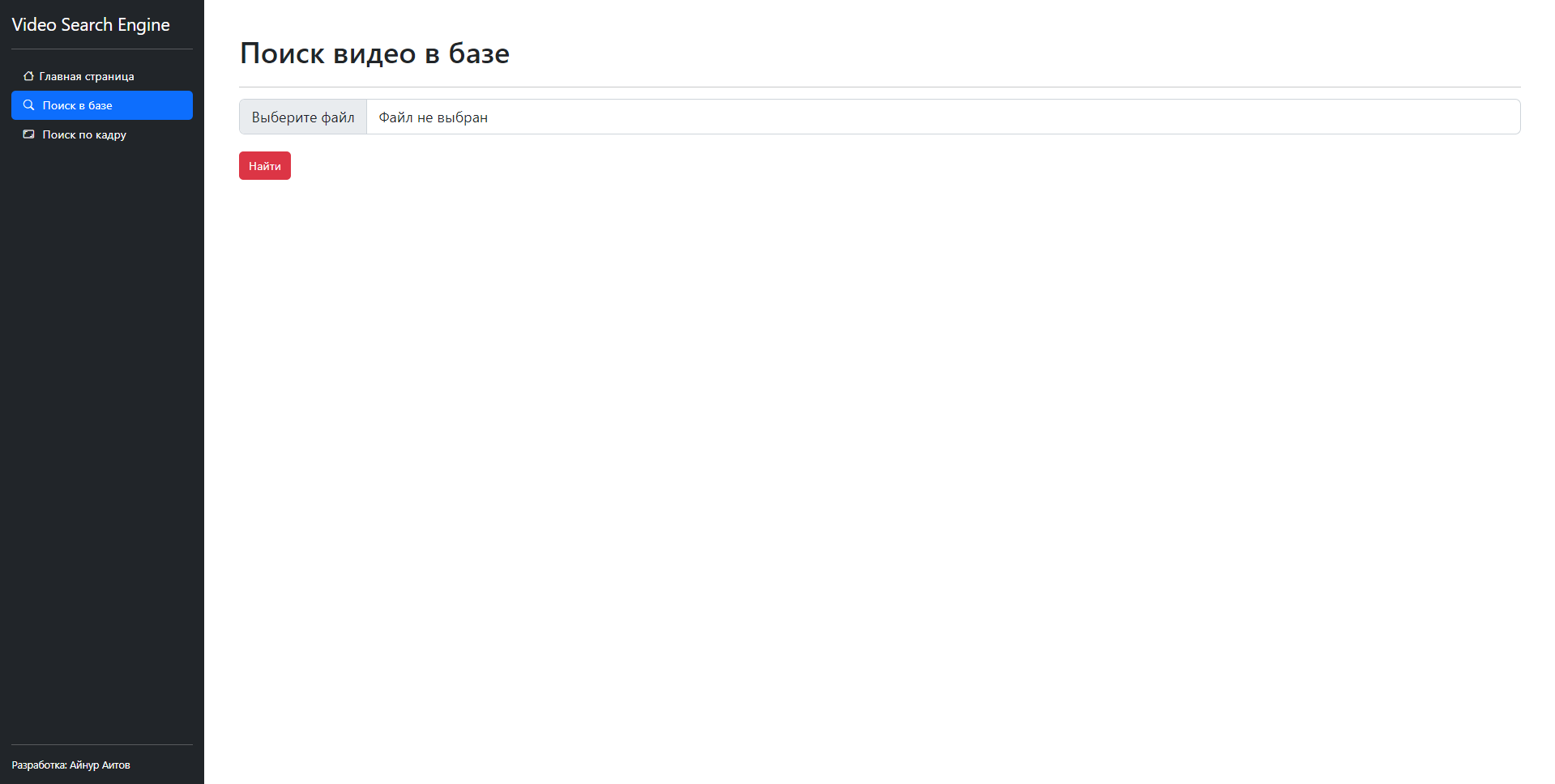


Рисунок 3.3 – Экранная форма «Поиск видео в базе»

При использовании мобильных устройств страница так же адаптируется под экран пользователя (рисунок 3.4).

