

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет **ФИЗИЧЕСКИЙ**

Кафедра Автоматизации физико-технических исследований

Направление подготовки **03.03.02 ФИЗИКА**

Образовательная программа: **БАКАЛАВРИАТ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Мустафин Владимир Михайлович

Тема работы:

“Разработка алгоритма поиска специфических сигнатур и их переходов в
видеопотоке”

«К защите допущена»

Заведующий кафедрой:

Лысаков Константин Федорович

Научный руководитель

Таранцев Игорь Геннадьевич

К. Т. Н.

ученая степень, звание

Без степени

должность, место работы:

должность, место работы
Лаборатория №13 ИАиЭ СО РАН.

Лысаков К.Ф./.....

(фамилия И., О.) / (подпись, МП)

Таранцев И.Г./.....

(фамилия И., О.) / (подпись, МП)

«.....».....20...г.

«.....».....20...г.

Дата защиты: «.....».....20...г.

Новосибирск, 2021

Оглавление

Введение.....	3
Глава 1. Вычисление сигнатур.....	4
1.1 Использование цветовой и яркостной гистограмм в качестве сигнатур	4
1.2 Сигнатура на основе порядковой меры.....	6
1.3 Сигнатура на основе использования быстрого преобразования Фурье	6
1.4 Итоги.....	8
Глава 2. Анализ видеороликов	8
2.1 Разбиение на сцены	8
2.2 Выделение ключевых сигнатур.....	12
2.3 Особенности вычисления сигнатур с помощью БПФ.....	15
Глава 3. Разработка алгоритма.....	16
3.1 Разбиение ролика на сегменты.....	20
3.2 Анализ входящего видеопотока	23
3.2.1 Определение границ окна поиска в новых сегментах.....	23
3.2.2 Корректировка окна поиска.....	26
3.2.3 Работа с активным сегментом	29
3.2.4 Обработка результатов поиска в базе данных	31
Глава 4. Оценка сложности алгоритма.....	33
Глава 5. Тестирование и реализация	34
Заключение	36
Список литературы	37

Введение

В современном мире огромное количество телеканалов – эфирных, спутниковых и интернет. Мониторинг контента на любом из них является важной и актуальной задачей. Особенно актуальна эта задача в сфере рекламного бизнеса, где важно получить ответ на такие вопросы как: «Была ли показана моя реклама?», «Была ли показана моя реклама в полном объеме?», «В какое время была показана реклама?». Не менее важным является вопрос соблюдения авторских прав.

Для решения задач мониторинга контента необходимо сравнивать изображения из видеопотока прямого вещания с набором заранее заданных видеороликов. При применении метода прямого сравнения изображений (попиксельное сравнение), производительности современных компьютеров достаточно примерно для $120 \frac{\text{сравнений}}{\text{кадр}}$. Это означает, что за один кадр входящего видеопотока, возможно произвести поиск по видеороликам суммарной длиной не более шести секунд. В рамках данной дипломной работы, необходимо разработать алгоритм обрабатывающий базу размером 1000 видеороликов или $1.5 * 10^6$ кадров в темпе поступления данных.

Существует возможность построить некоторый слепок изображения – его сигнатуру. При этом схожесть сигнатур коррелирует со схожестью изображений. Использование сигнатур позволяет сократить размер хранимых данных и уменьшить время сравнения изображений. Существует исследование [3], показывающее достаточную точность сравнения сигнатур для сравнения изображений. Производительности современных компьютеров достаточно для выполнения примерно $1.5 * 10^4 \frac{\text{сравнений}}{\text{кадр}}$ (для сигнатур размером 128 слов). Соответственно, для поиска по базе данных $1.5 * 10^6$ сигнатур необходимо увеличение производительности алгоритма сравнения минимум на 2 порядка.

Известно, что в видеороликах соседние кадры похожи друг на друга. Используя эту избыточность данных, можно попытаться уменьшить количество сравнений сигнатур.

Целью текущей дипломной работы является уменьшение количества сравнений за счет анализа видеороликов и выделения в них ключевых сигнатур. Количество сравнений должно быть достаточным для работы с потоком видео реального времени и обработки базы с характерным размером 10^6 сигнатур.

Для достижения поставленной цели необходимо решить следующие задачи:

- Провести анализ предметной области
- Разработать алгоритм выделения ключевых сигнатур
- Разработать алгоритм поиска видеороликов по ключевым сигнатурам
- Реализовать и протестировать разработанные алгоритмы

Глава 1. Вычисление сигнатур

1.1 Использование цветовой и яркостной гистограмм в качестве сигнатур

Использование яркостной и цветовой гистограмм, для обработки, а также подписи изображений является одним из самых старых и простых методов [5]. Так же, построенную яркостную гистограмму можно использовать в качестве сигнатуры изображения. Алгоритм состоит в том, чтобы отобразить цвета изображения M в дискретное цветовое пространство, содержащее n цветов, тогда цветовая гистограмма $H(M) = (h_{c1}, h_{c2}, \dots, h_{cn})$ это вектор, где h_{cj} является числом точек изображения M с цветом c_j . Если N число точек на изображении M , тогда $\sum_{i=1}^n h_{ci} = N$. Для определения расстояния между гистограммами H и I используются различные функции:

$$d_{L1}(H, I) = \sum_{l=1}^n |h_{cl} - i_{cl}|$$

$$d_{L2} = \sqrt{\sum_{l=1}^n (h_{cl} - i_{cl})^2}$$

$$d_{\infty} = \max_{1 \leq l \leq n} |h_{cl} - i_{cl}|$$

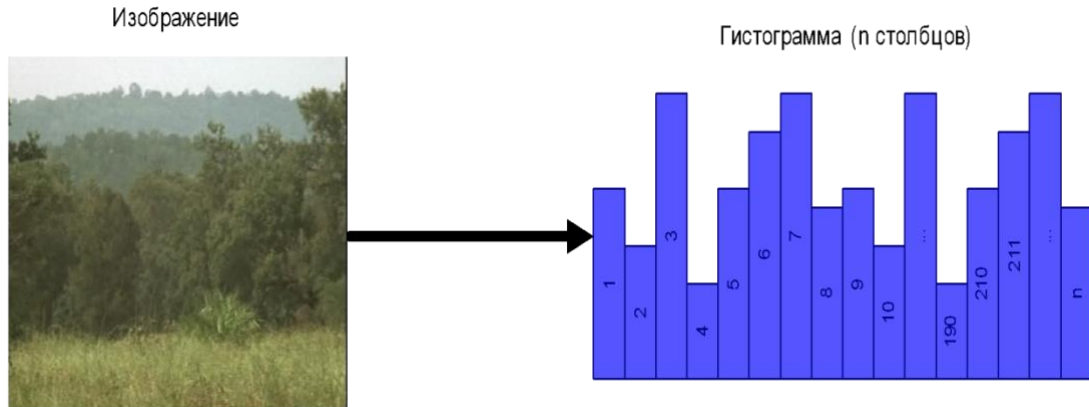


Рис. 1. Преобразование изображение в цветно-яркостную гистограмму

Цветовая гистограмма достаточно устойчива к шуму в изображении, а так же к поворотам и переносам. Но у неё есть несколько недостатков. Один из них, это то, что гистограмма описывает только распределение цвета или яркости на изображении, не учитывая при этом какую-либо пространственную информацию. На рисунке 2 представлены изображения, имеющие одинаковую гистограмму.

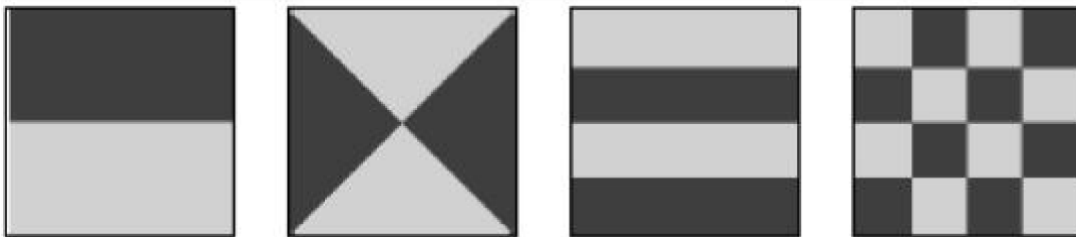


Рис.2. Пример изображений с одинаковой гистограммой

Также, если в кадре есть небольшой участок визуальной важной информации, при использовании данного метода расчета сигнатур, данный участок может не внести существенного вклада в гистограмму и сравнение будет некорректным. Данный алгоритм в основном используется для разбиения последовательности изображения на похожие фрагменты, однако для нашей задачи он не подходит.

1.2 Сигнатура на основе порядковой меры

Алгоритм на основе порядковой меры [6] является неким улучшение гистограммного метода. В данном алгоритме сначала изображение разбивается на N областей. Для каждой из этих областей считается среднее значение яркости. Далее каждой области присваивается порядковый номер от нуля до N , в зависимости величины средней яркости. Вектор, компонентами которого являются эти числа, и есть сигнатура изображения. Основными преимуществами данной сигнатуры являются высокая скорость её вычисления, устойчивость к изменению яркости, а также некоторый учёт пространственной характеристики изображения. Однако существует большое количество реальных изображений, которые имеют одинаковую порядковую меру, но совсем разное содержание. Это основная причина, по которой данный алгоритм не может быть использован в данной работе.

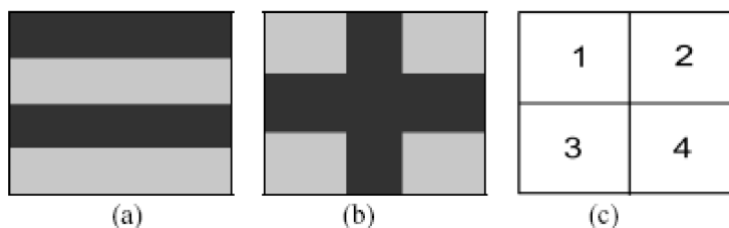


Рис. 3. Разбиение изображение на области

1.3 Сигнатура на основе использования быстрого преобразования Фурье

Как дискретное преобразование Фурье (ДПФ), так и дискретно-косинусное преобразование (ДКП) широко используются при обработке видеосигнала. Известно, что большая часть информации об изображении содержится в низкочастотных коэффициентах ДПФ и ДКП. Поэтому эти преобразование позволяют уменьшить размерность при обработке изображения.

