

МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

К защите
Заведующий кафедрой

«_____» _____ 2015г.

Выпускная квалификационная работа бакалавра

на тему: «ДИНАМИЧЕСКИЕ ВИДЕО КАРТИНКИ В СИСТЕМАХ
ТЕЛЕВИЗИОННОГО ВЕЩАНИЯ И СПОСОБЫ АНАЛИЗА
КОМПЕНСАЦИИ ДВИЖЕНИЯ»

Выпускник _____ **Нахшванов Р.Ф.**
(подпись) (Фамилия)

Консультант _____ **Рахимов Б.Т.**
(подпись) (Фамилия)

Рецензент _____
(подпись) (Фамилия)

Консультант
по БЖД и Э _____ **Амурова Н.Ю.**
(подпись) (фамилия)

Ташкент-2015

Введение

В своем докладе Президент Республики Узбекистан Ислам Каримов на заседании Кабинета Министров, посвященном основным итогам 2012 года и приоритетам социально-экономического развития на 2013 год 18 января 2013 года отметил, что осуществлен поэтапный переход на цифровое телевидение путем установки телепередатчиков в Ферганской, Навоийской, Сырдарьинской и Сурхандарьинской областях, что позволило довести уровень охвата населения цифровым телевидением до 42 процентов [1].

Чрезвычайно важное значение приобретает в этом плане безусловное выполнение разработанной на 2013 год Инвестиционной программы, в которой предусматривается осуществление более 370 стратегически важных проектов.

Из общего объема выделяемых на эти цели 13 миллиардов долларов 75 процентов составляют средства, финансируемые за счет внутренних источников, остальная часть – это привлекаемые иностранные инвестиции [2].

Все большее значение приобретает ускоренная реализация мер и проектов в сфере информационно-коммуникационных и телекоммуникационных технологий. Необходимо в кратчайшее время не только устранить имеющее место отставание по многим видам оказания информационных услуг, но и выйти в разряд передовых стран с высоким уровнем внедрения информационно-коммуникационных технологий.

Долгосрочное архивирование имело место еще в древние времена у различных цивилизаций. При этом выбранная стратегия показалась настолько эффективной, что выдолбленную в камне информацию люди вполне способны разобрать и сейчас, спустя тысячи лет. Но современное человечество столкнулось с проблемой использования сведений, сохраненных сегодня, через несколько лет. Растущий объем

цифровизации влечет за собой потребность в сохранении все большего объема контента, поэтому выбор правильного пути архивирования сегодня особенно актуален.

Ведь стремительное устаревание воспроизводящих средств (как следствие смены технологий) затрудняет доступ к материалам, сделанным в форматах видеозаписей предыдущих поколений. Все чаще становится невозможным использование архивных материалов из-за отсутствия как самих исправных воспроизводящих устройств, так и квалифицированных специалистов, которые умеют с ними обращаться. Более того, стоимость содержания зданий оборудованных хранилищ, в которых хранятся пленки и видеоленты, в несколько раз превышает стоимость технологического перевооружения.

Объем производства медиа продукта лавинообразно растет, и размер архивов (как в старых, так и в новых форматах) постоянно увеличивается. До недавнего времени это были преимущественно материалы стандартной четкости, но теперь существуют материалы, записанные в форматах ТВЧ (телевидение высокой четкости).

Долговременное хранение контента - динамичный процесс. Начинается он с правильного учета битов и байтов, включает в себя их структуризацию, хранение, распределение и архивирование. Таким образом, без специализированных систем управления контентом (систем управления коллективным контентом - Enterprise Content Management, ESM) не обойтись.

За последние годы рынок ESM претерпел важные структурные изменения. В их основе лежит коррекция потребностей заказчиков в области ESM и новые возможности, открываемые развитием самих ИТ.

Большое значение при архивировании информации имеет формат, в котором должны храниться оцифрованные данные. Независимость от производителя в данном случае играет ведущую роль. Только таким

образом можно избежать ситуации, когда в какой-то момент файл невозможно прочитать из-за того, что производитель перестал поддерживать какой-либо формат.

Поэтому идеальным вариантом в современных медиа архивах на первоначальной стадии обработки является перевод в файловую форму, имеющую возможность автоматически трансформироваться в новые форматы по мере их возникновения. Данная операция называется "инвестирование". В результате появляется так называемый мастер-файл, который существует в первоначальном виде десятки лет. Инвестирование - важный момент в процессе архивирования, включающий в себя две основные составляющие: предварительная обработка материала (очистка аудио/видеоматериала) и его описание (добавление мета контента). Причем созданный на стадии инвестирования мастер-файл должен обладать следующими свойствами: высокое качество, пригодность как для хранения, так и для любых других применений (от передачи до редактирования).

Весь медиа контент, который был ранее переведен в цифровую форму, необходимо грамотно упорядочить для того, чтобы обеспечить в дальнейшем простой поиск информации. Обработанные файлы затем необходимо классифицировать и разделить на основе мета контента.

В крупных медиа компаниях, где ежедневно обрабатываются большие объемы контента, рассматриваемый этап работы может стать затруднительным. В этом случае могут выручить программные модули, автоматизирующие классификацию контента. В дальнейшем результаты классификации используются при распределении по категориям или рассматриваются как ключевые слова, входящие в метаданные контента.

АЛГОРИТМЫ СОКРАЩЕНИЯ ВИДЕО КАРТИНКИ БЕЗ ПОТЕРЬ

1.1. Представление цифровых видео картинки.

Компьютерное видео картинки в его цифровом представлении является набором значений интенсивностей светового потока, распределенных по конечной площади.

Для простоты рассмотрим сначала монохромные видео картинка. Интенсивность излучаемой световой энергии с единицы поверхности в точке с координатами (x, y) видео картинка можно представить некоторым числом $B(x, y)$. Единичный элемент видео картинка, характеризующий определенным значением (x, y) , называется пикселем, а величина $z = f(x, y)$ - яркостью [4].

Прежде чем рассмотреть алгоритмы сокращения видео картинки, необходимо определить что в дальнейшем будет пониматься под видео картинкам.

С математической точки зрения, видео картинка в градациях серого можно представить как вещественную функцию I двух вещественных переменных x и y . Функция $I(x, y)$ видео картинка в общем случае определяется в прямоугольной области, но для удобства исследований в работе все видео картинка определяются в квадратных областях, т.е. $x \in [0; W]$, а $y \in [0; H]$, где W – ширина видео картинка, а H – высота видео картинка и $W=H$.

Все видео картинка можно подразделить на две группы: с палитрой и без неё. У видео картинки с палитрой в пикселе (одном из отчётов видео картинка – значение функции $I(x, y)$ для конкретного x_i и y_i) храниться число – индекс в некотором одномерном векторе цветов, называемом палитрой. Палитры обычно бывают 8, 16 и 256 – цветов[4].

Видео картинка без палитры обычно бывают в определенной системе светопредставления или в градациях серого. В градациях серого значения каждого из пикселей определяется как яркость точки. Наиболее часто встречаются видео картинка с 2-мя, 16-ю и 256-ю уровнями серого.

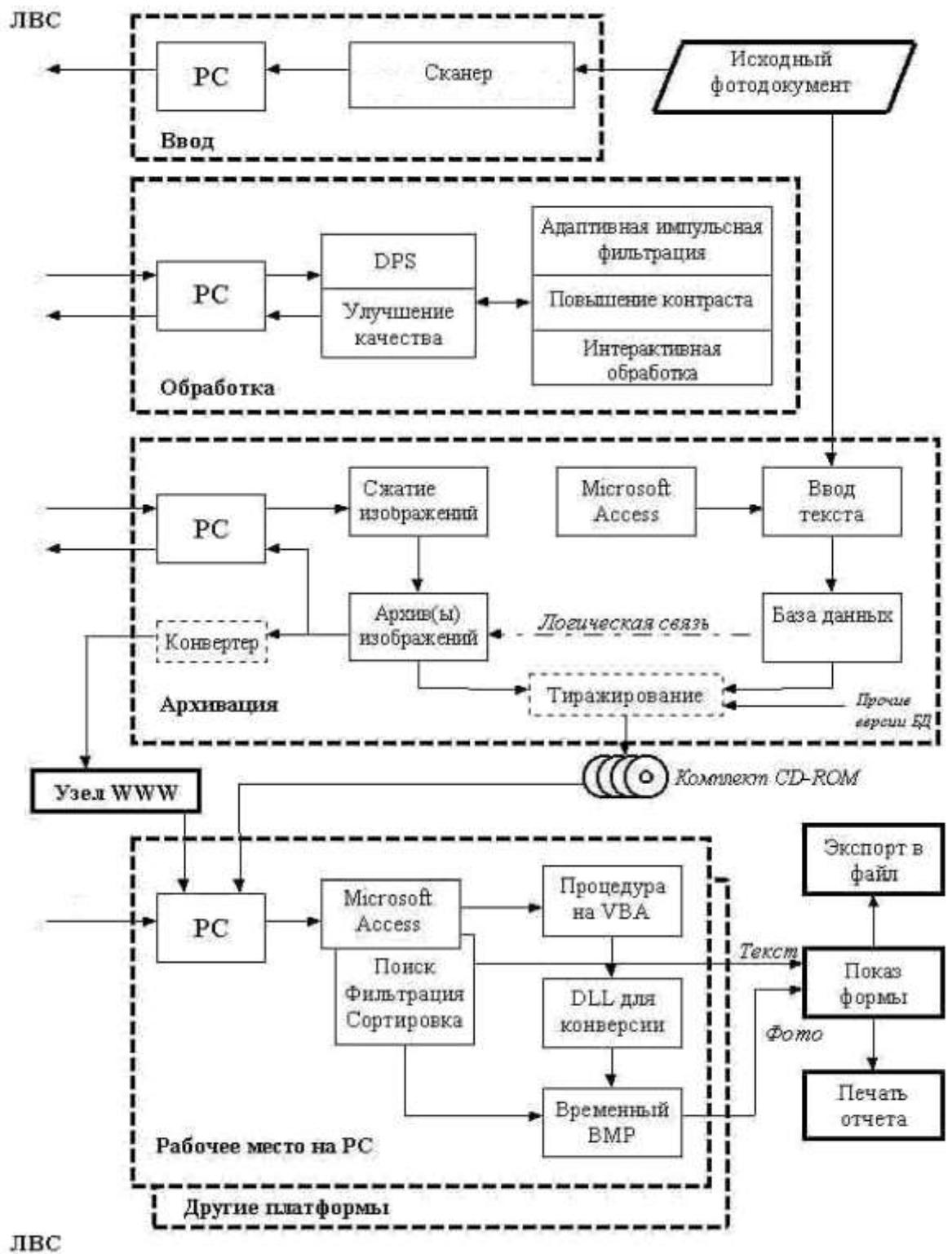


Рисунок 1.1. Технические средства телекоммуникационной системы для передачи видео контента и их функциональное назначение.