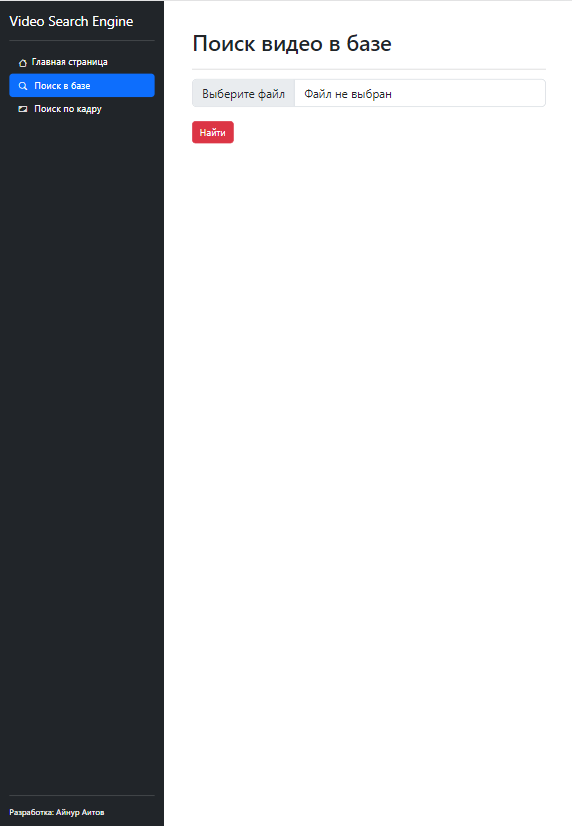


Рисунок 3.4 – Экранная форма «Поиск видео в базе» на мобильных устройствах

После успешной загрузки информационная система извлекает все ключевые кадры с видеофайла с временными метками кадров и сохраняет их в отдельную папку в разрешении 8 на 8. После, извлекает хэши ключевых кадров и удаляет извлеченные кадры. Далее происходит формирование индексов с последующим занесением их в базу данных. Общие индексы видеофайлов формируются и возвращаются пользователю в специальной форме.

Результат работы алгоритма можно увидеть на странице «Результат поиска в базе» (рисунок 3.5). В плеере красными метками отображаются временные метки общих границ сцен.