В алгоритме, рассмотренном в [1], используется трёхмерное ДКП для учёта пространственно-временных характеристик видео клипа. Сначала производится преобразование видео последовательности к единому пространственно-временному формату, в результате которого получается

последовательность из 64 изображений, размерами 32x32 пикселя. Далее из них составляется матрица 32x32x64, над которой выполняется трёхмерное дискретно-косинусное преобразование. Для получения сигнатуры используются 64 низкочастотных коэффициентов ДКП ($x_i, i = 0 \dots 63$) из левой верхней подматрицы размером 4x4x4. Далее производится сортировка и квантование коэффициентов. Все коэффициенты, которые больше медианы объявляются равными 1, меньшие же коэффициенты – 0. Медиана считается следующим образом $m = \frac{C_{32} + C_{33}}{2}$, где $C_i, i = 0 \dots 63$ - отсортированный список низкочастотных коэффициентов ДКП. Одно битное квантование добавляет устойчивости к небольшим различиям в сигнатуре. Для определения дистанции между сигнатурами в рассмотренной работе используется расстояние Хэмминга — мера различия объектов одинаковой размерности ($d_H(X_i, X_j) = \sum_{s=1}^p |x_i^s - x_j^s|$, где X_i, X_j — объекты размерности p). Результаты экспериментов, проведённых в рассмотренной работе, показывают, что полученная таким образом сигнатура является устойчивой ко многим типам шумов (размывание, изменение яркости, небольшие геометрические изменения, артефакты сжатия), а также обладает уникальностью, т.е. она различна у двух разных изображений.

В работе [2] используется подобный метод построения сигнатуры. Существуют несколько отличий. Во-первых, не учитываются временные характеристики, т.е. ДКП выполняется над двумерной матрицей, представляющей собой обрабатываемое изображение. Во-вторых, квантование происходит относительно среднего значения подматрицы низкочастотных коэффициентов ДКП, т.е. $m = \frac{\sum_{i=0}^{n^2} x_i}{n^2}$, где $n \times n$ — размерность подматрицы (в работе $n = 5 \dots 8$). В качестве функции сравнения, так же, как и в [1], использовалось расстояние Хэмминга.

При использовании данного алгоритма в выпускной работе на соискание степени магистра, выпускника кафедры автоматизации физико-технических исследований [3], в ходе работы, было рекомендовано использовать быстрое

преобразование Фурье. В реализации алгоритма, используемого в текущей дипломной работе, в качестве сигнатуры были использованы коэффициенты низкочастотной подматрицы без выполнения их квантования. В этом случае в качестве функции сравнения, в работе [3], было выбрано Эвклидово расстояние между двумя одномерными векторами размерности

$$n^2: d_e(X_i, X_j) = \sqrt{\sum_{s=1}^{n^2} (x_i^s - x_j^s)^2}.$$

1.4 Итоги

Алгоритм использования цветовой и яркостной гистограмм в качестве сигнатур, не учитывает какую-либо пространственную информацию, а также небольшие участки визуальной важной информации не вносят существенный вклад в сигнатуру, поэтому от данного алгоритма было решено отказаться.

Алгоритм расчёта сигнатур на основе пространственной меры, не обладает достаточной уникальностью, т.к. реальные изображения, могут иметь одинаковую пространственную меру, но совсем разное содержимое, поэтому от данного алгоритма было решено отказаться.

Из рассмотренных алгоритмов, наиболее устойчивым к многим типам шумов, таким как, размытие, изменение яркости, небольшие геометрические изменения, артефакты сжатия и т.п., является алгоритм вычисления сигнатур на основе быстрого преобразования Фурье. Так же, данный алгоритм обладает высоким показателем уникальности [1]. Данный алгоритм был выбран в качестве основного в текущей работе.

Глава 2. Анализ видеороликов

2.1 Разбиение на сцены

В современном мире существует бесчисленное количество различных видеороликов. В каждом видеоролике есть так называемые сцены. Как правило, сцена характерна тем, что кадры внутри одной сцены обладают высокой степенью схожести, а между сценами кадры сильно отличаются. В

данной главе степень схожести будет измеряться путем сравнения сигнатур, построенных на основе кадров видеоролика. С помощью измерения схожести сигнатур кадров видеоролика можно попытаться разделить видеоролик на сцены.

Для исследования были взяты несколько видеороликов, которые отличаются по своему содержанию. Первое видео было взято по типу



Рис. 4. Пример соседних кадров интервью

“интервью”, для такого видео характерно большое количество кадров, похожих друг на друга (рис. 4), а смена сцены характеризуется резкой сменой ракурса съемки (рис. 5)



Рис. 5. Соседние кадры интервью при резкой смене ракурса

Что бы наглядно продемонстрировать большое количество похожих кадров, было произведено покадровое сравнение всего ролика. А именно, брались соседние кадры и считалась разница сигнатур. На рис. 6, изображен график отличия соседних сигнатур.



Рис. 6. График результатов сравнения соседних кадров для первого видео

Как видно из графика выше, разница между соседними кадрами в основном не превышает абсолютной величины, равной 0.05. На рис. 6 красным квадратом выделены особые случаи, с величиной больше 0.4, именно на этих кадрах и происходит смена сцены. Таким образом, появляется возможность разделить видео на сцены, анализируя схожесть сигнатур соседних кадров – если разница больше порога, то это новая сцена.

В качестве второго видео, было взято вступление к новостному выпуску.



Рис. 7. Раскадровка второго видео

В этом ролике сцены меняются гораздо чаще чем в предыдущем, а также внутри сцен камера движется относительно объектов, а объекты не статичны. В итоге это дает низкую степень схожести соседних кадров. На рис 8 изображен результат сравнений соседних сигнатур.

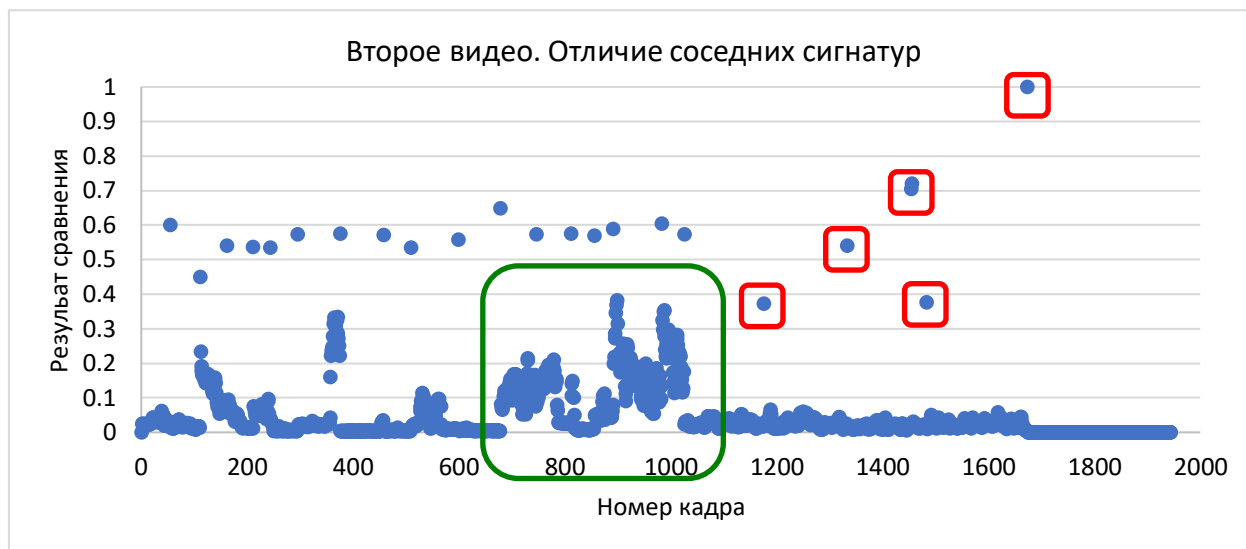


Рис. 8. График сравнения соседних сигнатур. Второе видео

Как видно из представленного графика, разделение на сцены данного видео по какому-то одному фиксированному порогу не так очевидно. Например, в промежутке кадров $[1100; 1600]$, степень схожести сигнатур в основном колеблется в диапазоне значений от 0.0009 до 0.05, а красными квадратами выбраны особые случаи отличия соседних сигнатур, по которым можно детектировать новые сцены. В этом промежутке кадров четкое разделение куска видео на сегменты происходит при пороге 0.2. Однако стоит обратить внимание на промежуток сигнатур, выделенных зеленым квадратом. В этой части графика прослеживается более сложная кластеризация результатов сравнения сигнатур. Задача выбора порога для разделения на сцены становится уже не такой тривиальной. Например, если порог равен 0.5, то разделение ролика на сцены окажется просто не корректным.

Как видно из примеров выше, разбиение ролика на сцены путем сравнения соседних сигнатур кадров с фиксированным порогом в каких-то случаях оказывается корректным. Однако в других случаях, какой бы мы порог не выбрали, разбиение на сцены будет неправильным.

2.2 Выделение ключевых сигнатур

Изначально, идея разбиения на сцены была взята как способ сократить количество сравнений. Например, все ролики разбиваются на сцены, где кадры внутри сцены обладают высокой степенью схожести, как на рисунках 4 и 5. Далее, среди похожих сигнатур выбирается ключевая сигнатура, которая будет являться представителем всей сцены. При этом, если входная сигнатура (построенная на основе кадра входного потока) отличается от ключевой сигнатуры достаточно сильно, то все сигнатуры соответствующей сцены точно не совпадают со входной сигнатурой. Таким образом, сохраняя в базе поиска не все сигнатуры видеоролика, а только ключевые и выполняя поиск по ним, появляется возможность резко сократить количество сравнений. Чем больше кадров в одной сцене, тем меньше количество ключевых сигнатур в базе данных. Однако остается вопрос, как наиболее правильным образом разделить видеоролик на некие сегменты, чтобы кадры внутри обладали высокой степенью схожести, а выделение ключевой сигнатуры стало тривиальной задачей. Ведь как видно из видеоролика №2, предложенный способ сравнения соседних кадров и разбиение по порогу не дает желаемых результатов.

Далее предлагается использовать другой подход при разбиении ролика на сегменты. В качестве ключевой сигнатуры, выбирается первая сигнатура сегмента. Все последующие сигнатуры сравниваются с ключевой сигнатурой, и если результат сравнения превышает некоторый заданный порог, то формирование сегмента заканчивается, а новая сигнатура является началом нового сегмента. В результате, все сигнатуры в сегменте должны отличаться от ключевой не более заданного порога.