Если видео картинка представлено в какой-то системе светопредставления, то каждый её пиксель является структурой,

описывающий компоненты цвета. Наиболее распространённой системой цветопреставления, используемой в электронных и компьютерных системах, является система RGB. В этой системе цвет определяется как комбинация красного, зелёного и синего цвета. И на каждую из составляющих приходится по одному байту. Существуют и другие системы цветопреставления, такие как CMYK, CIE, YUU и YCrCb [3,4].

Для того чтобы корректнее оценивать степень сокращения видео картинка, и применимости того или иного алгоритма сокращения к данному изображению вводится понятие класса видео картинка [4].

Под классом цифрового видео картинка понимается совокупность видео картинки, применение к которым, алгоритм сокращения даёт качественно одинаковый результат. Например, для одного класса алгоритм сокращения даёт превосходный коэффициент сокращения, а для другого класса видео картинка наоборот, увеличивает объём сжимаемого файла [3].

Условно можно выделить следующие классы видео картинки:

- видео картинка с небольшим количеством цветов и большими областями, заполненными одним цветом. В видео картинки отсутствуют плавные переходы цветов. К таким классам обычно относится деловая графика, научно-техническая, инженерная или плакатная графика;
- видео картинка с плавными переходами цветов, построенные на компьютере: графика презентаций и виртуальные модели;
- фотореалистичные видео картинка, полученные после цифровой фотосъёмки, сканирования, а также постобработка этих видео картинки.

Можно выделить и специфические классы видео картинки, такие как рентгеновские снимки, тома графические видео картинка, радиолокационные планы местности и т.д. Но для сравнения алгоритмов сокращения видео картинки всегда необходимо определять класс видео картинки, с которыми они работает.

В процессе работы с видео картинками приложения, осуществляющие обработку, предъявляют различные требования к алгоритмам сокращения видео картинки. Из-за специфики приложений такие требования иногда могут противоречить друг другу. В общем случае можно выделить следующие требования к алгоритмам сокращения видео картинки:

- высокая степень компрессии;
- высокое качество сокращённого видео картинка (данное требование противоречит выполнению предыдущего требования, поэтому всегда приходится искать компромисс между степенью сокращения и качеством восстановленного видео картинка);
- высокая скорость компрессии (данное требование актуально для приложений, занимающихся кодированием видео картинки в реальном масштабе времени: цифровых фотоаппаратов, видеокамер);
- высокая скорость декомпрессии (данное требование актуально почти для всех приложений).
- возможность показать приблизительное видео картинка, не дожидаясь полной его загрузки (данное требование актуально для сетевых приложений и для приложений, занимающихся передачей больших видео картинка).
- учёт специфики видео картинка (данное требование реализуют алгоритмы сокращения, основанные на определении «области особого назначения» (ROI – regions of interest)).

1.2. Классификация способов сокращения. Основные характеристики

Все способы сокращения информации основаны на том простом предположении, что набор контента всегда содержит лишние элементы. Сокращённого объема достигается за счет поиска и кодирования лишних элементов [3,7].

Поток контента об видео картинке имеет существенное количество избыточной информации, которая может быть устранена практически без заметных для глаза искажений. При этом различают два типа лишности.

Статистическая лишность связана с корреляцией и предсказуемостью контента. Эта лишность может быть устранена без потери информации, исходные данные при этом могут быть полностью восстановлены [3,4]. Наиболее известные способы эффективного кодирования элементов основаны на знании частоты каждого элемента присутствующего в сообщении. Зная эти частоты, строят таблицу кодов, обладающую следующими свойствами:

- различные коды могут иметь различное количество бит;
- коды элементов с большей частотой встречаемости, имеют меньше бит, чем коды элементов с меньшей частотой;
- хотя коды имеют различную битовую длину, они могут быть восстановлены единственным образом, т.е. коды строятся как префиксные.

Этими свойствами обладает известный алгоритм Хаффмана [4].

Визуальная (субъективная) лишность, которую можно устранить с частичной потерей контента, мало влияющих на качество воспроизводимых видео картинки; это - информация, которую можно изъять из видео картинка, не нарушая визуально воспринимаемое качество видео картинка.

Устранение визуальной лишности видео картинки является основным резервом сокращения передаваемой информации [3,7]. Для оптимизации процесса кодирования в целях обеспечения передачи наименьшего объема информации необходимо, с одной стороны, не передавать лишнюю информацию, а с другой, - не допустить чрезмерной потери качества видео картинка.

До сих пор не существует простой и адекватной модели визуального восприятия видео картинки, пригодной для оптимизации их кодирования [7].

Задача сокращения видео картинка состоит из двух основных частей: кодирование и декодирование. Если декодированное видео картинка всегда в точности соответствует кодируемому изображению, то такой алгоритм кодирования-декодирования называется алгоритмом сокращения без потерь. Если декодированное видео картинка отличается от кодированного, то подобный алгоритм называют алгоритмом сокращения с потерями. Общая схема процесса сокращения видео картинка представлена на рисунке 1.2 [3].

Рассмотрим этапы процедуры сокращения контента в общем виде. Любой метод сокращения реализует три основных этапа (рисункунок. 1.2):

- кодирование или первичное сокращённого объема;
- вторичное сокращённого объема;
- декодирование или восстановление видео картинка. На первом

этапе выполняется преобразование исходных контента из одной

формы представления в другую. В частности, при сжатии видео картинка в зависимости от вида алгоритма сокращения может быть выполнен переход от исходного видео картинка к следующим видам представления (табл.1.1):

На втором этапе компоненты изменения формы квантуются и приводятся к виду удобному для статистического кодирования, а затем кодируются. На этом этапе обеспечивается уплотнение информационного потока.

Существует несколько различных подходов к проблеме сокращения информации. Одни имеют весьма сложную теоретическую математическую базу, другие основаны на свойствах информационного потока и алгоритмически достаточно просты. Любой способ, реализующий сокращения объема контента, предназначен для снижения объема выходного потока информации помощи обратимого или необратимого изменения формы. По этому все способы сокращения можно разделить на две категории: обратимое и необратимое сокращённого объема [3,7]. Способы сокращения цифровых видео картинки можно классифицировать по их основным характер рисунк, таким как: точность начальный вид, симметричность основного изменения формы и тип используемого изменения формы [7].

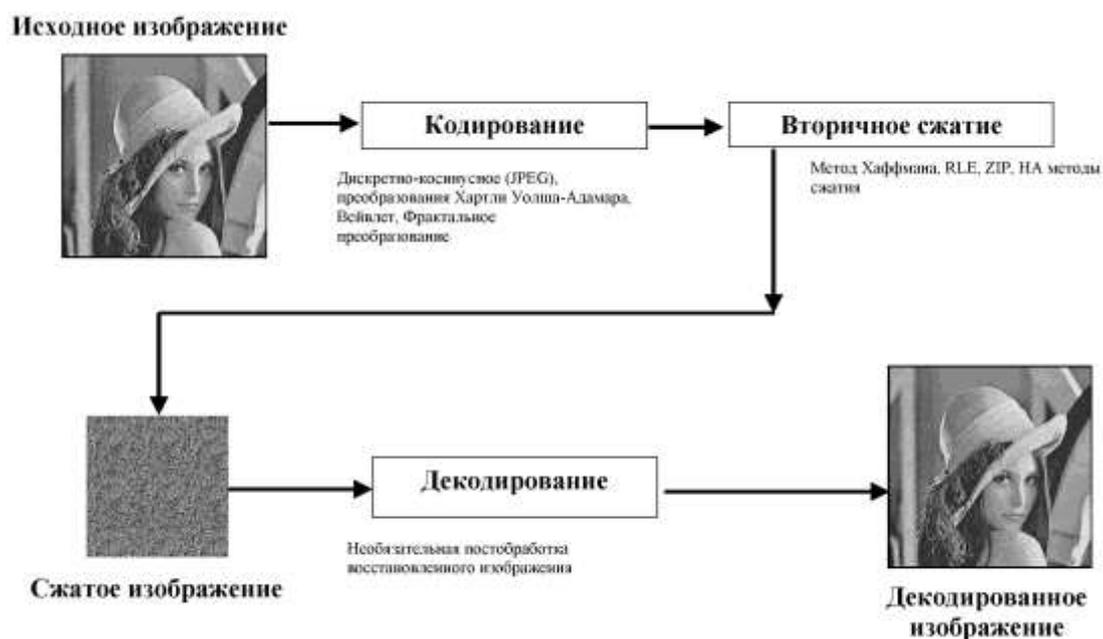


Рисунок.1.2. Основные этапы сокращения цифровых видео картинки

Таблица 1.1

Исходное видео картинки	Преобразованное видео картинки
Матрица пикселей (значения интенсивности)	Матрица компонент спектра (спектральные изменения формы)
	Набор коэффициентов изменения формы
	Описание объектов видео картинка (сокращённого объема с распознаванием)

Классификация наиболее распространенных способов сокращения приведена на рисунунок.1.3.