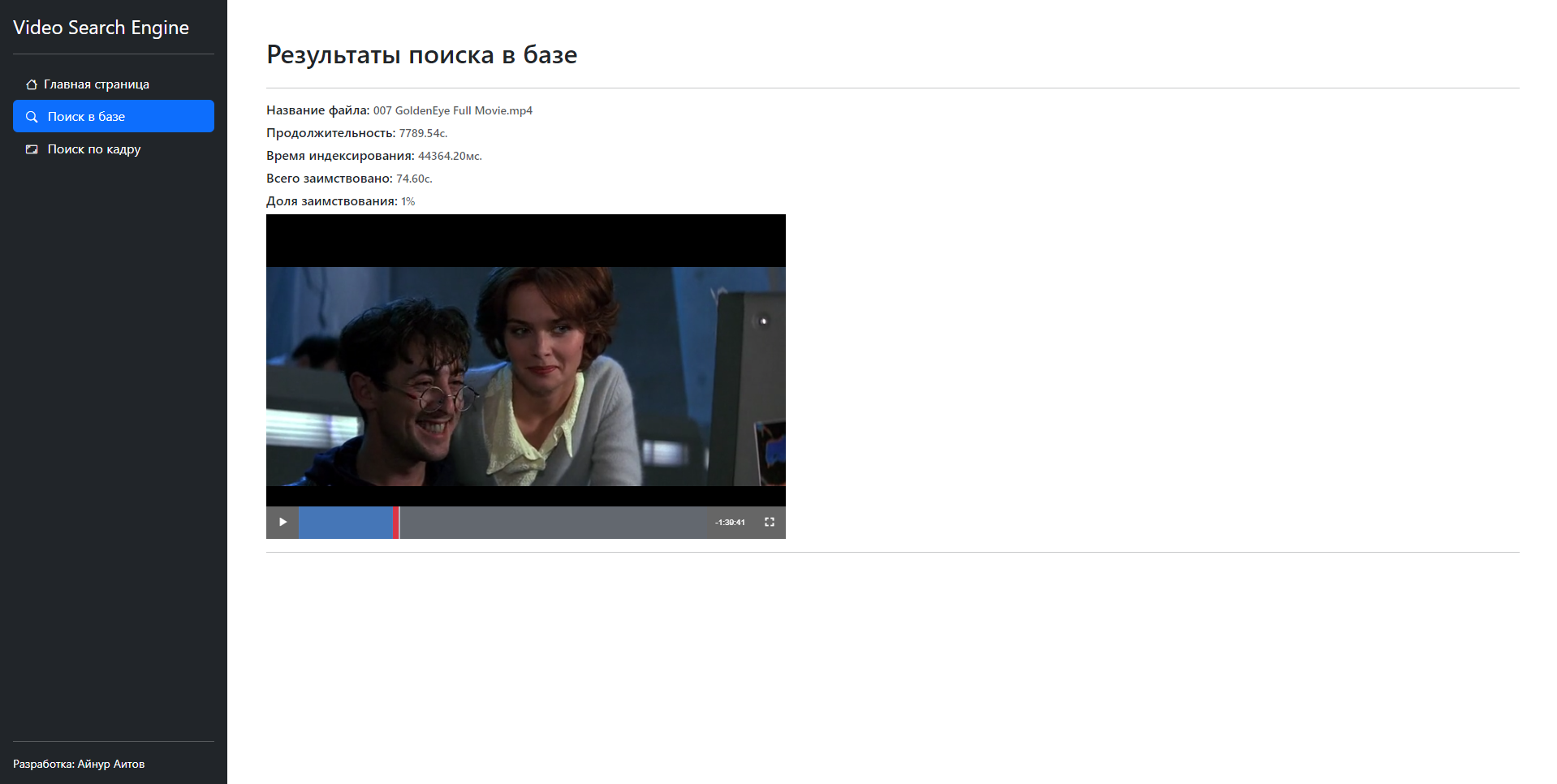


Рисунок 3.5 – Экранная форма с результатами поиска заимствований

Так же в приложении реализована возможность искать видеофайлы по кадру. При реализации данной возможности, информационная система сравнивает только перцептивные хэши. Результат поиска кадра (рисунок 3.6) можно увидеть на странице «Результаты поиска по кадру» (рисунок 3.7). В плеере красной меткой отображается место нахождения кадра.