На рис. 9 представлен график результатов работы такого подхода. На отрицательных значениях оси ординат синими точками помечены начала новых сегментов. В качестве исследуемого ролика, представлен первый ролик из предыдущей главы (рисунки 4 и 5).

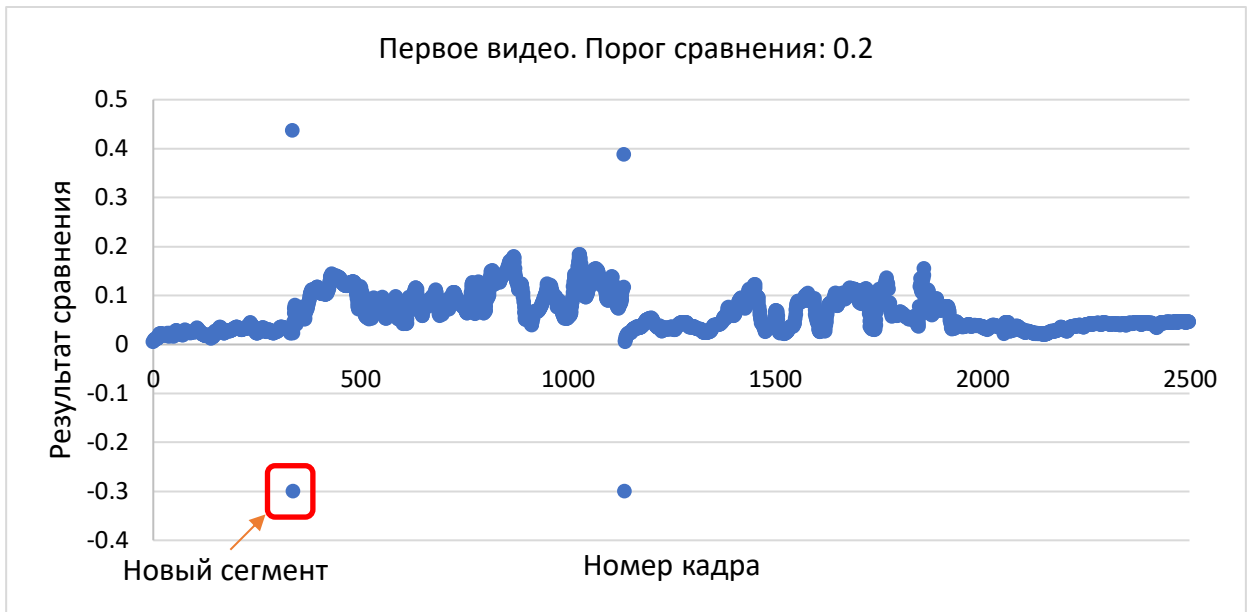


Рис. 9. График разделения на сегменты. Первое видео

Как видно из графика выше, разбиение ролика на сегменты таким способом совпадает с исходным разбиением ролика на сцены. Такое разбиение гарантирует, что внутри сегмента все сигнатуры не отличаются друг от друга больше, чем выбранный порог (на данном графике порог равен 0.2). После разбиения первого видео на сегменты, из 2600 кадров получается 2 сегмента и 2 ключевые сигнатуры, что позволяет при поиске сигнатуры сократить количество сравнений больше чем в 1000 раз.

Однако на более динамичных видео роликах не удастся добиться такого сильного уменьшения количества сравнений. На рис. 10 представлен результат работы разделения на сегменты способом описанном выше для второго видеоролика (рис. 7). На этом графике так же в отрицательных значениях оси ординат синими точками помечены начала новых сегментов.

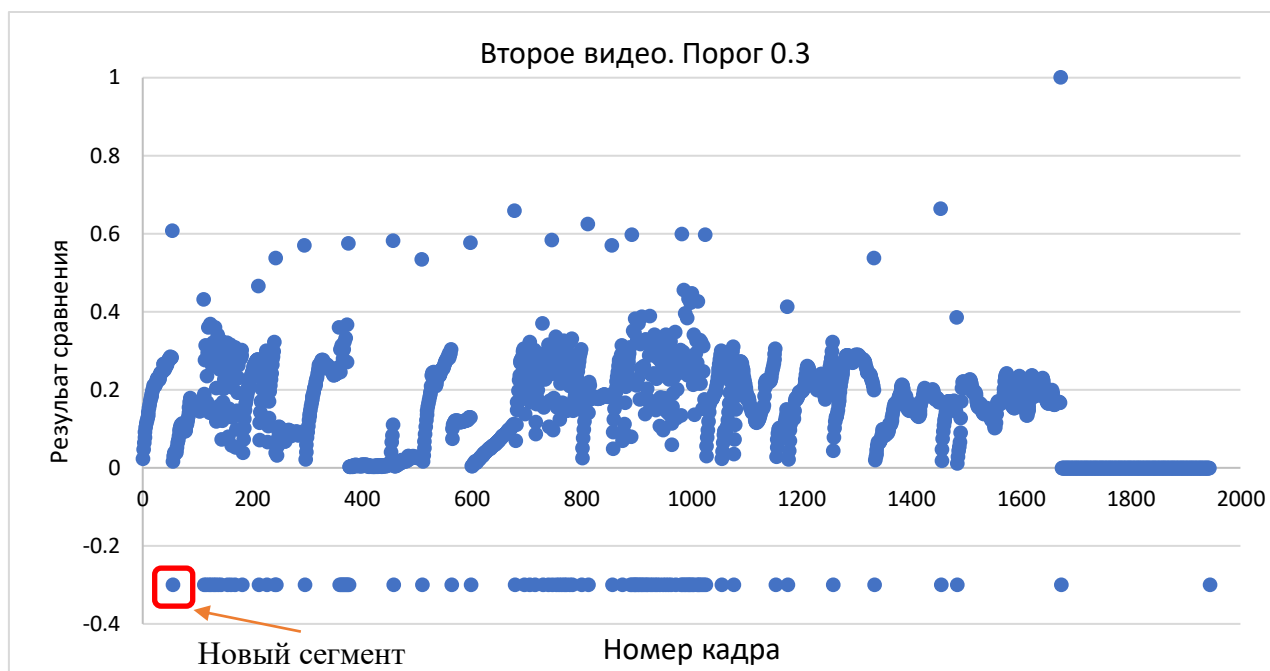


Рис. 10. График разделения на сегменты. Второе видео

В данном видео из 2000 сигнатур, с помощью представленного алгоритма, удалось сформировать 94 сегмента, что дает уменьшение количества сравнение при поиске по ключевым сигнатурам примерно в 21 раз.

Как можно заметить из сравнения графиков рис. 9 и 10, чем ролик более динамичный, тем большее количество сегментов формируется в результате работы предложенного алгоритма.

В качестве третьего ролика для исследования был выбран музыкальный клип. Для этого видеоролика характерно сильное изменение освещения от кадра к кадру (рис. 11), а также весь видеоролик состоит из крайне динамичных сцен. Визуально, характерное количество кадров в сцене можно оценить в 5-10 кадров.

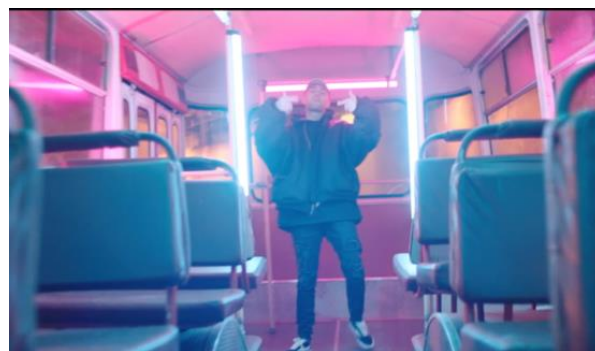
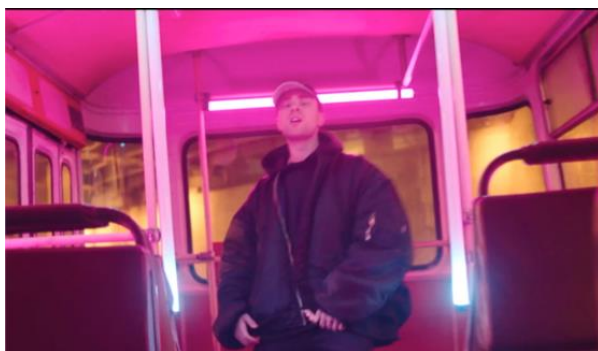


Рис. 11. Соседние кадры третьего видео

Как видно из рис. 12, такой видеоролик плохо разбивается на сегменты. В отрицательной части оси ординат точки, означающие начало нового сегмента, визуально образуют сплошную линию. А именно, на 2090 кадров видеоролика, было сформировано 462 сегмента. При этом, для обработки данного видеоролика, был выбран самый высокий порог смены сегмента, равный 0.4. Таким образом, количество сравнений удалось сократить только в 4 раза, что является неприемлемым в рамках нашей задачи.

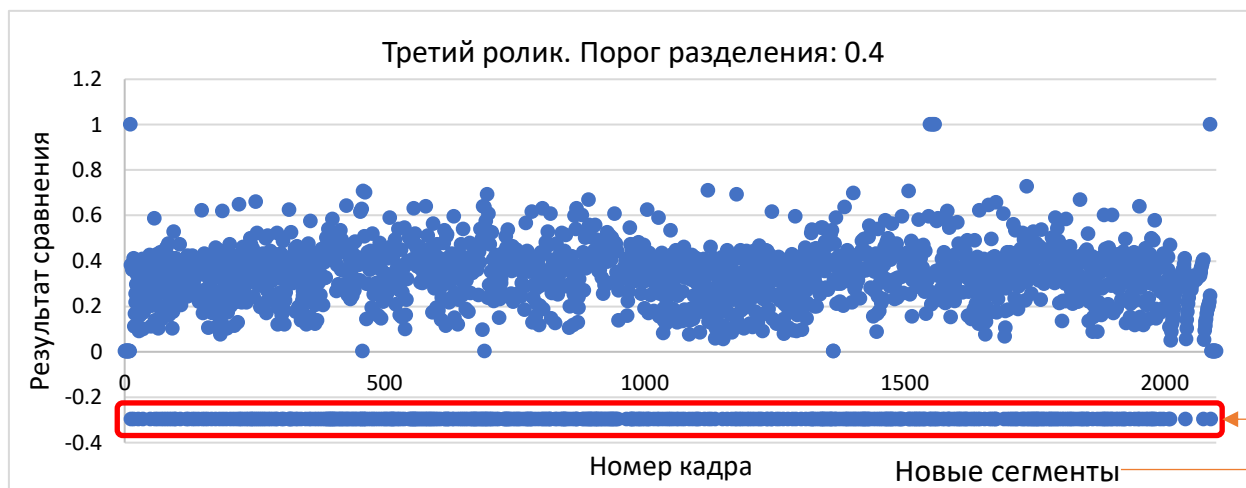


Рис. 12. График разделения на сегменты. Третье видео

На примере данного видео становится очевидной необходимость создания ограничения минимального размера сегмента. Такие видео сегменты нуждаются в особой постобработке. А именно, поскольку сигнатуры внутри одного сегмента сильно различаются, необходимо проводить дополнительное сравнение после обнаружения ключевой сигнатуры.