Обратимое сокращённого объема (сокращённого объема без потерь). Обратимое сокращённого объема всегда приводит к снижению объема выходного потока информации без изменения его информативности, т.е. без потери информационной структуры. Более

того, из выходного потока, при помощи восстанавливающего алгоритма, можно получить входной [3].

Необратимое сокращённого объема (сокращённого объема с потерями). Под необратимым сокращённым объемом подразумевают такое преобразование входного потока контента, при котором выходной поток, основанный на определенном формате информации, представляет достаточно похожий по внешним характеристикам на входной поток объект, однако отличается от него объемом. Степень сходства входного и выходного потоков определяется степенью соответствия некоторых свойств объектов (т.е. сжатой и несжатой информацией в соответствии с некоторым определенным форматом контента), представляемого данным потоком информации.

Такие алгоритмы используются для сокращения, например контента растровых графических файлов с низкой степенью повторяемости байтов в потоке. При таком подходе используется свойство структуры формата графического файла и возможность представить графическую картинку приблизительно схожую по качеству отображения (для восприятия человеческим глазом) несколькими способами. Поэтому, кроме степени или величины сокращения, в таких алгоритмах возникает понятие качества, т.к. исходное видео картинка в процессе сокращения изменяется. Под качеством можно понимать степень соответствия исходного и результирующего видео картинка. Для графических файлов такое соответствие определяется визуально, хотя имеются и соответствующие формализованные методики и оценки [4]. Необратимое сокращённого объема невозможно применять в областях, в которых необходимо иметь точное соответствие информационной структуры входного и выходного потоков.

Способы сокращения без потерь используются в основном в научных и медицинских приложениях, когда потеря информации

недопустима или сами шумы видео картинка являются главной информацией, например в системах оценки качества оптоэлектронных систем. Коэффициент сокращения, достигаемый этими методами не более 1,5 для реальных сцен. Способы сокращения с потерями позволяют получить существенно большие коэффициенты сокращения.

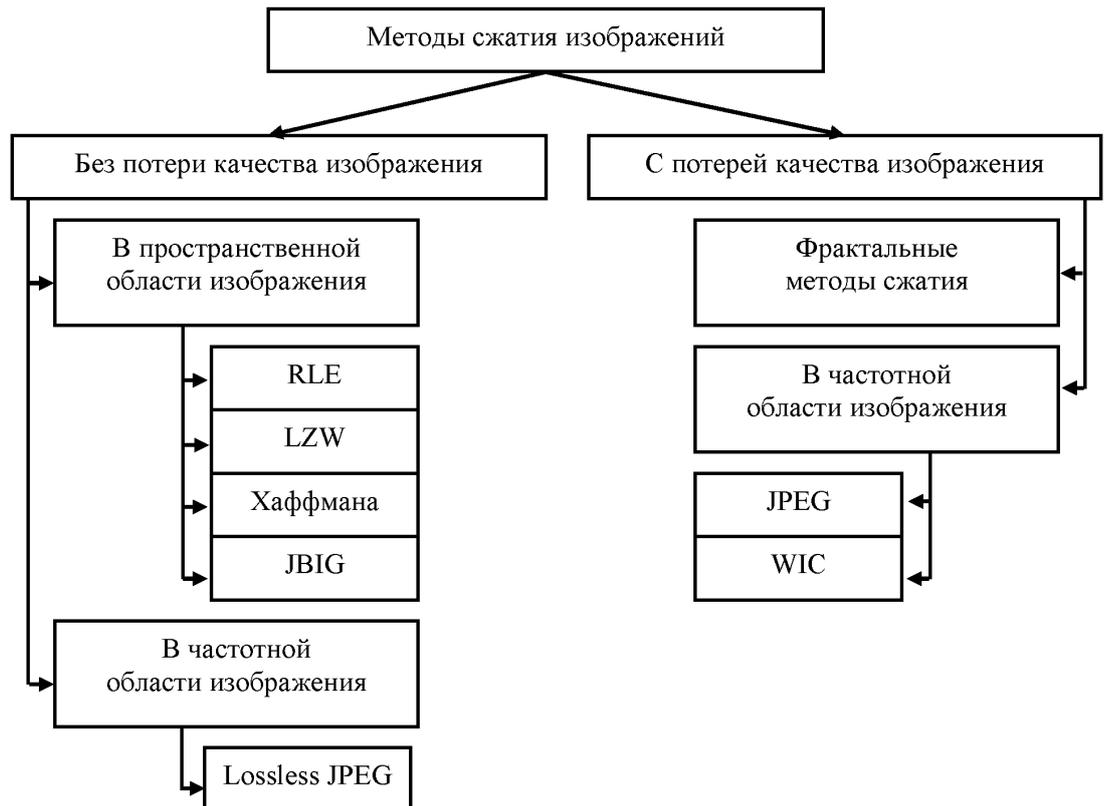


Рисунок.1.3. Классификация способов сокращения видео картинки

Однако при этом происходит искажение исходного видео картинка, ухудшение его качества. В связи с этим при сравнении различных способов сокращения помимо коэффициента сокращения нужно учитывать качество начальный вид видео картинка.

Для симметричных способов сокращения процедуры сокращения и начальный вид однотипны. Время сокращения и начальный вид для таких способов сравнимы. Для несимметричных способов процедура

сокращения отличается от процедуры начальный вид и обычно занимает большее машинное время.

Определим основные величины, характеризующие метод сокращения.

Оценка качества декодированного видео картинка.

Одна из проблем машинной графики заключается в том, что до сих пор не найден адекватный критерий оценки потерь качества видео картинка [4].

Качество теряется при оцифровке, при переводе в ограниченную палитру цветов или в другое цветовое пространство, а так же при сжатии видео картинка с потерями.

Пусть есть два видео картинка: $f(x, y)$ - оригинал, и $\hat{f}(x, y)$ - восстановленное видео картинка размером $M \times N$, тогда одним из простых словами оценки потери качества является среднеквадратическое отклонение значений пикселей сокращённого видео картинка от оригинала:

$$d(x, y) = \sqrt{\frac{\sum_{x,y}^{M,N} (f(x, y) - \hat{f}(x, y))^2}{M \cdot N}} \quad (1.1)$$

По этому критерию видео картинка будет сильно испорчено при изменении яркости всего на 5%. В тоже время видео картинка со снегом, резким изменение цвета отдельных точек будут признаны почти не изменившимися.

Другими словами является максимальное отклонение от оригинала:

$$d(x, y) = \max_{x,y} |f(x, y) - \hat{f}(x, y)| \quad (1.2)$$

Даная мера крайне чувствительна к биению отдельных пикселей, т.е. в видео картинка может измениться только один пиксель, и данный критерий признает видео картинка сильно испорченным.

На практике используемой мерой качества видео картинка является критерий соотношения сигнал/шум (PSNR).

$$d(x, y) = 10 \log_{10} \frac{255^2 \cdot M \cdot N}{\sum_{x,y} (f(x, y) - \hat{f}(x, y))^2} \quad (1.3)$$

Эта мера аналогична среднеквадратическому отклонению, но пользоваться ей удобнее из-за логарифмического масштаба шкалы.

Лучше всего потери в качестве оценивает человеческий глаз. Сокращённого объема видео картинка можно считать отличной, если на глаз невозможно отличить оригинал от сокращённого видео картинка. Но на практике при сжатии с потерями в видео картинка всегда вносятся какие-либо искажения заметные при сравнении оригинала и сокращённого видео картинка.

К другим наиболее употребляемым критериям оценки качества видео картинка относятся:

Средняя разность:

$$AD = \sum_{x=1}^M \sum_{y=1}^N [f(x, y) - \hat{f}(x, y)] / (MN) \quad (1.4)$$

Коэффициент кросс-корреляции:

$$K = \sum_{x=1}^M \sum_{y=1}^N f(x, y) \cdot \hat{f}(x, y) / \sum_{x=1}^M \sum_{y=1}^N [f(x, y)]^2 \quad (1.5)$$

Верность видео картинка (image fidelity):

$$AD = 1 - \left(\sum_{x=1}^M \sum_{y=1}^N [f(x, y) - \hat{f}(x, y)]^2 / \sum_{x=1}^M \sum_{y=1}^N [f(x, y)]^2 \right) \quad (1.6)$$

Среди большого числа критериев оценки качества видео картинка в работе для оценки качества восстановленного видео картинка были выбраны среднеквадратическое отклонение и соотношение сигнал/шум (как наиболее распространённые критерии), а для визуальной оценки используется разностное видео картинка.

1.3.Алгоритмы сокращения видео картинки без потерь

Сокращённого объема способом кодирования серий (RLE)

Наиболее известный и простой алгоритм сокращения информации обратимым путем - это кодирование серий последовательностей (Run Length Encoding -RLE) [3,7]. Суть данного подхода состоит в замене цепочек или серий повторяющихся байтов или их последовательностей на один кодирующий байт и счетчик числа их повторений. Проблема всех аналогичных способов заключается лишь в определении способа, при помощи которого распаковывающий алгоритм мог бы отличить в результирующем потоке байтов кодированную серию от других - некодированных последовательностей байтов. Решение проблемы достигается обычно простановкой меток вначале кодированных цепочек. Такими метками могут быть, например, характерные значения битов в первом байте кодированной серии, значения первого байта кодированной серии и т.п. Лучший, средний и худший коэффициенты сокращения -1/32, 1/2, 2/1.

Данные способы, как правило, достаточно эффективны для сокращения растровых графических видео картинки (BMP, PCX, TIFF), т.к. последние содержат достаточно длинных серий повторяющихся последовательностей байтов.

Недостатком метода RLE является достаточно низкая степень сокращения или стоимость кодирования файлов с малым числом серий и, что еще хуже - с малым числом повторяющихся байтов в сериях. К положительным сторонам алгоритма, пожалуй, можно отнести только то, что он не требует дополнительной памяти при работе, и быстро выполняется. Интересная особенность группового кодирования в формате PCX заключается в том, что степень архивации для некоторых

видео картинки может быть существенно повышена всего лишь за счет изменения порядка цветов в палитре видео картинки.