Рисунок 3.6 – Использованный для поиска кадр

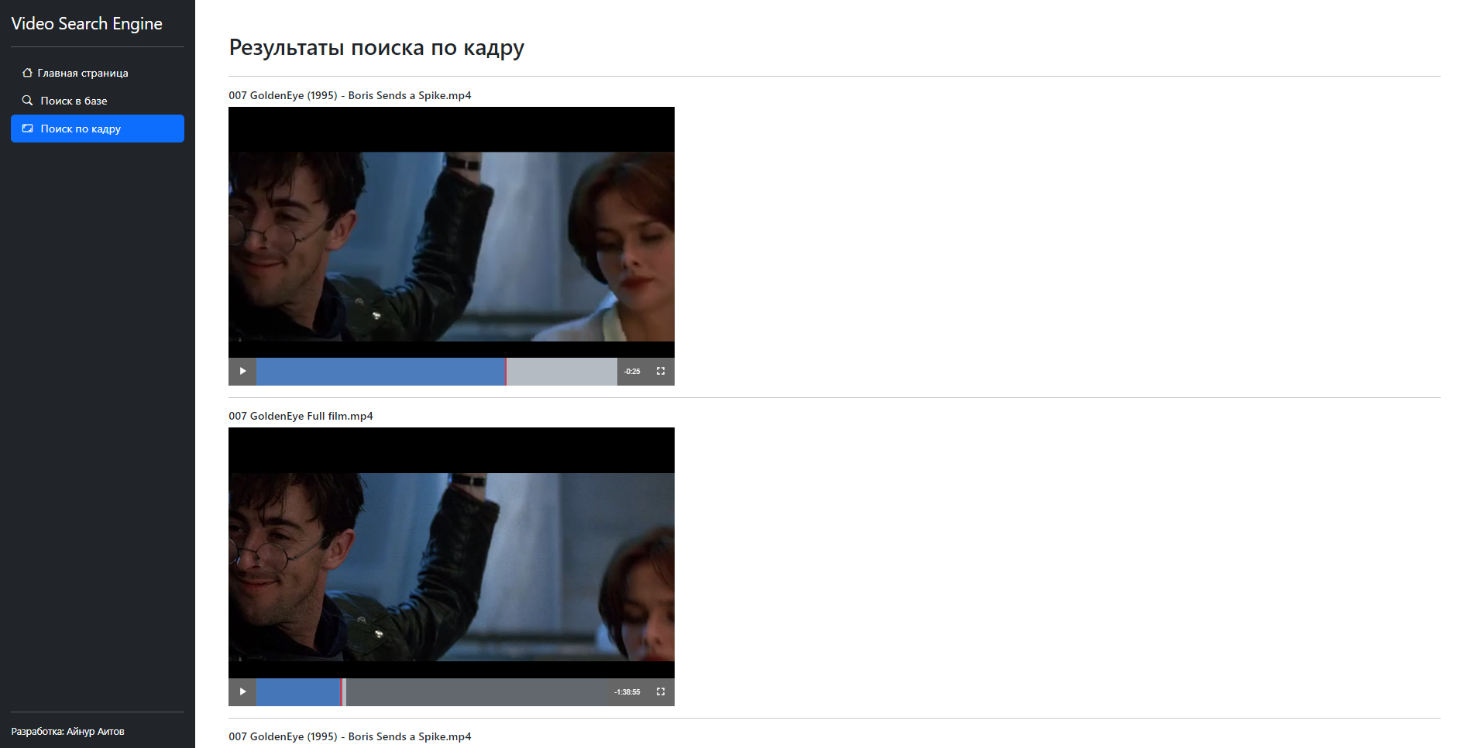


Рисунок 3.7 – Экранная форма поиска видеофайла по кадру

* 1. Описание процесса тестирования и сбора доказательной базы

Тестирование разработанных методов индексирования видеофайлов производится на задаче поиска схожих сцен в видеопотоке по образцу. Были использованы наборы из видеофайлов. Тестовый набор был разбит на 4 группы, характеристики которых перечислены в таблице 3.1.

Таблица 3.1 – Описание тестовой базы видеофайлов

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер группы** | **Количество видеофайлов** | **Размер (МБ)** | **Описание** |
| 1 | 20 | 1190 | Случайным образом  выбранные 20 видео.  Заимствований нет, используются  для проверки стабильности  алгоритма. |
| 2 | 10 | 2262 | Видеозаписи по запросу «007 Золотой глаз». Есть дублирующие видеозаписи. Присутствует сам видеофайл |
| 3 | 10 | 1720 | Запрос «Книга Бобы Фетта». Большое количество спецэффектов, сложно искать дубликаты вручную. |
| 4 | 10 | 75 | Дублирующие видеозаписи с различной частотой кадров. Различаются форматы хранения и разрешение видеопотока. Присутствует водяной знав в видеопотоке. |
| 5 | 10 | 611 | Видеозаписи с небольшим количеством смен сцен, с плавными переходам. Есть дубликаты, перекодированные с другой частотой кадров. |

Конфигурация компьютера, на котором проводились тесты, представлена ниже:

* Процессор: Intel Xeon E5-2640 v3 8x3400МГц.
* Память: 16Gb DDR4 1866 МГц (в двухканальном режиме).
* ОС: Windows 10 x64.
  1. Интерпретация результатов и оценка эффективности

Для оценки времени работы алгоритма поиска дубликатов был проведён поиск дубликатов в базе видео методом полного перебора. То есть, производилось сравнение каждого фильма с каждым (меньшее по длине видео искалось в большем) за исключением сравнения с самим собой. Таким образом, для базы из *N* фильмов выполнялось *N(N - 1) / 2* сравнения. Время работы алгоритма приведено в таблице 2.4.

Таблица 3.2 – Скорость построения индексов

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер группы** | **Общая продолжительность видеофайлов (сек)** | **Общее время**  **построения индексов (мс)** | **Средняя скорость индексации одной секунды видеофайла (мс)** |
| 1 | 7898 | 110153 | 13,943 |
| 2 | 25337 | 135863 | 5,36 |
| 3 | 19672 | 98713 | 5,01 |
| 4 | 826 | 6059 | 7,33 |
| 5 | 1952 | 32494 | 16,64 |
| Итого | 47787 | 383282 | 48,283 |
| Среднее | 9557,4 | 76656,4 | 9,6566 |

Таблица 3.3 – Достоверность индексирования

|  |  |
| --- | --- |
| **Номер группы** | **Средняя достоверность** |
| 1 | 100% |
| 2 | 90,2% |
| 3 | 87,5% |
| 4 | 77,6% |
| 5 | 48,3% |
| Среднее | 81% |

Средняя достоверность индексирования для 4 группы составил 81,5 процента. Наихудший показатель в 79% показал видеоролик с добавлением водяного знака. При изменении значения точности расстоянии Хэмминга в задаче сравнения индексов с 2 до 3, погрешность повышается до 86 процента. Однако увеличение данного значения может привезти к ложным срабатываниям.