В текущей дипломной работе размер минимального видео сегмента составляет 25 кадров. В действительности, такое ограничение можно менять в зависимости от типа ролика. Так же, такое ограничение позволяет фиксировать уменьшение количества сравнений минимум в 25 раз.

2.3 Особенности вычисления сигнатур с помощью БПФ

В ходе проведения исследований было выявлено, что из практически одноцветных кадров рассчитывается нулевая сигнатура (все числа сигнатур равны нулю). В фильмах или в видеопотоке телевидения, довольно часто

переход между сценами и отдельными видеороликами выполняют через так называемый эффект перехода “Fade in” и “Fade out”. Самое главное, что этот эффект выполняет переход через одноцветные кадры (черные или белые). Эти кадры не несут в себе никакой полезной информации для распознавания роликов. Такие наборы кадров лишь увеличивают размеры базы и время поиска. В текущей дипломной работе было решено избавиться от таких кадров, т.е. игнорировать их при подготовке видеоролика, а также игнорировать их при анализе входного видеопотока прямого вещания. Поскольку для точного распознавания видеороликов необходимо сохранять порядок кадров в ролике, достаточно вместо таких сигнатур сохранять их количество.

Глава 3. Разработка алгоритма

Практически в любом видео существуют последовательности похожих кадров. Они могут быть разной длины и разной степени схожести. Можно объединять такие последовательности в сегменты и для каждого сегмента задавать одну «ключевую» сигнатуру. Если сравнивать сигнатуру входящего видеопотока не со всеми сигнатурами подряд, а только с ключевыми сигнатурами видео сегмента, то можно сократить количество сравнений в десятки или даже в сотни раз. Сегменты формируются путем расчета разницы между первой и последней сигнатурой. Если сигнатура очередного кадра отличается от сигнатуры первого кадра сегмента выше определенного порога, то с очередного кадра начинается новый сегмент.

Алгоритм обработки входного видеопотока представляет из себя конечный автомат (рис 13).

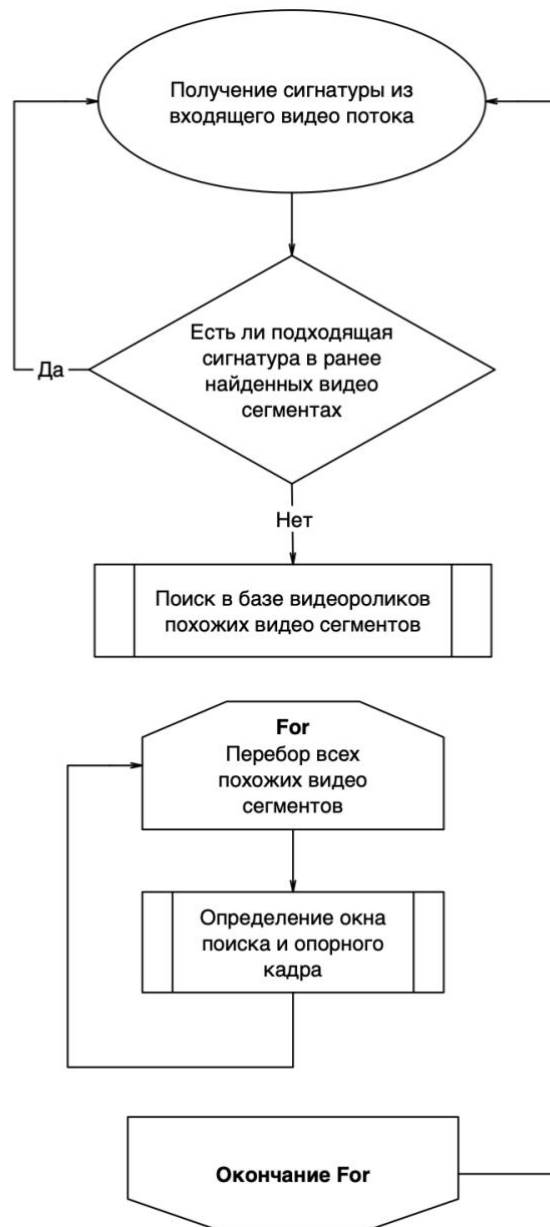


Рис. 13. Схема работы конечного автомата

После получения очередной сигнатуры из бесконечной входящей видеопоследовательности, эта сигнатура подается на вход алгоритму поиска похожих видео сегментов. Поиск выполняется с большим порогом, чтобы отсеять только гарантированно несовпадающие видео части. Алгоритм поиска выполняет сравнение с ключевыми сигнатурами видео сегментов. Если ключевая сигнатура отличается от входной сигнатуры больше порога, то этот видео сегмент игнорируется. Иначе он помечается как активный, т.е. находящийся в работе. При этом алгоритм пропускает ключевые сигнатуры ранее найденных активных сегментов, поскольку они и так уже в работе.

После получения набора похожих видео частей, необходимо определить точную позицию совпадающей сигнатуры. Для выполнения данной задачи необходимо сравнить все сигнатуры внутри сегмента с сигнатурой из входного видеопотока. Причем в этом случае сравнение сигнатур должно выполняться с очень маленьким порогом. Величина порога сравнения сигнатур должна учитывать только изменения, вносимые различными алгоритмами сжатия и шумами в тракте передачи телевизионных изображений. Если в видео сегменте не нашлось ни одного кадра с похожей сигнатурой, то сегмент исключается из работы (не помечается как активный).

Внутри сегмента может находиться несколько сигнатур, совпадающих с входной сигнатурой. Такие случаи возникают, например, при очень медленной смене сцены, или когда в видео картинка не меняется некоторое время. В таких случаях точное определение позиции текущего входного кадра внутри видео сегмента становится невозможным. Для того, чтобы корректно определить позицию входной сигнатуры внутри сегмента, используется окно поиска. То есть вместо точной позиции задается две границы – левая и правая граница окна поиска. Внутри окна поиска все сигнатуры видео сегмента слишком похожи на текущую входную сигнатуру, то есть разница их сигнатур не превышает заданного минимального порога. Индекс опорной сигнатуры совпадает с левой границей окна. Опорная сигнатура это сигнатура кадра, которая является искомой на данной итерации конечного автомата. Если на следующей итерации конечного автомата сигнатура из входящего видеопотока не совпадает с ожидаемой сигнатурой из видео сегмента, то выполняется корректировка окна таким же способом, как и его определение. Однако, если корректировка окна не требуется, или если после корректировки левая граница не изменилась, то индекс опорной сигнатуры увеличивается на единицу, а предыдущая сигнатура помечается как совпавшая.

На каждой итерации конечного автомата имеется текущий список активных сегментов, причем каждый раз для каждого активного сегмента индекс опорной сигнатуры увеличивается. В какой-то момент времени индекс

опорной сигнатуры доходит до конца видео сегмента и работа с этим видео сегментом заканчивается. В этом случае выполняется переход на следующий видео сегмент в текущем видео ролике. Если это последний сегмент в ролике, то обработка ролика заканчивается. Также, обработка ролика заканчивается, если на очередной итерации входная сигнатура сильно отличается от любой из сигнатур в текущем окне поиска.

В момент окончания обработки ролика выполняется проверка длительности обнаруженного фрагмента ролика. Для этого в обратном порядке, начиная с текущего активного видео сегмента перебираются все сегменты ролика и в каждом из них подсчитывается количество совпавших сигнатур. Если полное число совпавших сигнатур превышает заданный порог минимальной длительности обнаружения, то на выход системы (например, в лог-файл) заносится информация о выходе в эфир конкретного фрагмента ролика с информацией о совпавших кадрах. При этом все обработанные видео сегменты инициализируются и исключаются из работы. При инициализации сбрасывается информация о текущем окне поиска, об индексе текущей сигнатуры и о числе совпавших сигнатур.

В процессе сравнения сигнатур необходимо учитывать возможность возникновения ошибок – повторение или пропуск кадра. Из-за ошибок в потоке передачи телевизионного сигнала, а также из-за разницы хода часов передающего и принимающего телевизионных устройств, в принимаемом сигнале могут пропускаться или повторяться некоторые кадры. Причем частота таких событий небольшая – не более одной ошибки на 10000 кадров. Соответственно, при сравнении текущей сигнатуры с сигнатурами видео сегмента нужно всегда выходить за границы текущего окна поиска на один кадр вправо и влево. Если совпадение за границами окна, то счетчик ошибок увеличивается на единицу. При этом, если общее число ошибок превышает одну ошибку на 10000 совпавших сигнатур, то обработка ролика заканчивается.

Вопрос быстрого поиска сигнатуры в полном списке ключевых сигнатур очень важен для ускорения процесса работы итогового алгоритма, однако он выходит за рамки данной дипломной работы, поскольку реализуется в другой дипломной работе. Время, затраченное на процедуру поиска совпадающих ключевых сигнатур, не будет учитываться в данной работе.

3.1 Разбиение ролика на сегменты

Как было показано в главе “Разбиение на сцены”, разбиение ролика на сцены с помощью фиксированного порога и сравнения соседних кадров оказалось недостаточно эффективным решением. Так же при помощи этого разбиения кадры внутри сегмента могут сильно отличаться от ключевой сигнатуры, что неприемлемо в рамках разработанного алгоритма.

В главе “Выделение ключевых сигнатур”, был разработан способ разделения видеоролика на сегменты, который агрегирует похожие сигнатуры внутри сегмента с некоторым порогом, обозначим его “ C_3 ”. Стоит отметить, что чем выше порог разделения сцены, тем более длинные сегменты сигнатур сформируется. Это значит, что в базе данных поиска будет храниться меньше ключевых сигнатур. Однако, как было отмечено во введении данной главы для определения окна поиска, в худшем случае, необходимо сравнить входную сигнатуру со всеми сигнатурами видео сегмента. Такой случай может выйти за рамки задачи реального времени.