1.4. Сокращённого объема по методу Хаффмана

Для текстовых файлов чаще других употребляется кодировка Хаффмана, заключающаяся в том, что элементы текста заменяются цепочками бит разной длины. Методика Хаффмана гарантирует однозначное построение кода с наименьшим для данного распределения вероятностей средним числом элементов на букву [4].

Применительно к сжатию видео картинки в основе такого метода лежит учет частоты появления одинаковых байт в видео картинке. При этом пикселям исходного видео картинка, которые встречаются большее число раз, сопоставляется код меньшей длины, а встречающимся редко - код большей длины (т.е. формируется префиксный код переменной длины). Для сбора статистики требуется два прохода по файлу - один для просмотра и сбора статистической информации, второй - для кодирования [3]. Коэффициенты сокращения: 1/8, 2/3, 1.

При использовании такого метода требуется запись в файл и таблицы соответствия кодируемых пикселей и кодирующих цепочек. Такое кодирование применяется в качестве последнего этапа архивации в JPEG. Способы Хаффмана дают достаточно высокую скорость и умеренно хорошее качество сокращения.

Основным недостатком данного метода является зависимость степени сокращения от близости вероятностей элементов к величине 2^{-m} , поскольку каждый элемент кодируется целым числом бит. Так, при кодировании контента с двухэлементным алфавитом сокращённого объема всегда отсутствует, т.к. несмотря на различные вероятности появления элементов во входном потоке алгоритм фактически сводит их до 1/2. Такой алгоритм реализован в формате TIFF.

1.5. Алгоритм Лемпеля-Зива-Велча (Lempel-Ziv-Welch -LZW)

Данный алгоритм отличают высокая скорость работы как при упаковке, так и при распаковке, достаточно скромные требования к памяти и простая аппаратная реализация. Недостаток - низкая степень сокращения по сравнению со схемой двухступенчатого кодирования. Алгоритм преобразует поток элементов на входе в поток индексов ячеек словаря на выходе. Существует довольно большое семейство LZW - подобных алгоритмов, различающихся, например, методом поиска повторяющихся цепочек [7].

Коэффициенты сокращения: 1/1000, 1/4, 7/5. Коэффициент 1/1000 достигается только на одноцветных видео картинках размером больше 4 Мб. Ситуация, когда алгоритм увеличивает видео картинки, встречается крайне редко. Сокращённого объема обеспечивается за счет одинаковых подцепочек в потоке. Алгоритм является почти симметричным, при условии оптимальной реализации операции поиска строки в таблице.

LZW универсален - именно его варианты используются в обычных архиваторах. Он реализован в форматах GIF, TIFF и TGA.

1.6. Алгоритм JBIG и алгоритм Lossless JPEG

Алгоритм разработан группой экспертов ISO (Joint Bi-level Experts Group) специально для сокращения однобитных черно-белых видео картинки (например, для факсов или отсканированных документов). В принципе может применяться и к 2-х, и к 4-х битовым картинкам. При этом алгоритм разбивает их на отдельные битовые плоскости. JBIG позволяет управлять такими параметрами, как порядок разбиения видео картинка на битовые плоскости, ширина полос в видео картинке, уровни

масштабирования. Последняя возможность позволяет легко ориентироваться в базе больших по размерам видео картинки, просматривая сначала их уменьшенные копии. Настраивая эти параметры, можно использовать интересный эффект при получении видео картинка по сети или по любому другому каналу, пропускная способность которого мала по сравнению с возможностями процессора. Распаковываться видео картинка на экране будет постепенно, как бы медленно "проявляясь". При этом человек начинает анализировать картинку задолго до конца процесса разархивации. Алгоритм построен на базе Q-кодировщика, патентом на который владеет IBM. Q-кодер также, как и алгоритм Хаффмана, использует для чаще появляющихся элементов короткие цепочки, а для реже появляющихся длинные. Однако, в отличие от него, в алгоритме используются и последовательности элементов. Характерной особенностью JBIG является резкое снижение степени сокращения при повышении уровня шумов исходного видео картинка [3,7].

Алгоритм Lossless JPEG

Этот алгоритм разработан группой экспертов в области фотографии (Joint Photographic Expert Group). В отличие от JBIG, Lossless JPEG ориентирован на полноцветные 24-битные видео картинка [3,4]. Коэффициенты сокращения: 1/20, 1/2, 1.

Стандарт сокращения видео картинка JPEG включает два способа сокращения: первый предназначен для сокращения без потерь, второй – сокращения с потерей качества. Метод сокращения без потерь, используемый в стандарте lossless JPEG основан на методе разностного (дифференциального) кодирования. Основная идея дифференциального кодирования состоит в следующем. Обычно видео картинка характеризуются сильной корреляцией между точками видео картинка. Этот факт учитывается при разностном кодировании, а именно, вместо

сокращения последовательности точек видео картинка x_1, x_2, \dots, x_N , сжатую подвергается последовательность разностей $y_i = x_i - x_{i-1}$, $i = 1, 2, \dots, N$, $x_0 = 0$. Числа y_i называют ошибками предсказания x_i . В стандарте losslessJPEG предусмотрено формирование ошибок предсказания с использованием предыдущих закодированных точек в текущей строке и\или в предыдущей строке.

Lossless JPEG рекомендуется применять в тех приложениях, где необходимо побитовое соответствие исходного и разархивированного видео картинки.

Попробуем сделать некоторые обобщения. С одной стороны, приведенные алгоритмы достаточно универсальны и покрывают все типы видео картинки, с другой - они, по сегодняшним меркам, обеспечивают слишком маленький коэффициент архивации. Используя один из алгоритмов без потерь, можно обеспечить коэффициент архивации видео картинка примерно в два раза. В то же время сокращения с потерями оперируют с коэффициентами 10-200 раз. Помимо возможности модификации видео картинка, одна из основных причин подобной разницы заключается в том, что традиционные алгоритмы ориентированы на работу с цепочкой. Они не учитывают так называемую "когерентность областей" в видео картинках. Идея когерентности областей заключается в малом изменении цвета и структуры видео картинка на небольшом участке. Все алгоритмы, о которых речь пойдет ниже, были созданы позднее специально для сокращения графики и используют эту идею.

2. СПОСОБЫ КОМПЕНСАЦИИ ДВИЖЕНИЯ ДИНАМИЧЕСКИХ ВИДЕО КАРТИНКИ

2.1 Общие положения

В телевидении видео картинка передаются в виде последовательности кадров, информация в соседних кадрах которых относительно мало изменяется (рисунки 2.1). По этому, если передавать только межкадровые изменения, то скорость передачи значительно сокращается. По этому, во всех системах потокового видео используется межкадровая обработка, обеспечивающая основное сокращённого объема видеопотока [5].

Наиболее просто межкадровую лишность можно устранить простой межкадровой разницей значений пикселей двух смежных кадров, как показано на рисунки 2.2. Однако, в реальных ТВ видео картинках такой метод обладает низкой эффективностью, поскольку, при перемещении объектов сцены или ракурса съемки, вычитаются значения пикселей разных координат. Поэтому для повышения эффективности межкадровой разницы применяют межкадровую разницу с компенсацией движения всего видео картинка или его отдельных фрагментов или блоков.

Для этой цели для фрагментов видео картинка 1 кадра ищется его наибольшее соответствие в следующем кадре в зоне поиска его предполагаемого смещения. Если такое соответствие найдено, то вместо содержимого этого фрагмента передаются только его координаты предполагаемого смещения (вектор смещения).

Кадры, относительно которых производится поиск соответствия их фрагментов в последующих кадрах, называются опорными (ключевыми).

Кадры, в которых посредством ключевых кадров производится компенсация движения, называются смежными.



Рисунок. 2.1. Видео картинки 2 смежных кадров.



Рисунок.2.2. Видео картинка межкадровой разницы.

Ключевые кадры подразделяются на фрагменты, по которым и производится поиск соответствия. К подобным фрагментам относятся: макроблоки (блоки, на которые подразделяются ключевые кадры, при наложении на них двумерной блочной сетки), а также объекты (предметы, фон, звук) при объектно – ориентированном кодировании (метод сегментации объектов). К стандартным размерам относятся макроблоки следующих размеров: 4x4, 8x8, 16x16, 32x32 пикселей.

После передачи ключевого кадра, состоящего из блоков (макроблоков), осуществляется поиск соответствия между блоками ключевого и смежного кадров. При обнаружении подобного соответствия в смежном кадре в MPEG-4 передается не весь блок, а лишь его координаты. Эта тактика очень удобна, так как видео картинка часто включают изменяющиеся объекты в видео потоке, как, например, фон видео картинка.

Однако, при блочном кодировании часто возникает ситуация, когда множество блоков в кадровой последовательности составляют какой-либо объект со сложным контуром, передвигающийся от кадра к кадру как группа блоков. В этом случае намного экономичнее и удобнее использовать кодирование всего объекта, а не составляющих его блоков.

Видео-кодек производит поиск объектов, запоминание их контура и кодирование пикселей внутри контура объекта. Информация о цветности и яркости пикселей внутри контура объекта кодируется. Контур объекта запоминается и передается вместе с бинарной альфа-картой (бинарное представление объекта), которая используется для поиска местоположения этого же объекта в смежных кадрах, а также, фиксирует изменение размеров объекта или его формы [10].

Затем, передается информация о цвете и яркости пикселей внутри этого объекта. Таким образом, в смежных кадрах видео потока производится поиск объектов, аналогичных кодированным объектам в ключевых кадрах. При их кодировании передается информация о координатах перемещения данного объекта, изменении масштаба или формы.

Все объекты подразделяются на объекты переднего и заднего плана. Даже, если в какой-то промежуток объект переднего плана частично закрывает собой объект заднего плана, видео кодек все равно должен различить их как два самостоятельных объекта. Это очень важно для поиска объектов и правильного определения их контура, на основе которого будет кодироваться вся информация о цветности пикселей внутри контура. Данный метод уже применяется в кодеке MPEG-4, однако, в разрабатываемом видео-кодеке MPEG-7 способы объектно-ориентированного кодирования будут существенно улучшены [3].