Пятая группа показала наихудший результат среди тестов. Средняя достоверность индексирования составила 48 процентов. Наихудший результат 24%. Низкая достоверность обусловлена тем, что в данных видеороликах используются плавные переходы между сценами и сцены плохо отличаются друг от друга.

В ходе работы, информационная система активно взаимодействует с базой данных. Оценена зависимость времени работы программы с базой данных, в зависимости от количества индексов (таблица 2.4). Нагляднее зависимость можно наблюдать в рисунке 2.1.

Таблица 3.4 – Зависимость времени запросов к базе данных от количества индексов

|  |  |
| --- | --- |
| **Количество индексов** | **Время запросов к базе данных (мс)** |
| 107 | 73 |
| 5041 | 86 |
| 38993 | 174 |
| 56865 | 251 |
| 85328 | 272 |
| 112469 | 334 |
| 209326 | 406 |

Получается, что на каждые 75 тысяч индексов время запросов увеличивается на 200 миллисекунд. По прогнозу, когда количество индексов достигнет миллиона, время запросов к базе данных увеличится до 2666 миллисекунд.



Рисунок 3.8 – Зависимость времени запросов к базе данных от количества индексов

Выводы по разделу

В третьем разделе обоснованы средства реализации и описана программная реализация.

Был описан процесс тестирования разработанного алгоритма и сбора доказательной базы. В конце приводятся данные об эффективности проекта.

# **Заключение**

В ходе работы над выпускной квалификационной работой были решены все поставленные задачи.

Был выполнен анализ предметной области исследования. Определена научная проблема. Подтверждена актуальность научной проблемы. Сформулирована гипотеза.

Проанализированы особенности систем видеокопрессии и существующие алгоритмы обработки ключевых кадров видеопотока для определения их схожести. Разработан алгоритм создания уникальных подписей ключевых кадров. Была построена модель TO-BE информационной системы с использованием методологии DFD, которая отражает движения потоков данных. Спроектирована база данных информационной системы и обоснован выбор системы управления базами данных.

Была реализована информационная система с реализацией разработанного алгоритма и оценена ее эффективность.

Таким образов разработанный алгоритм создания уникальных подписей ключевых кадров видеопоследовательности эффективно решает задачу поиска заимствований видеофрагментов.