На вход алгоритму разделения на видео сегменты подается массив сигнатур и массив информации о нулевых промежутках. Нулевые промежутки были описаны в главе “Особенности вычисления сигнатуры с помощью БПФ”. Как было исследовано в данной главе, нет необходимости хранить нулевые сигнатуры, достаточно хранить лишь информацию об их длине и место в массиве сигнатур. Информация о количестве нулевых сигнатур добавляется в видео сегмент. Такая информация нужна для точной синхронизации по времени с входящим видеопотоком.

На примере видеоролика №3 из главы “Выделения ключевых сигнатур” было показано, что необходимо ограничение на минимальный размер видео сегмента. Такие видео сегменты помечаются как “особенные”. Их особенность заключается в определении ключевой сигнатуры. В обычном случае ключевой сигнатурой является первая сигнатура видео сегмента. В особом видео сегменте ключевой сигнатурой выбирается последняя сигнатура видео сегмента. Такой прием позволит своевременно выполнять постобработку при анализе входного видео потока и будет подробно объяснен в следующей главе.

На рис. 14 представлена принципиальная схема разработанного алгоритма разбиение видеоролика на видео сегменты.

Далее информация о видео сегментах и сигнатурах сохраняется в базе данных. Поскольку в видео сегменте хранится информация о ключевой сигнатуре, появляется возможность производить поиск по таким сигнатурам в базе данных.

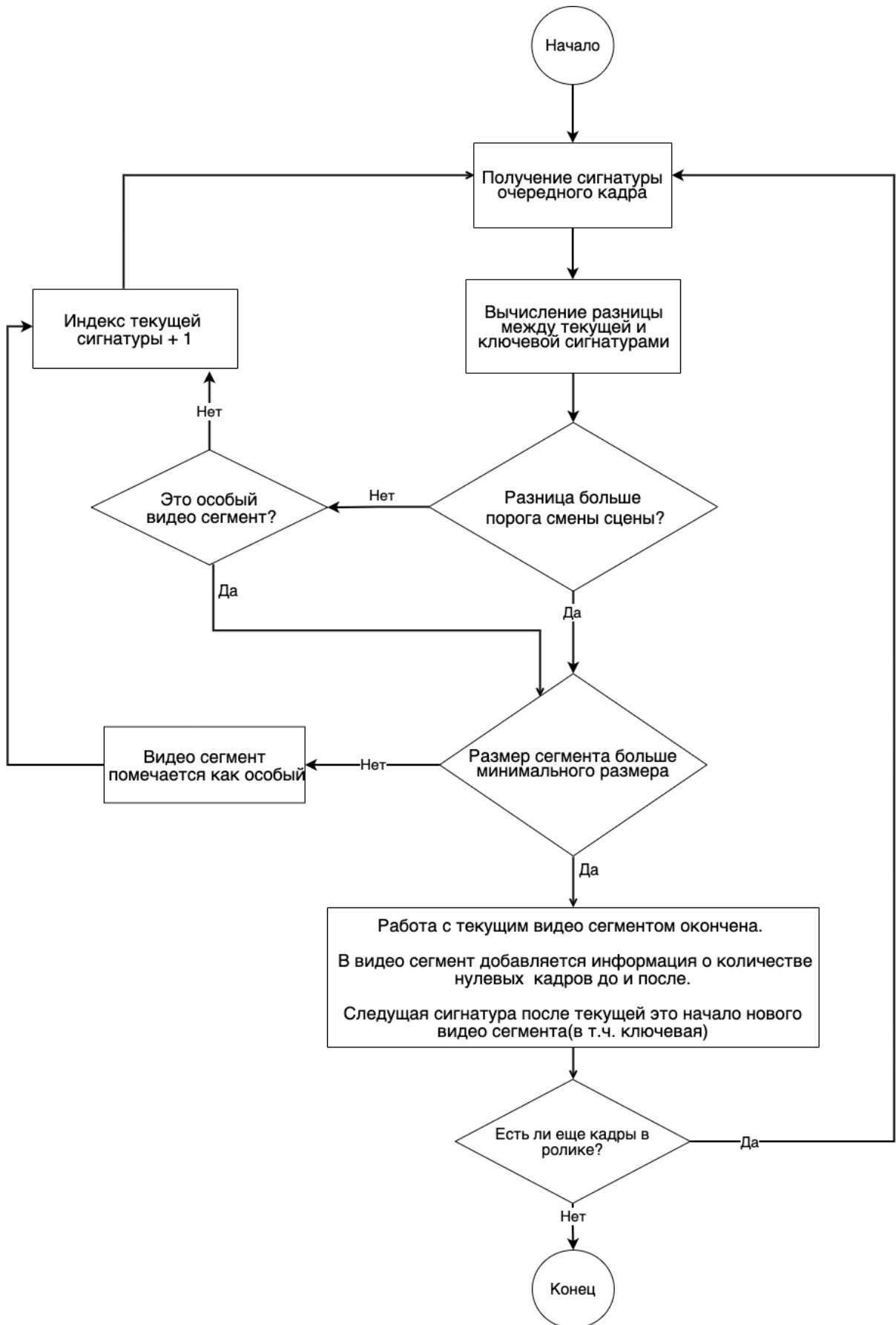


Рис. 14. Схема алгоритма разбиения видео на сегменты

3.2 Анализ входящего видеопотока

3.2.1 Определение границ окна поиска в новых сегментах

Как было замечено в главе “Разработка алгоритма”, в случаях плавной смены кадров или «стоячего» кадра, возникает необходимость использования окна поиска. Внутри сегмента сравнение сигнатур происходит с очень низким порогом C_2 . Поскольку видео сегменты состоят из похожих кадров, то внутри сегмента может быть несколько сигнатур, совпадающих с текущей входной сигнатурой. Как правило они располагаются непрерывно, поэтому имеет смысл искать границы окна начиная с краев сегмента до первого похожего кадра. То есть для определения левой границы окна необходимо сравнивать сигнатуры в сторону увеличения индекса начиная с первого кадра сегмента до тех пор, пока разница сигнатур не станет меньше минимального порога C_2 . Соответственно для определения правой границы окна необходимо сравнивать сигнатуры в сторону уменьшения индекса, начиная с последнего кадра сегмента до тех пор, пока разница сигнатур не станет меньше минимального порога C_2 . На рис. 15 и 16 представлена схема алгоритма. **rightIndex** обозначена права граница окна поиска, а **leftIndex** – левая граница. C_3 – порог разделения на сегменты из главы “Разбиение ролика на сегменты”.