Механизм компенсации движения относительно прост в реализации, предусматривает использование множества способов устранения временной лишности и позволяет существенно увеличить коэффициенты сокращения потокового видео за счет устранения временной лишности. Все это делает механизм компенсации движения очень выгодным для потокового видео.

2.2. Сопоставление блоков

В видео кодеках поколения MPEG применяется метод сопоставления блоков, как один из способов компенсации движения (VQ-векторная квантизация). Он включает в себя два основных принципа: деление кадра на блоки (размеры блоков 4x4, 8x8 и 16x16 и другие) и поиск подобных блоков, как частей видео картинка, в других кадрах. На рисунке 2.1 видео картинка делится на блоки размером 16x16 пикселей. Размер данного видео картинка не кратен количеству блоков. Видео картинка имеют разные размеры и не всегда кратны количеству блоков, выбранного размера.

Основная идея данного метода заключается в разбиении видео картинка на макроблоки. При этом, некоторые макроблоки оказываются похожими друг на друга.



Рисунок 2.1. Деление видео картинка на блоки 16x16 пикселей.

В этом случае компрессор идентифицирует класс похожих блоков и заменяет их одним общим блоком. После этого генерируется двоичная таблица (карта или кодовая книга) таких общих блоков из самых коротких кодовых слов (двоичные коды). VQ-декодер, используя таблицу, собирает общее видео картинки по этим блокам. Потери в качестве при использовании данного метода кодирования объясняются тем, что схожесть блоков лишь относительна [5].

Процесс кодирования отличается своей длительностью и трудоемкостью, так как кодеру необходимо выявлять принадлежность каждого блока видео картинка к какому-нибудь общему блоку. При этом, задача декодирования сводится к задаче построения видео картинка по заданной карте из общих блоков и не занимает много аппаратных и временных ресурсов. Наибольшее сокращённого объема с использованием алгоритма VQ можно достичь путем уменьшения числа классов общих блоков. Это достигается за счет более обширной аппроксимации блоков на основании их схожести. В результате происходит ухудшение качества воспроизводимого видео.

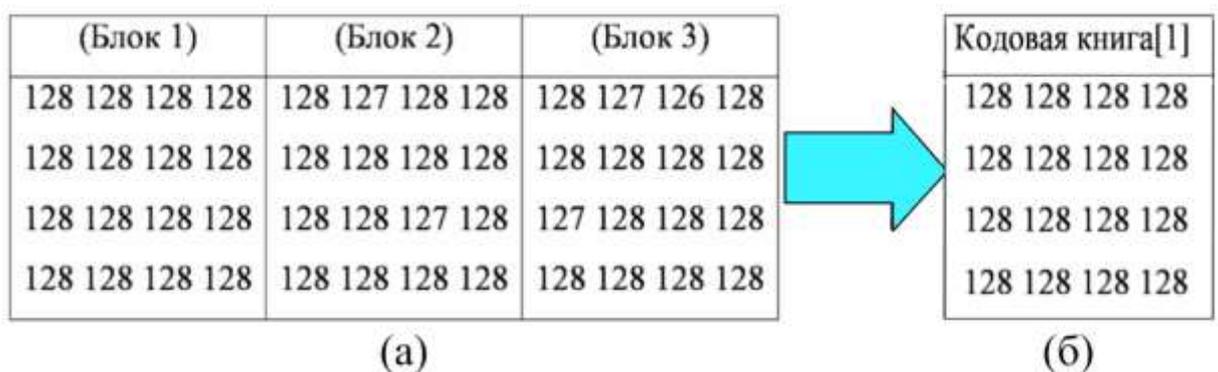


Рисунок 2.2 Пример сопоставления 3 блоков 4x4 (а) с последующей их заменой одним общим блоком (б).

На рисунке 2.2 (а) показаны три блока, различия между которыми для человеческого глаза неотличимы. Поэтому, 2-ой и 3-ий блоки можно заменить первым блоком (б).

В результате такой аппроксимации получается некоторая необратимая потеря информации и появляется блочная структура видео картинка.

2.3. Способы выделения объектов в ТВ видео картинках

При обработке потокового видео часто возникают ситуации, когда намного удобнее обрабатывать объекты кадра целиком, а не разбивать видео картинку на блоки. Данная методика выделения объектов носит название – сегментация объектов. Подобный метод стал нужен для увеличения качества видео картинка и эффективности его сокращения.

Метод выделения объектов предполагает наличие функций, описывающих взаимодействие этих объектов, построение сложных иерархий, создание мультимедиа сцены с отдельным кодированием видео объектов и звука. Подобный метод увеличивает эффективность и возможности MPEG-4 как для специалистов, так и для пользователей, и выводит MPEG-4 на новый уровень по сравнению с MPEG-1 и MPEG-2.

В кадре объект перемещается по кадру целостной структурой, то его составляющие макроблоки имеют одинаковые векторы движения, ошибки предсказания движения, и перемещаются единой группой. На основе этих свойств объектов механизм компенсации движения предусматривает метод сегментации объектов для повышения эффективности сокращения [10].

При этом, сегментация объектов используется в MPEG-4 как один из способов механизма компенсации движения, основанного на ДКП. Сегментация объектов производится наряду с другими методами (метод

параметрических моделей, метод сопоставления блоков и другие). Однако, в отличие от других методик компенсации движения, метод сегментации объектов может быть согласован с применением вейвлет-преобразований. В этом случае, сам объект будет кодирован целиком посредством вейвлет-изменения формы. На данный момент существует ряд подобных кодеков, использующих сегментацию объектов как методику компенсации движения, основанную на вейвлет изменении форм. Однако, подобные видео-кодеки не являются широкопользовательскими, так как требуют много времени на обработку видео – потока.

Метод сегментации движения построен на нескольких алгоритмах, и довольно сложен в реализации. Это связано с некоторыми сложностями выделения объектов из общего фона:

- 1) Во время движения объекты переднего плана могут перекрывать объекты заднего плана.
- 2) Яркость и цветность объекта может меняться от кадра к кадру (падение тени).
- 3) Контур объекта может меняться от кадра к кадру (параметрические модели, вращательное движение самого объекта).

Однако все эти изменения происходят поэтапно и постепенно, поэтому, благодаря предсказанию ошибки MPEG-4 распознает изменившийся объект.

Существуют два стандарта, по которым производится сегментация видео картинки:

- 1) сегментация на контекстной основе;
- 2) сегментация посредством bitmap (битовая карта) метода;

При применении сегментации на контекстной основе система распознавания объектов сводится к распознаванию контура объекта. В

этом случае контур объекта представляется в качестве замкнутой кривой [4]. С точки зрения алгоритмов сегментирования, каждый объект является многослойным. К слоям объекта относятся:

- 1) контур объекта;
- 2) текстура объекта (цветность и яркость);
- 3) бинарная альфа карта объекта;

Текстура объекта представляется тремя цветными компонентами YUV. Слои объекта кодируются отдельно.

Бинарная альфа карта объекта создается посредством контура объекта. В этом случае все пиксели, внутри контура приравниваются к единице и становятся белыми, а пиксели вне контура приравниваются к нулю и становятся черными. Подобные пиксели еще называются прозрачными. Их нулевое значение означает принадлежность к фону видео картинка.

В итоге, получается бинарное представление объекта, состоящее из двух цветов: черного и белого. Подобная альфа карта помогает найти локальное расположение контура объекта в кадре, после чего производится восстановление текстуры внутри контура.

Визуальные спецификации MPEG-4 поддерживают несколько типов визуальных объектов, среди которых **VideoObject** (VO – Видео-Объект). Визуальные объекты в видео потоке могут быть представлены как последовательность двумерных образов, где каждый образ может иметь пространственную поддержку произвольной формы, определяющую их положение в кадре. В качестве подобной пространственной поддержки обычно выступает бинарная альфа карта объекта. Особым случаем в VideoObject является, когда пространственная поддержка осуществляется в прямоугольной форме, и когда размер и позиция объекта изменяется соответственно времени.

Выделение объекта производится по следующим критериям [9]:

- 1) выделение по яркости;
- 2) выделение по цвету;
- 3) выделение по текстурным признакам;

Наиболее простым и часто используемым методом выделения объектов является выделение по яркостным характеристикам. При использовании этого метода выделения объектов или фаз производится выбор всех точек видео картинка, попадающих в определенный интервал яркости. Такой отбор осуществляется по гистограмме яркости для заданного видео картинка. Оцифрованное видео картинка представляет собой набор точек, яркость которых (для 8ми битных видео картинка) лежит в интервале от 0 (абсолютно черный цвет) до 255 (абсолютно белый). Гистограмма яркости видео картинка показывает процентное содержание точек определенной яркости. Выделение объектов по яркости обычно производится методом установки пороговых значений яркости, и все точки видео картинка, лежащие в пределах контента значений, будут выделены на видео картинка как маски объектов, которые и будут затем измеряться.

Вторым указанным методом является метод выделения объектов по цвету. Он производится с использованием цветных моделей и является более сложным, чем простое выделение по яркости. Данный метод позволяет добиться лучших результатов в тех случаях, когда объекты с одинаковыми яркостными характеристиками отличаются по цвету. В современных анализаторах видео картинка выделение объектов видео картинка по цвету реализуется с использованием метода обучения, когда пользователь системы указывает участки видео картинка, которые необходимо выделять. Обученная таким образом система на следующем видео картинка будет автоматически выделять объекты, соответствующие по цвету указанным участкам.

Выделение по текстурным особенностям наиболее близко к тем механизмам упорядочивания объектов, которые использует человеческий мозг при их распознавании. Этот принцип состоит в опознании объекта или фазы не по яркости или цветовым характеристикам, а по внешним признакам или текстурным особенностям. Это относится к таким ситуациям, когда необходимо выделить слой (участок) с ориентированной структурой внутри однородной структуры, которая не отличается от слоя ни по яркости, ни по цвету, но отличается исключительно по внешнему виду. Этот способ выделения наиболее сложен для реализации, и далеко не все существующие в настоящее время анализаторы видео картинки способны решать задачу выделения по текстурным признакам.