# **Список использованных источников**

1. Российский рынок VoD 2021 // ТМТ Консалтинг URL: https://www.tadviser.ru/images/c/cd/ТМТ-рейтинг-VoD-2021-2.pdf (дата обращения: 21.04.2022).
2. MUSO DISCOVER Q1 2022 DIGITAL PIRACY DATA INSIGHTS // MUSO - Protect, Discover, Connect URL: https://www.muso.com/magazine/muso-discover-q1-2022-digital-piracy-data-insights (дата обращения: 21.04.2022).
3. Паршин, А. Е. Алгоритм поиска дубликатов в базе видеопоследовательностей на основе сопоставления иерархии смен сцен / А. Е. Паршин, И. В. Глазистов // Новые информационные технологии в автоматизированных системах. – 2009. – № 12. – С. 51-61.
4. Никитин, И. К. Методы поиска нечетких дубликатов видео / И. К. Никитин // Нейрокомпьютеры: разработка, применение. – 2015. – № 9. – С. 60-66.
5. Law-To. J., Buisson O., Gouet-Brunet V., Boujemaa N. Robust voting algorithm based on labels of behavior for video copy detection // Proceedings of the 14th ACM International Conference on Multimedia. – Santa Barbara: Association for Computing Machinery, 2006. – P. 835-844.
6. Vaiapury K., Atrey P. K., Kankanhalli M. S., Ramakrishnan K. Non-identical Duplicate Video Detection Using The SIFT method // Visual Information Engineering, 2006. VIE 2006. IET International Conference on. – 2006. – Sept. – P. 537-542.
7. Douze M., Gaidon A., Jegou H., Marszalek M., Schmid C. Inria-Learars video copy detection system // TREVCID. – 2008.
8. Dong W., Wang Z., Charikar M., Li. K. Efficiently matching sets of features with random histograms // MM. – ACM, 2008. – P. 179-188.
9. Hua X.-S., Chen X., Zhang H.-J. Robust video signature based on ordinal measure // 2004 International Conference on Image Processing. – Singapore: IEEE, 2004. – P. 685-688.
10. Wang Y., Belkhatir M., Tahayna B. Near-Duplicate Video Retrieval Based on Clustering by Multiple Sequence Alignment // Proceedings of the 20th ACM international conference on Multimedia. – New York: ACM, 2012. – P. 941-944.
11. Лузин, В. И. Основы формирования, передачи и приема цифровой информации : учебное пособие / В. И. Лузин, Н. П. Никитин, В. И. Гадзиковский ; науч. ред. В. И. Гадзиковский. - Москва : ООО «СОЛОН-Пресс», 2020. – 316 с.
12. Практическое использование видеокомпрессии в телевидении // Общественно-образовательный портал URL: https://sneg5.com/nauka/tehnika-i-tehnologii/mpeg.html (дата обращения: 21.04.2022).
13. Байгарова Н.С. Методы индексирования и поиска изображений и видеоданных на основании визуального содержания / Ю.А. Бухштаб, А.А. Горный, Н.Н. Евтеева, В.Ю. Лялин, А.В. Монастырский, А.Ю. Стрелков // Электронные библиотеки: перспективные методы и технологии, электронные коллекции. – Протвино: Институт прикладной математики им. М. В. Келдыша РАН, 2002. – С. 263-273.
14. Ельчанинов В.С. Поиск подобных изображений в базе данных индексированных видео / В.С. Ельчанинов, Д.Н. Лясин // Электронный научно-практический журнал «Современные научные исследования и инновации». – 2014. – №5
15. Построение признаков и сравнение изображений: глобальные признаки. Лекции от Яндекса // Хабр URL: https://habr.com/ru/company/yandex/blog/255627/ (дата обращения: 21.04.2022).
16. Рудаков, И. В. Исследование перцептивных хеш-функций изображений / И. В. Рудаков, И. М. Васютович // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. – 2015. – № 8. – С. 269-280.
17. Огневой, Г. Д. Методы и алгоритмы поиска изображений / Г. Д. Огневой // Информационные технологии в образовании, науке и производстве. – 2014.
18. Расстояние Хэмминга // Викиконспекты Университета ИТМО URL: https://neerc.ifmo.ru/wiki/index.php?title=Расстояние\_Хэмминга (дата обращения: 21.04.2022).
19. Сравнение современных СУБД | Заметки программиста [Электронный ресурс]. – Режим доступа: http://drach.pro/blog/hi-tech/item/145-db-comparison
20. Node.js в действии / Кантелон М., Хартер М., Головайчук T., Райлих Н.. — СПб: Питер, 2015. — 448с.
21. Система ejudge - EjudgeWiki [Электронный ресурс]. – Режим доступа: https://ejudge.ru/wiki/index.php/Система\_ejudge
22. Bootstrap. Документация на русском языке [Электронный ресурс]. – Режим доступа: https://bootstrap-4.ru/
23. Влацкая, И.В. Проектирование и реализация прикладного программного обеспечения: учебное пособие / И.В. Влацкая, Н.А. Заельская, Н.С. Надточий; Оренбургский государственный университет. – Оренбург: ОГУ, 2018. – 118 с. Режим доступа: http://www.iprbookshop.ru/.
24. ГОСТ Р 43.0.11-2014 Информационное обеспечение техники и операторской деятельности. Базы данных в технической деятельности. – Введ. 01.09.2015. – М.: Изд-во стандартов, 2015. – 24 с.

# Приложение А

Исходный код экранной формы «Главная страница»

...

Исходный элемента формы «head»

...

Исходный элемента формы «footer»

...

Исходный элемента формы «left-menu»

...

Исходный элемента формы «menu»

...

Исходный элемента формы «scripts»

...

Исходный код экранной формы «Поиск по базе»

...

Исходный код экранной формы «Результаты поиска в базе»

...

Исходный код экранной формы «Поиск по кадру»

...

Исходный код генерации экранной формы «Результаты поиска по кадру»

...

Исходный код сервера приложения

...

Приложение Б

Выпускная квалификационная работа выполнена мною самостоятельно. Используемые в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них. Список использованных источников наименований.

Один печатный экземпляр и один электронный экземпляр на CD сдан на кафедру.

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

\_\_\_\_\_\_\_\_\_\_\_\_\_ / /

(подпись) (Ф.И.О.)