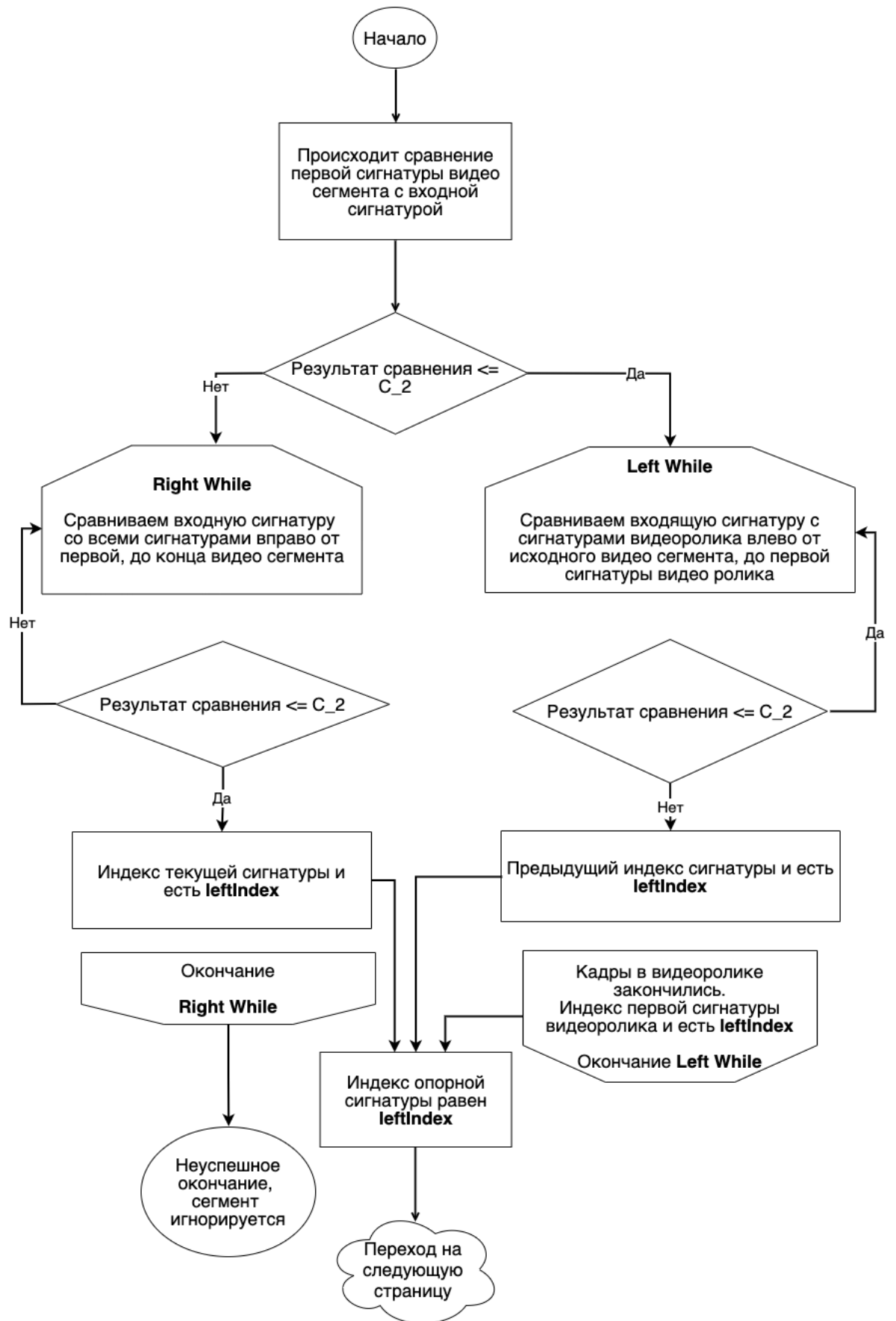


Рис. 15. Схема алгоритма определения границ окна. Часть 1

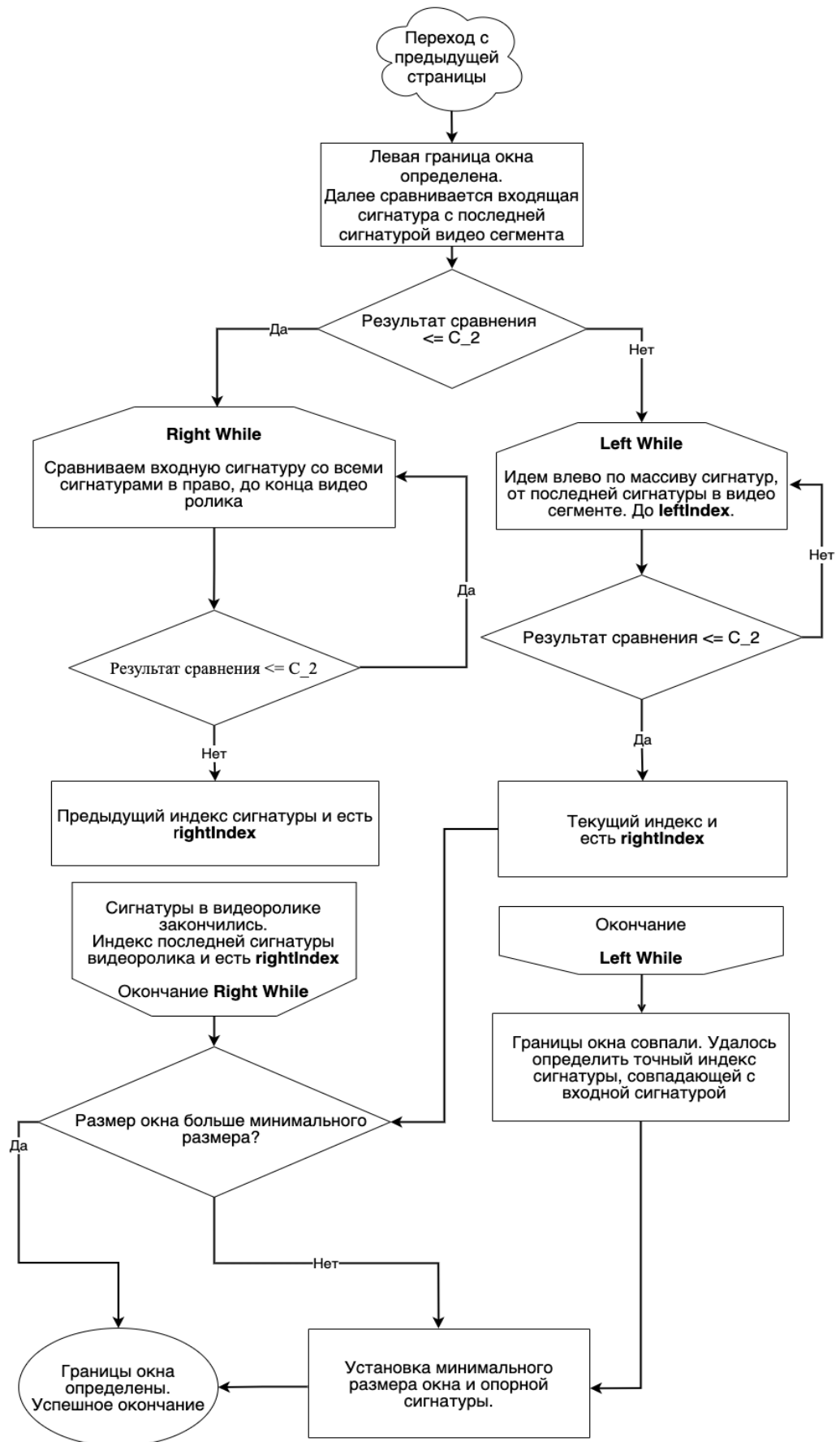


Рис. 16. Схема алгоритма определения границ окна. Часть 2

Стоит отметить, что при плавной смене сцены алгоритм, предложенный в главе “Разбиение ролика на сегменты”, сегментирует видео ролик не всегда корректным образом. Для такого видео разделение на сегменты производится на кадре, для которого отличие сигнатуры первого кадра сегмента от сигнатуры текущего кадра превышает заданный порог C_3 . При этом разница сигнатур последнего кадра первого сегмента и первого кадра второго сегмента может быть очень маленькой, меньше минимального порога сравнения сигнатур C_2 . В таких случаях алгоритм поиска сигнатур может решить, что текущая входная сигнатура совпадает с ключевой сигнатурой второго сегмента, но не совпадает с ключевой сигнатурой первого сегмента. Хотя на самом деле на вход поступил кадр из «хвоста» первого видео сегмента. Чтобы избежать подобных ситуаций необходимо при включении в работу нового видео сегмента проверить границы окна за пределы самого сегмента, до тех пор, пока разница между текущей сигнатурой и сигнатурами ролика меньше минимального порога C_2 .

Также к отдельной категории относятся ситуации пропуска кадра в входящем видеопотоке, для обработки таких случаев, минимальный размер окна должен быть не менее ± 1 кадр влево и вправо от опорной сигнатуры.

3.2.2 Корректировка окна поиска

Алгоритм, описанный в предыдущей главе, необходим только для определения границ окна, однако он не подходит для корректировки границ. Корректировка окна поиска не происходит на той же итерации, что и определение его границ. Алгоритм корректировки окна поиска необходим на следующих итерациях работы конечного автомата. Концептуально он совпадает с алгоритмом определения границ окна, однако рамки, в которых происходит определение нового окна поиска, ограничены старыми границами окна.

Алгоритм выполняет поиск совпадающей сигнатуры с левого края окна поиска до правого, которая отличается от текущей входной сигнатуры меньше

минимального порога C_2 . Такая сигнатура становится опорной, а левая граница окна корректируется. Если левая граница окна не изменилась, то индекс опорной сигнатуры увеличивается на единицу. Для определения правой границы окна поиска выполняется поиск справа налево по окну поиска. Первый кадр, который подходит по условию, является правой границей нового окна поиска. Обозначения аналогичны предыдущей главе: **leftIndex** - левый край окна в массиве сигнатур ролика, **rightIndex** - правый край окна в массиве сигнатур ролика.