В результате, работа MPEG-4 с видео картинками определяется тем фактом, что объект может иметь любую форму, и, кроме того, форма, размер и положение объекта может измениться от одного кадра к другому. Согласно его общему представлению, **Video Object** составлен из трех цветных компонентов (YUV) и альфа компоненты. С помощью YUV компонентов можно выразить изменения яркости и цветности структуры внутри контура объекта. Альфа компонента лишь помогает выделенному контуру объекта найти свое положение в кадре. После этого контур объекта заполняется текстурой. Бинарная альфа карта, контур и текстура передаются последовательно как элементы Visual Object Plane. На ключевом кадре объект полностью восстанавливается. Подобным образом передается несколько десятков кадров, где сегментированные объекты восстанавливаются с помощью алгоритмов сегментации.

Способ кодирования бинарных объектов может использоваться для взаимодействующих друг с другом мультимедиа объектов. Построенные

на бинарных объектах, они представляют собой текстурные объекты, имеющие свою бинарную форму, как показано на Рисунке 2.3;

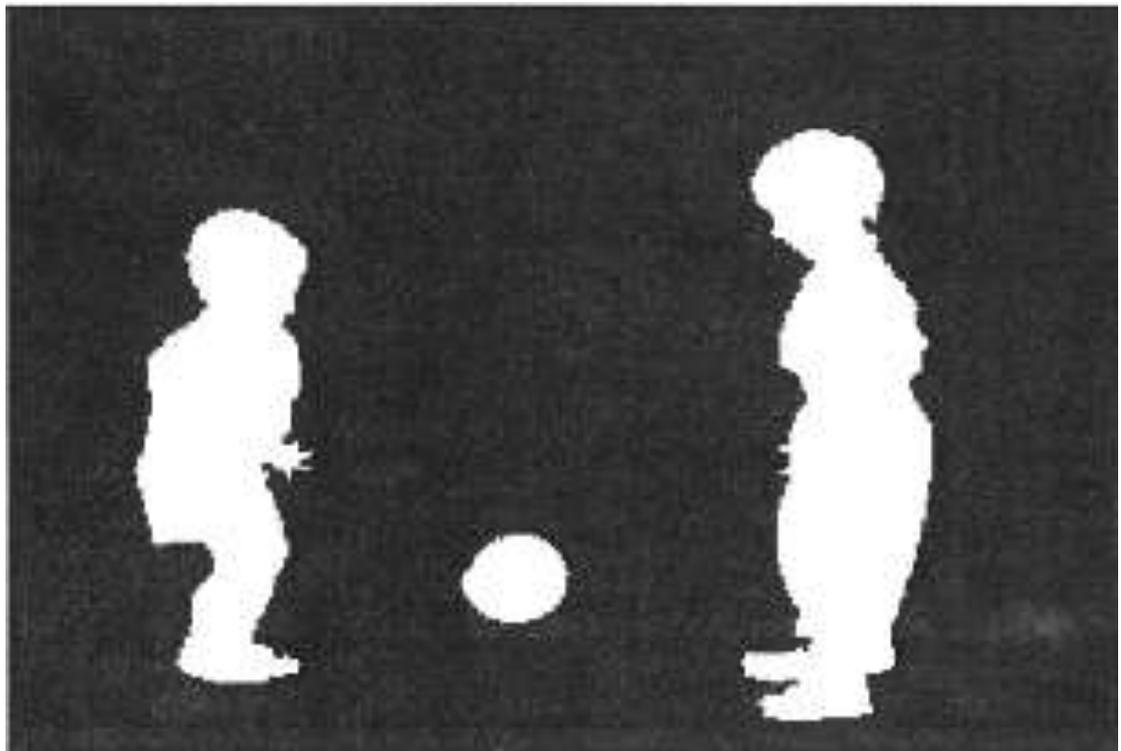
Текстуры объекта и его форма представляется с использованием палитры серого уровня. При этом, для этого объекта, альфа карта представляет собой образ, состоящий из палитры серого цвета с 256 возможных уровней цветности пикселей. Эта информация альфы серого уровня используется для определения характеристик прозрачности объекта в течении процесса видео обработки, где понятие прозрачности и непрозрачности пикселя определяется степенью яркости пикселя [15].

В этом случае, абсолютно черный пиксель считается прозрачным, а белый пиксель не прозрачным. Тогда пиксель какого-либо оттенка серого обладает некоторой степенью прозрачности. Подобная маска прозрачности пикселей работает как своего рода фильтр, и помогает сглаживать градиционные переходы между контуром объекта и фоном.

Необходимость данного фильтра исходит из того, что объекты не всегда имеют четкий контур. Возникают погрешности, которые могут быть вызваны тем, что алгоритм не всегда правильно причисляет граничащие пиксели к объекту или к фону, окружающему данный объект. Это характерно для граничных пикселей, обладающих большой степенью прозрачности. Основной задачей является сегментирование всех пикселей, принадлежащих объекту.



а)



б)

Рисунок 2.3. Исходный кадр с несколькими движущимися объектами(а) и его бинарная альфа карта (б).

Как правило, сегментация происходит с некоторой лишностью. Это связано с тем, что могут возникнуть большие искажения, если несколько граничных пикселей объекта будут причислены к фону, а не к контуру данного объекта.

Сегментированные объекты заносятся в память, где, после обработки необходимой части видео-потока между ними устанавливаются иерархии и соотношения [10]. Это производится путем вычисления глобального и локального движения объектов в кадре с последующим использованием этой информации для определения позиций объектов переднего плана на кадрах. Это упрощает кодирование, особенно при перекрытии объектов заднего плана объектами переднего плана, как показано на Рисункунок 2.4;



(a)

(б)

Рисунок 2.4. Исходный кадр(а) и результат сегментации объектов(б).

При применении контента способов также применяется сглаживание межкадровой разницы по траектории контура. Это связано с тем, что при движении объекта по любой траектории возникает проблема в отображении участка фона. Двигаясь, объект, закрывает один участок

фона, но, переместившись, освобождает некоторую область. Эта область заполняется текстурой фона. Это происходит следующим образом:

1) производится обращение к следующему ключевому кадру, где данный объект находится в другой области кадра;

2) с помощью межкадровой разницы обнаруживается необходимый участок фона;

3) выделенный участок ставится в пустующую область кадра, расположенную позади двигающегося объекта;

Алгоритм выделения контуров сравнительно новая технология [11]. Однако, эффективность данной методики позволяет ей найти широкое применение в области цифрового телевидения. Методика на контурной основе позволяет осуществить поиск и трекинг (то есть определение параметров движения заданного объекта относительно рамки кадра на основании сопоставления двух близких кадров, содержащих этот объект) объектов переднего плана в видео. Это производится путем вычисления глобального и локального движения объектов в кадре с последующим использованием этой информации для определения позиций объектов переднего плана на кадрах. Существенным достоинством данного метода является возможность правильного определения положения объектов переднего плана даже в случае очень медленного движения.

Другие важные достоинства включают:

1) регулируемое соотношение скорость-качество;

2) наличие нескольких уровней точности сегментации;

3) нет необходимости в ручной сегментации;

В отличие от подхода на контурной основе, bitmap способы применяются непосредственно к самому исходному бинарному изображению [12]. Данные способы были приспособлены для работы в пределах обычного видео блочной структуры. Способы на контурной основе в то время еще не были изобретены.

В случае применения bitmap сегментирования видео объекты, рассматриваются как прямоугольные видео картинка (например, обычные ТВ кадры), кодируемые с использованием гибридного кодирования. Bitmap включает предсказание с оценкой и компенсацией движения для макроблоков 16x16 пикселей, и ДКП ошибки предсказания в блоках 8x8 пикселей.

Метод предсказания заключается в том, что на область видео-объекта накладывается двухмерная сетка (как показано на рисунке 2.5), которая делит видео картинку на макроблоки. Сам объект определяется из относительного сходства векторов предсказания движения макроблоков в кадре. Это связано с тем что, макроблоки, образующие объект передвигаются группой и имеют однотипные векторы движения. Подобные группы макроблоков начинают рассматриваться как единое образование (объект).

После обнаружения объекта MPEG-4 производит поиск блоков объекта в следующих кадрах, согласно, “предсказанию” на основе вектора движения данного объекта и параметрических моделей. Это означает, что поиск блоков объекта в кадре происходит там, куда, вероятнее всего, согласно расчетам, переместится объект в следующем кадре.

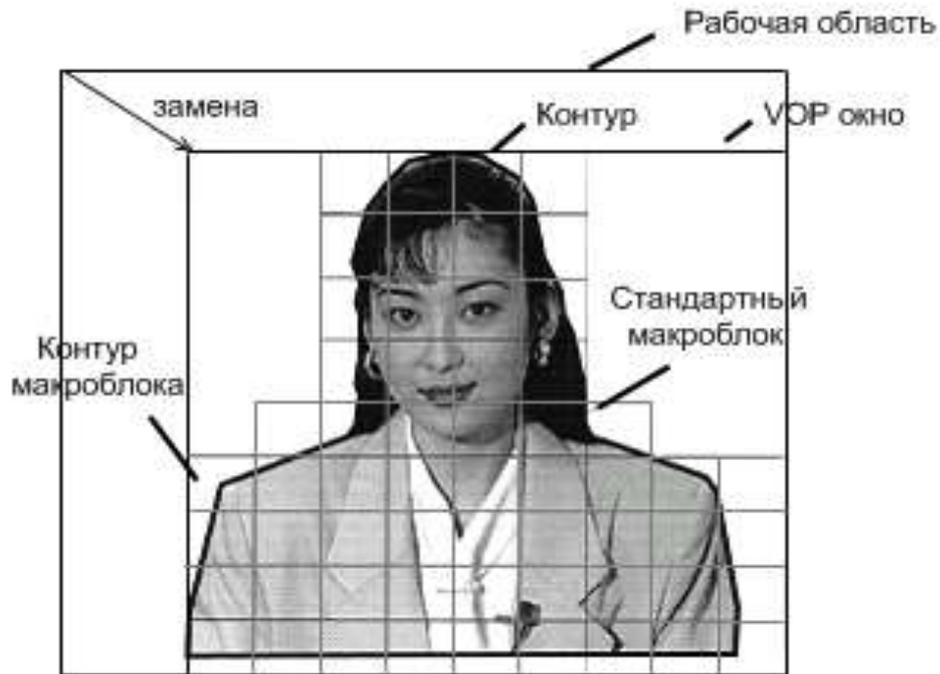


Рисунок.2.5. Двухмерная сетка, наложенная на видео-объект в кадре.

Затем, вокруг объекта выбирается прямоугольная область (рабочая область на рисунке 2.5). Рабочая область или область поиска представляет собой зону, в пределах которой будет осуществляться поиск соответствующих блоков объекта в последующих кадрах. Формирование подобной области сокращает область поиска блоков, что автоматически уменьшает время обработки видео-потока.

Данная методика также предполагает использование бинарных альфа карт. Однако в отличие от способов на контурной основе, битмар способы используют бинарные карты не всего видео картинка, а лишь его макроблоков. Подобная бинарная карта носит название – Бинарный Альфа Блок (Binary Alpha Block – BAV). Подобные альфа блоки помогают установить границу объекта.

При делении кадра на макроблоки возможны три ситуации [15]:

- 1) когда макроблок внутри контура объекта;
- 2) когда макроблок вне контура объекта;

3) когда макроблок содержит контур объекта;

Первые два случая достаточно просты в реализации. Если макроблок внутри объекта, то он передается вместе с остальными макроблоками, образующими объект, единой группой. В том случае, если макроблок находится вне контура объекта, то он просто не передается.

Самым сложным с точки зрения реализации является тот случай, когда контур объекта находится внутри макроблока. В данном случае, внутри макроблока осуществляется поиск контура объекта. Он основывается на градиентных алгоритмах. При этом, контур объекта рассматривается как резкая смена яркостей по сравнению с фоном. После обнаружения контура производится работа по аппроксимации полупрозрачных пикселей для получения более плавного контура объекта. Как и при работе с методами на контурной основе, bitmap способы используют маску прозрачности для каждого макроблока. Также, осуществляется подгонка граничных макроблоков, которая заключается в том, что прозрачные пиксели каждого граничного макроблока на ссылочном VOP экстраполируются (окрашиваются) по горизонтали и вертикали с помощью непрозрачных пикселей как показано на рисунках. 2.6 [15].

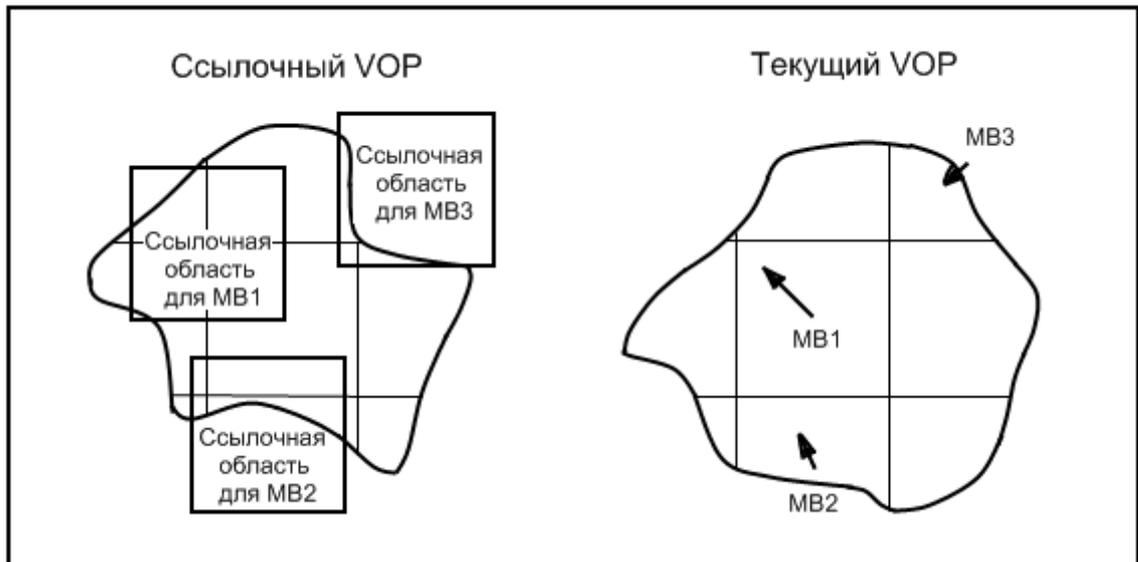


Рисунок. 2.6. Примеры ссылочных областей с прозрачными пикселями.

При подгонке по горизонтали, цвет непрозрачных пикселей по краям ВАВ используется для окраски прозрачных пикселей, расположенных на той же строке. Если группа пикселей белого цвета граничит с непрозрачными пикселями только с одной стороны строки, то ближайший непрозрачный пиксель копируется во все прозрачные пиксели. Также, возможна ситуация, когда строка пикселей ограничена непрозрачными пикселями с обеих сторон (например, верхняя строка на рисунке. 2.7 (а)). В этом случае, в качестве цвета прозрачных пикселей выбирается среднее значение двух соседних непрозрачных пикселей. Результат подобной аппроксимации продемонстрирован на рисунке. 2.7;

На втором этапе подгонки применяется экстраполяция по вертикали. При этом, непрозрачные пиксели (включая пиксели, заполненные цветом, на первой стадии подгонки) экстраполируются по вертикали для окраски оставшихся прозрачных пиксельных позиций. Столбец прозрачных пикселей, имеющих только один соседний пиксель, заполняется его

значениями, а столбцы, ограниченные непрозрачными пикселями с двух сторон, окрашиваются средним значением верхнего и нижнего непрозрачного пикселя этого столбца.

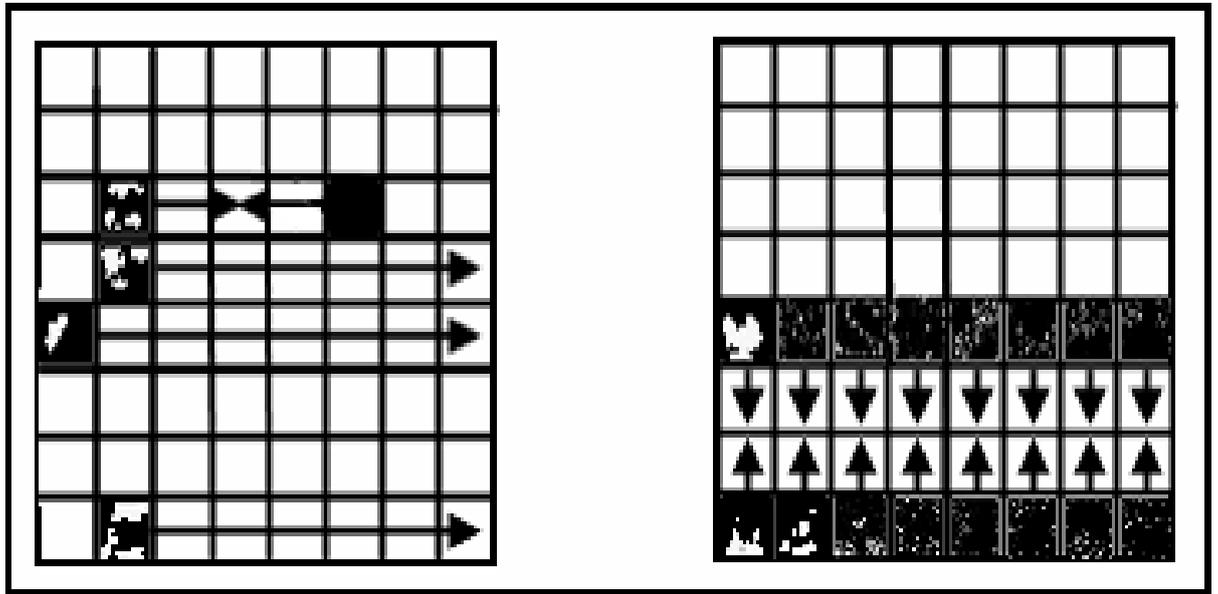


Рисунок 2.7. Подгонка по горизонтали и вертикали граничного макроблока.

Таким образом, определяется контур объекта. При этом, объект, восстановленный по данным альфа-карты, контура и текстуры не будет иметь ярко выраженных градационных переходов на границе. То есть, края объекта не будут сильно выраженными. Это объясняется подобной работой экстраполяционных механизмов выделения контура объекта.

Однако, из-за использования этих механизмов подгонки также присутствует некоторая вероятность ошибки. Это связано с тем, что присукнуноквоение граничным пикселям цветов соседних пикселей может стать причиной присукнуноквоения пикселей фона к объекту или наоборот. Способы на контурной основе позволяют избежать подобных ошибок.

В результате, задача данной выпускной квалификационной работы сводится к разработке объектно-ориентированного кодека, в котором механизм сокращения осуществлялся путем сегментации при помощи выделения границ объекта и последующего кодирования данного объекта.

3. АЛГОРИТМЫ СОКРАЩЕНИЯ ДИНАМИЧЕСКИХ ОБЪЕКТОВ В ВИДЕОКОДЕКЕ СТАНДАРТА MPEG-4

3.1. Общие положения

Как было установлено при анализе способов сокращения видео, для обеспечения больших коэффициентов сокращения видео в настоящее время широкое распространение начинают получать кодеки стандарта MPEG-4, основанные на сложных алгоритмах обработки видео картинки, способные распознавать объекты сцены и осуществлять интерактивное управление медиа-сценой.