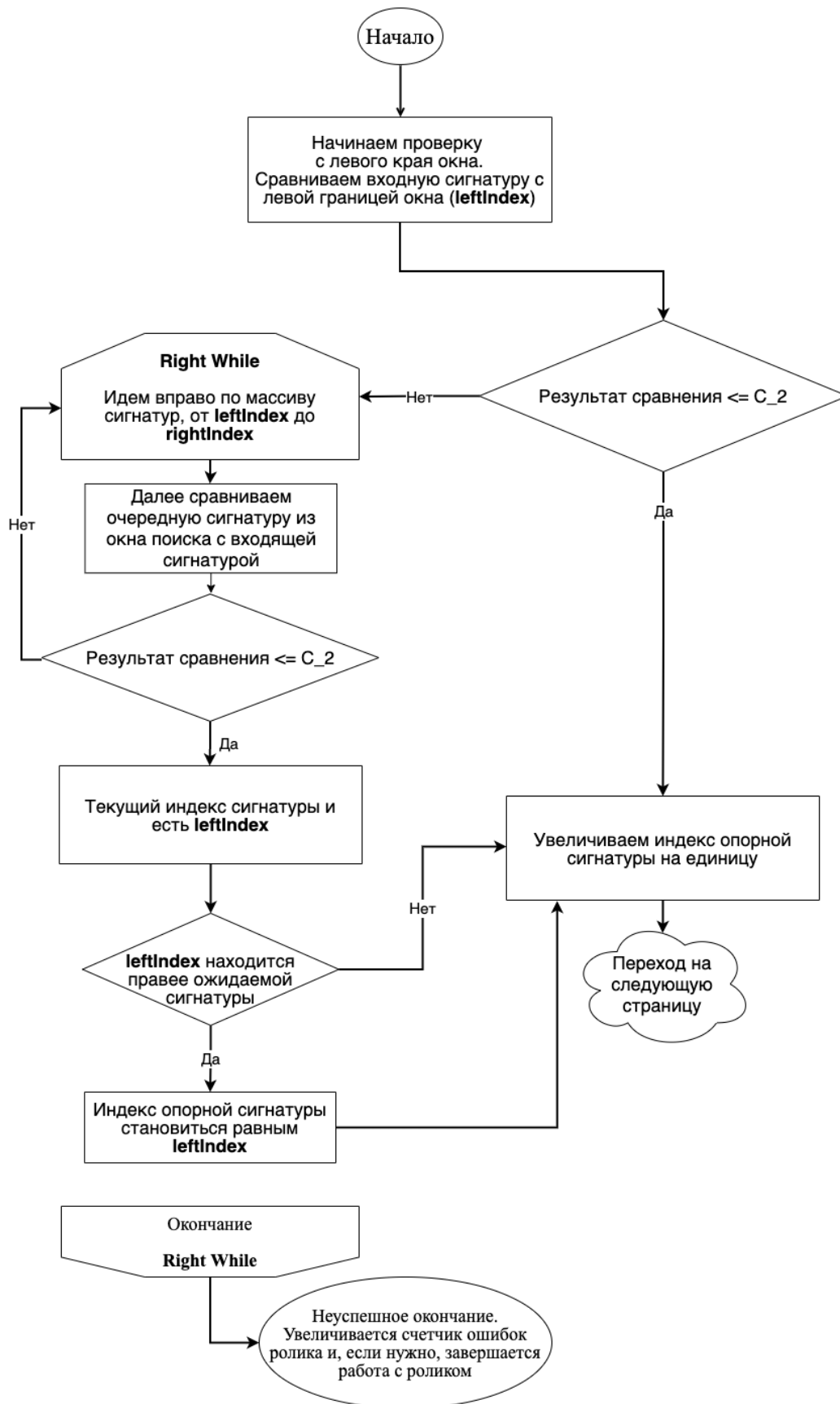


Рис. 17. Схема алгоритма корректировка окна поиска. Часть 1

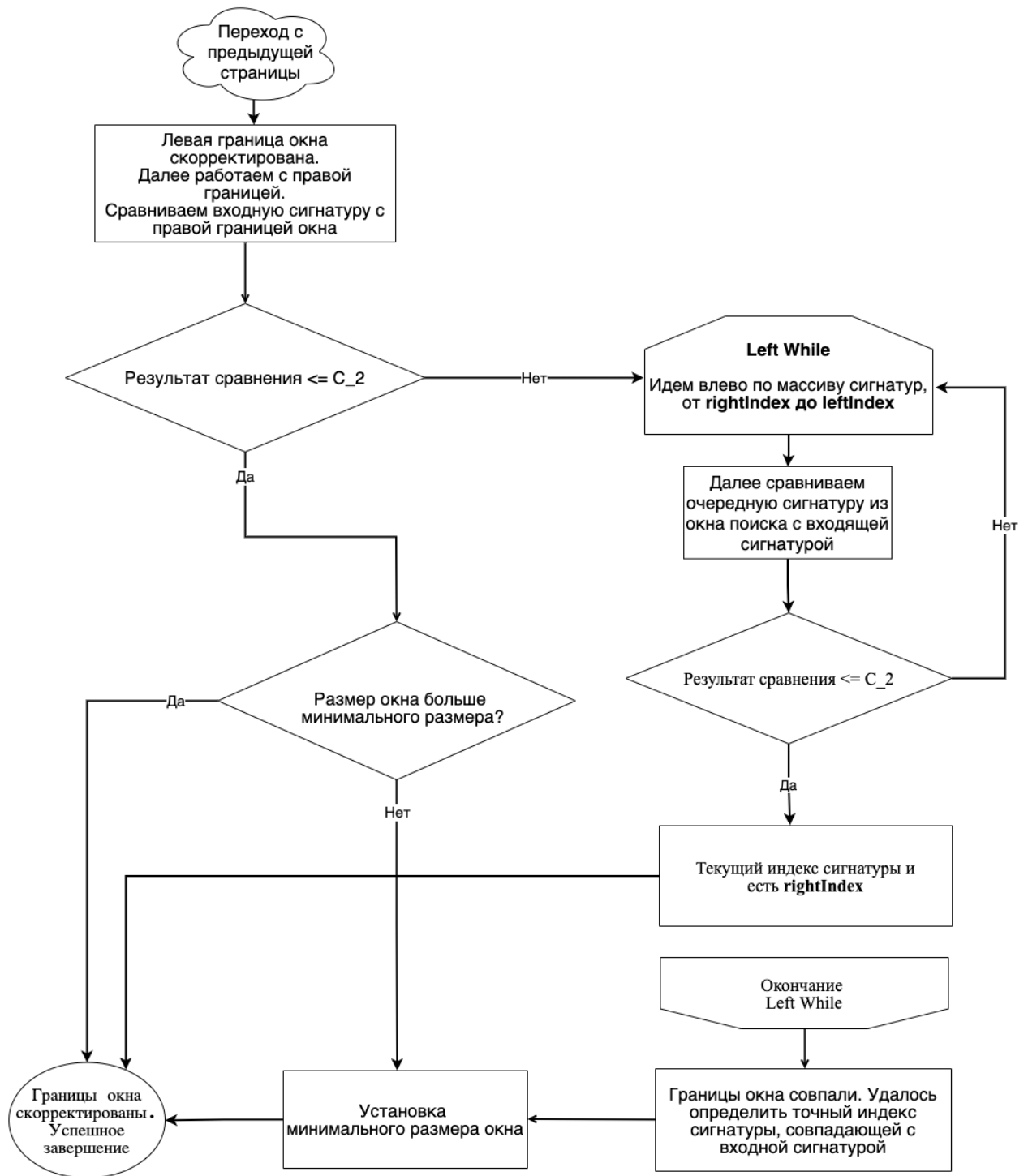


Рис. 18. Схема алгоритма корректировка окна поиска. Часть 2

3.2.3 Работа с активным сегментом

После определения окна поиска в новых видео сегментах и добавления в список активных сегментов, необходимо выполнить обработку видео сегментов, которые стали активными на предыдущих итерациях алгоритма.

Корректировка окна – это достаточно ресурсоемкая операция (по количеству сравнений), поэтому выполнение этой операции на каждой итерации алгоритма является не обоснованным. В этой главе предлагается дополнительная проверка, которая позволяет избежать корректировки окна. В каждом активном видео сегменте есть следующая ожидаемая сигнатура, это сигнатура, которая ожидается на следующей итерации алгоритма из входного видеопотока. Если разница входящей сигнатуры с ожидаемой сигнатурой из видео сегмента меньше минимального порога C_2 , то корректировка окна не производится. Также в этой главе описывается удаление видео сегмента из списка активных и переход от одного видео сегмента к другому. В случае, когда происходит переход окна поиска на следующий видео сегмент, производится добавление этого видео сегмента в список активных. При этом предыдущий видео сегмент не удаляется из списка активных, чтобы алгоритм поиска ключевых сигнатур игнорировал этот видео сегмент, но этот видео сегмент не обрабатывается, поскольку окно поиска нулевое. Если в видео сегменте не было найдено сигнатур похожих на входную сигнатуру, т.е. видео сегмент оказался неподходящим, то происходит удаления сегмента из списка активных, как и всех предыдущих активных видео сегментов до него. На рис. 19 представленная схема алгоритма.

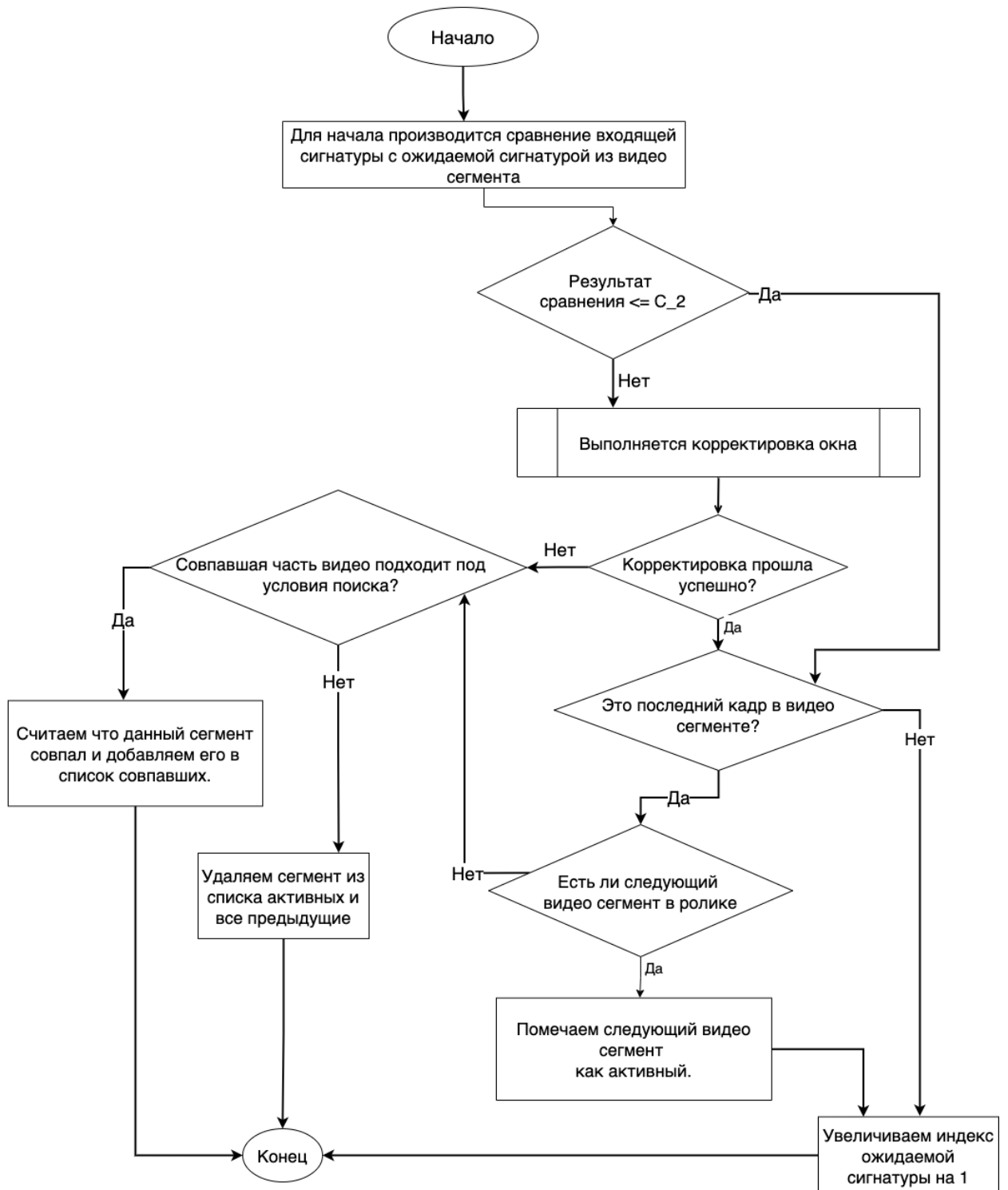


Рис. 19. Схема алгоритма обработки активного сегмента

3.2.4 Обработка результатов поиска в базе данных

После получения очередного кадра из входящего видеопотока выполняется поиск по базе данных подготовленных видеороликов. Данный алгоритм

производит поиск с порогом, равным порогу разбиения на сегменты C_3 из главы “Разбиение ролика на сегменты”. Результатом поиска является набор видео сегментов, которые подходят под условия поиска. Далее необходимо обработать как вновь полученные видео сегменты, так и сегменты, полученные на предыдущих итерациях алгоритма.

На рис. 20 представлена схема обработки результатов поиска в базе данных.

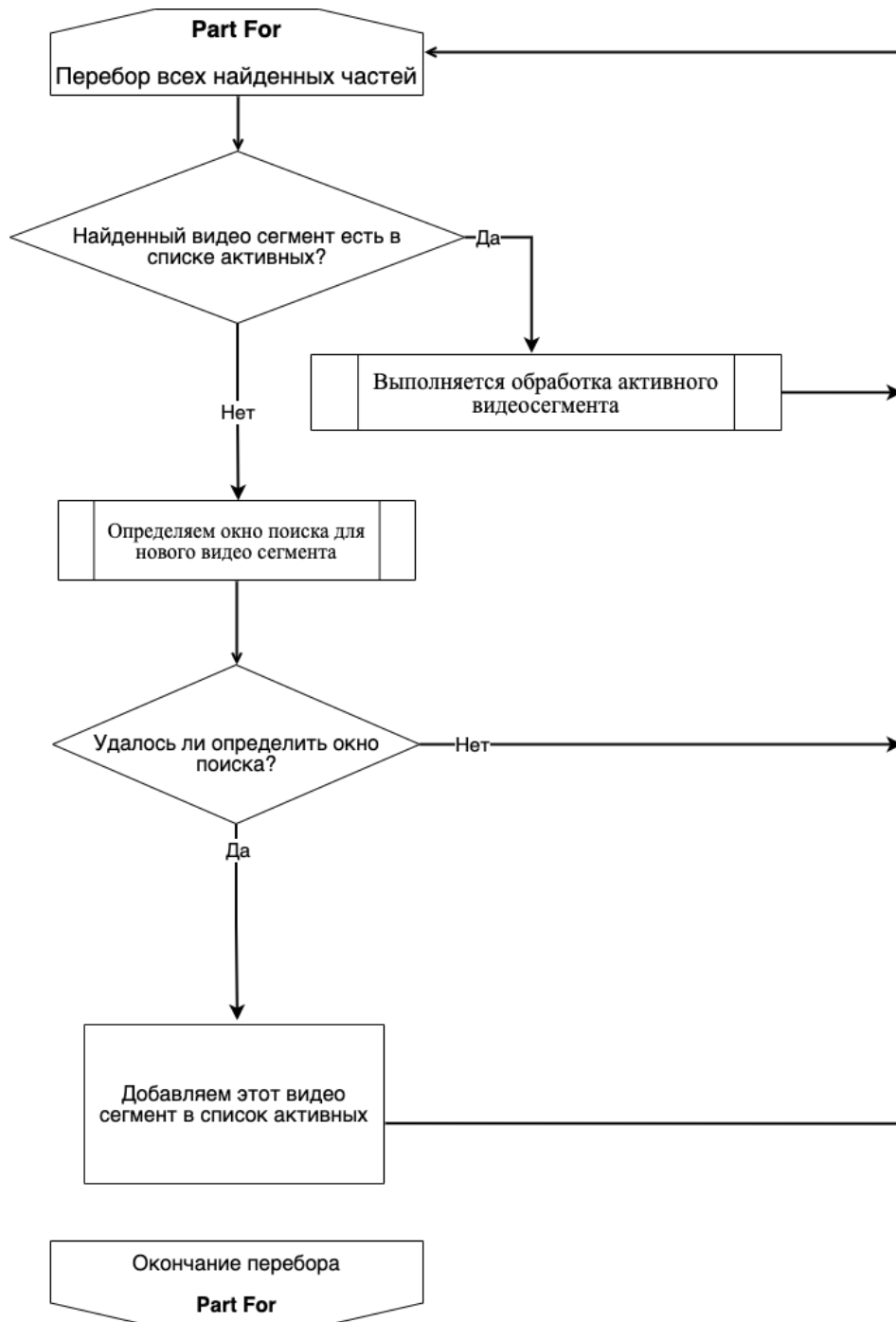


Рис. 20. Схема алгоритма обработки результатов поиска в базе данных

После окончания работы с найденными видео сегментами выполняется повторный проход по массиву активных видео частей и обработка тех сегментов, с которыми не производилась работа на текущей итерации.

Глава 4. Оценка сложности алгоритма

Как было упомянуто в главе “Разбиение ролика на сегменты”, порог C_3 влияет на итоговое количество сегментов видеоролика. А именно, чем ниже порог, тем большее количество видео сегментов сформируется из ролика и, как следствие, большее количество ключевых сигнатур, с которыми будет производиться сравнение при поиске в базе данных. Чем выше порог разделения на сегменты, тем меньшее количество сегментов получается в результате. В текущей дипломной работе используется линейный поиск в базе данных. При поиске в базе данных сравниваются ключевые сигнатуры сегментов всех видео роликов. Результатами поиска являются только те видео сегменты, которые совпадают с входящей сигнатурой с порогом C_3 , т.е. таким же порогом, который используется при разделении на сегменты.

При оценке сложности алгоритма главным критерием в данной дипломной работе является количество сравнений сигнатур. В худшем случае в алгоритме определения окна поиска, описанном в главе “Определение границ окна поиска в новых сегментах” необходимо сравнить входную сигнатуру со всеми сигнатурами во всех новых сегментах:

$$n = \sum_{i=0}^q b_i \quad (1)$$

, где n – количество сравнений, b_i – количество сигнатур в i – ом сегменте.

Пусть q – среднее количество новых сегментов, которые будут найдены при поиске ключевых сигнатур для одного очередного кадра входного потока. Для оценки q произведем автокорреляцию базы сигнатур. В качестве входного потока используем все сигнатуры всех видео роликов. Выполним поиск каждой сигнатуры по базе данных сегментов, исключая из сравнения все видео сегменты того же ролика. Сигнатуры сравниваются с порогом C_3 . Произведя

$N * p$ сравнений, где N – общее количество сигнатур (суммарное количество кадров во всех роликах), а p – количество сегментов в базе, получим Q совпадений. Тогда на один кадр приходится q похожих сегментов:

$$q = \frac{Q}{N}$$

Важно отметить, что величина $q \sim p$, поскольку удвоение сегментов в базе в среднем приводит к удвоению количества найденных (похожих) сегментов.

Далее необходимо оценить μ – среднее количество кадров в видео сегменте. Для этого необходимо посчитать количество кадров в базе данных – N , а также общее количество сегментов в базе данных – p . Таким образом:

$$\mu = \frac{N}{p}$$

Формула 1 эквивалентна следующей формуле:

$$n = q * \mu$$

n – количество сравнений сигнатур на одну входящую сигнатуру.

Глава 5. Тестирование и реализация

Для оценки сложности алгоритмов необходимо определиться с порогом C_3 . Выбрав его достаточно высоким можно уменьшить количество сравнений в базе данных, однако таким образом мы увеличиваем длину сегмента μ . Величину q необходимо оценить экспериментально. В курсовой работе [4] было показано, что разница между соседними сигнатурами в разных сценах начинается от значения равного 0.4. Следовательно, некорректно устанавливать порог разделения на сегменты C_3 больше 0.4.

Для оценки величины q были выбраны следующие пороги: 0.2, 0.25 и 0.3. База роликов выбиралась таким образом, чтобы смысловая нагрузка в видео роликах была абсолютно разная, т.е. нет одинаковых заставок и тому подобное. Благодаря такому ходу можно экстраполировать полученные результаты на большие масштабы базы данных.

Для тестирования были отобраны 100 роликов, в которых суммарно 336 988 сигнатур. На рис. 21 отображены результаты для разных порогов в работе с построенной базой данных на основе 100 тестовых роликов.

Значение/порог	0.2	0.25	0.3
Кол-во видео сегментов	10184	9373	8616
q	$13 * 10^{-3}$	$17 * 10^{-3}$	$26 * 10^{-3}$
μ	33.1	36	39.1
n	0.44	0.61	1.02

Рис. 21. Таблица результатов

Как видно из таблицы (рис. 21), уменьшение порога C_3 приводит к увеличению количества сегментов. Следовательно, количество сравнений при поиске в базе данных также увеличивается. Однако несмотря на то, что с увеличением порога количество сегментов уменьшается при этом значение q увеличивается. Это значит, что уменьшение числа сегментов менее значимо, чем увеличение числа совпадений сигнатур, возникающие из-за увеличения порога сравнения. Как следствие, значение n так же растет. Можно сделать вывод, что для уменьшения числа сравнений выгоднее брать меньший порог. Однако при этом возрастает количество сегментов, и, как следствие, возрастает время поиска в базе данных ключевых сигнатур. Это значит, что выбор порога для деления на сегменты не такая однозначная задача. Обладая информацией, приведенной в данной главе, пользователь сможет подобрать порог деления на сегменты в своей задаче самостоятельно.

В текущей дипломной работе была поставлена задача разработать алгоритм, позволяющий работать в реальном времени с базой данных 10^6 кадров. Экстраполируя значения, полученные в таблице результатов (рис. 21) на порядок, при пороге 0.2, получаем $n = 4.4$. Т.е. около 5 сравнений на один кадр входного потока прямого вещания при том, что размер базы будет

составлять $\approx 3.3 * 10^6$ сигнатур. Таким образом, из максимально возможного числа сравнений сигнатур $1,5 * 10^4$ разработанному алгоритму требуется не более 5 сравнений. Это составляет порядка 0.1% от доступной производительности.

Для проверки работы алгоритма поиска видеороликов по ключевым сигнатурам было разработано тестовое приложение. Был смонтирован видео файл, состоящий из трех кусков роликов содержащихся в базе данных, а также из кусков, которых нет в базе роликов. Тестовая программа с точностью до кадра нашла совпадающие фрагменты видео роликов.

Заключение

В работе были проанализированы особенности формирования сигнатур. По итогам анализа был выбран алгоритм генерации сигнатур, обеспечивающий высокую корреляцию схожести сигнатур со схожестью изображений.

Был разработан алгоритм выделения ключевых сигнатур, позволяющий в десятки раз сократить количество сигнатур используемых для сравнения.

Был разработан алгоритм поиска видеороликов по ключевым сигнатурам, требующий небольшое число сравнений на один входной кадр, но обеспечивающий высокую точность нахождения фрагментов видео роликов.

Разработанные алгоритмы были реализованы и протестированы на тестовом наборе из 100 роликов разного типа. Тестирование показало корректность работы алгоритма. По итогам тестирования была произведена оценка ресурсоемкости разработанных алгоритмов. При использовании данного алгоритма удастся использовать только 0.1% от доступной производительности. Оставшиеся ресурсы можно использовать при вычислении сигнатуры входного потока, а также при поиске в базе данных.

Список литературы

1. Coskun B., Sankur B. Robust video hash extraction // Bebek, Istanbul, 2007, DOI:10.5281/ZENODO.38458
2. Naturel X. and Gros P. A Fast Shot Matching Strategy for Detecting Duplicate Sequences in a Television Stream // ACM. 2005. DOI:10.1145/1160939.1160947
3. Ромб Е. О. Поиск коротких последовательностей полноэкранных изображений в видео потоке. // г. Новосибирск, ИАиЭ СО РАН, кафедра АФТИ, 2008 г.
4. Мустафин В.М. Разработка библиотеки для поиска видеофрагментов в последовательности изображений, поступающих в реальном времени // г. Новосибирск, ИАиЭ СО РАН, кафедра АФТИ, 2020г.
5. MICHAEL J. Color Indexing // International Journal of Computer Vision 1991,7:1, 11-32.
6. Dinkar N., Shree K.. Ordinal Measures for Image Correspondence // IEEE transactions on pattern analysis and machine intelligence, APRIL 1998, Vol. 20, No. 4.