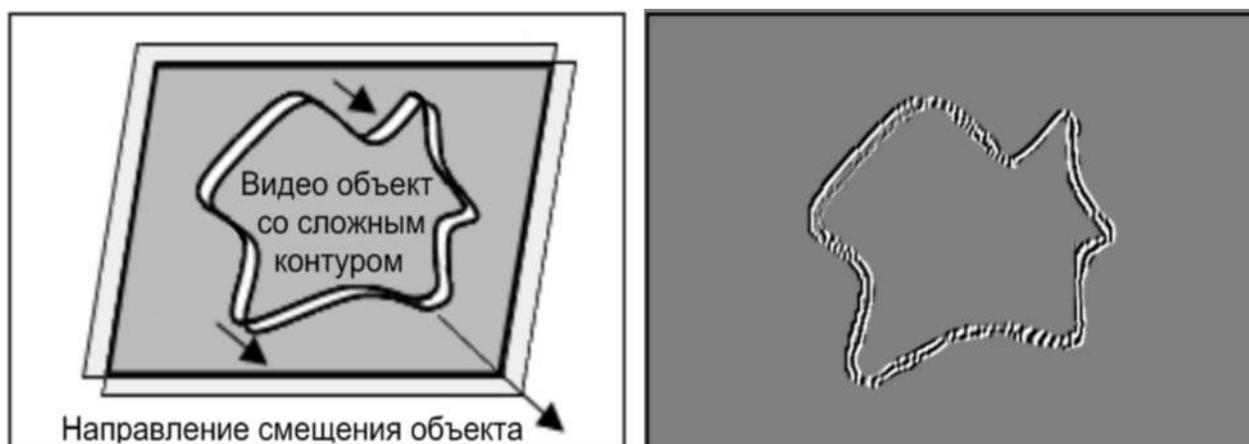
Однако, высокоточная сегментация объектов представляет собой весьма сложную задачу. К тому же, на практике чаще всего встречаются динамические объекты, контур которых не остается постоянным в процессе движения. Алгоритмы сегментации подобных объектов требуют больших затрат времени и вычислительных ресурсов, и в настоящее время еще недостаточно изучены. Поэтому, ставится цель снизить затраты на разработку видео кодека путем применения объектно-ориентированного подхода для более точного определения границ динамических объектов при компенсации движения. При этом, производится сегментирование динамического объекта путем определения его изменяющегося контура посредством нестационарного двумерного уравнения диффузии. В результате этого, осуществляется разбиение кадровой последовательности на составные части, которые движутся и могут быть сжаты посредством механизма компенсации движения.

При выделении динамических видео картинки удобно пользоваться межкадровой разницей. Это связано с тем, что у динамических объектов контур может меняться от кадра к кадру, что делает сегментацию подобных объектов и их распознавание крайне сложной задачей. В случае вычисления межкадровой разницы (когда из ключевого кадра вычитается смежный) статичные участки кадра (фон видео сцены), которые остаются неизменными на протяжении некоторого времени, в ходе вычитания исчезают.

Те участки кадра, которые изменились в ходе движения объекта, проявляются в результате применения межкадровой разницы. В ходе вычитания проявляются также

те области фона, с которых исчез объект, и те области фона, где данный объект появился в ходе своего передвижения, как это показано на рисунке 3.1.

Однако, метод межкадровой разницы целесообразно применять в тех случаях, когда имеются подвижные объекты на стационарном фоне. В реальном видео редко можно наблюдать абсолютно неподвижный фон. Это связано с движением самой камеры. Поэтому, метод межкадровой разницы удобно применять к кадровой последовательности лишь после того, как она будет обработана посредством параметрических моделей. Как метод механизма компенсации движения, параметрические модели позволяют скомпенсировать движение видео камеры. После этого, становится целесообразным применение метода сегментации динамических объектов посредством межкадровой разницы.



(a)

(б)

Рисунок 3.1. Отображение движущегося объекта разностью двух его последовательных положений (a) и межкадровая разница при сдвиге объекта на 3 пикселя (б).

Кроме того, подобный метод не требует сложных вычислительных алгоритмов и достаточно прост с точки зрения программной реализации. Структурная схема обобщенного алгоритма подобной объектно-ориентированной обработки видео картинки представлена на рисунке 3.2.

Данная схема работает следующим образом: осуществляется загрузка в буфер памяти программы двух последовательных кадров. Первый кадр выбирается опорным, а последующий кадр становится

смежным. При этом, опорным кадром является начальный кадр видеопотока, относительно которого производится дальнейшая обработка видео картинки. Чем меньше составляет отличие смежного кадра от ключевого кадра, тем точнее будет определен контур.

Следующим шагом реализации программного алгоритма является поступление этих двух кадров на преобразователь цветного видео картинка RGB в яркостное YUV. Это связано с тем, что работать с разницей яркостных видео картинки намного легче, так как, в противном случае, после межкадровой разницы цветных видео картинки можно получить разноцветный контур, цветность которого будет меняться от одной группы пикселей к другой.

Затем, осуществляется межкадровая разница, в ходе которой от каждого пикселя первого кадра отнимается соответствующий пиксель тех же локальных координат во втором кадре. В результирующем окне программы показывается получившаяся межкадровая разница в виде некоторого контура, отображающего разность двух последовательных положений движущегося объекта. Однако, яркость данного контура может варьироваться от одной группы пикселей к другой, поэтому необходимо использовать яркостную пороговую обработку.

Пороговая обработка получившейся межкадровой разницы состоит в следующем: пользователем программы устанавливается некоторое пороговое значение, являющееся наиболее приемлемым для данного типа видео.

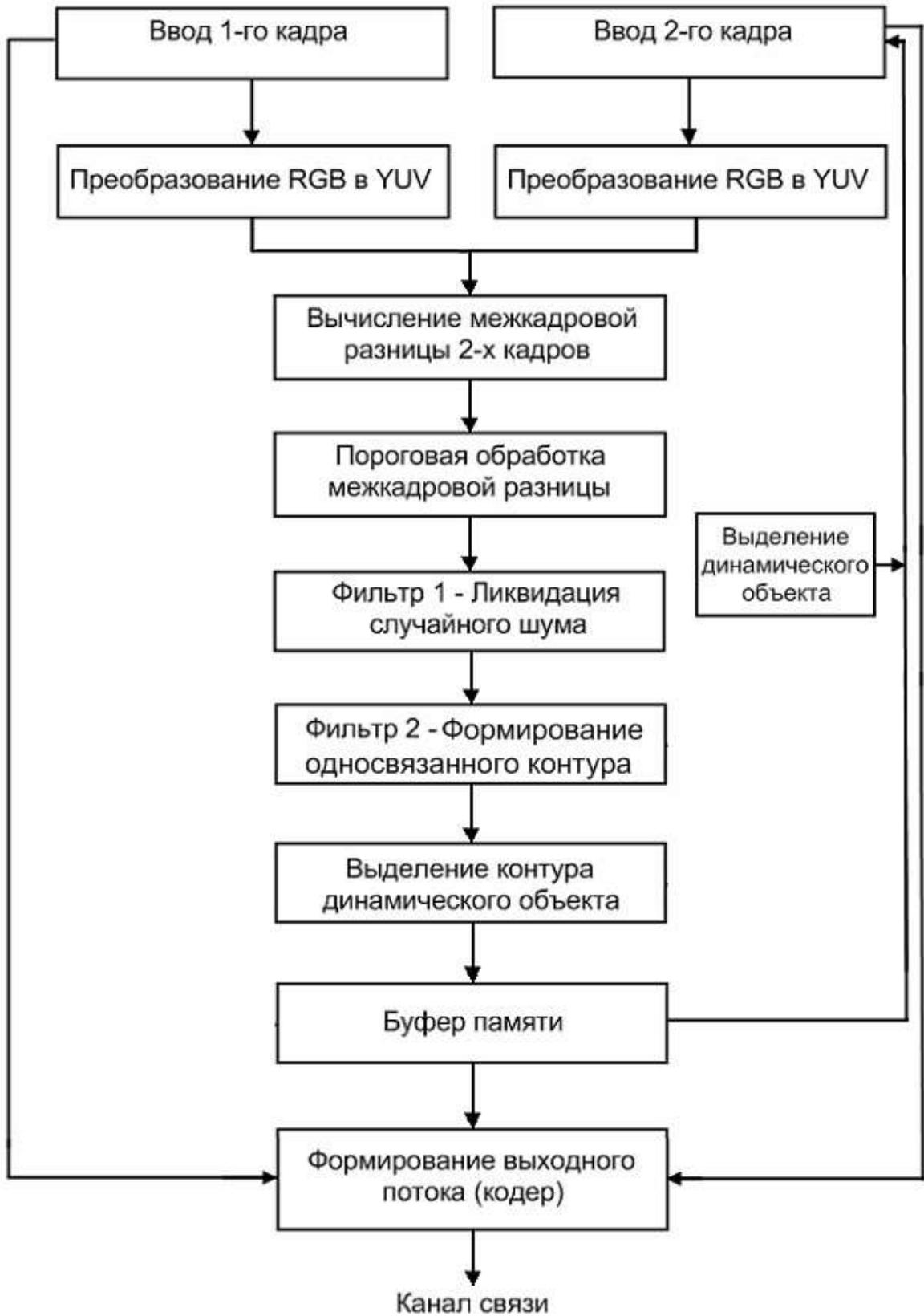


Рисунок.3.2. Структурная схема обобщенного алгоритма объектно-ориентированной обработки видео картинки.

В результате этого, пикселям, имеющим значения яркости выше пороговой величины, присуждается максимальное значение (255), а меньше – значение нуля. Таким образом, формируется черно-белое бинарное видео картинки имеющее, только два значения яркости – 0 или 1.

Тем не менее, в результирующем бинарном видео картинки крайне редко удается получить четкий контур динамического объекта. Чаще всего существует шум в виде случайных всплесков яркости из-за возможного “дрожания” фона, который отображается как несколько одиночных или небольших групп пикселей в кадре. Появление подобного шума связано с дефектами работы параметрических моделей.

Однако, данная программа не реализует параметрические модели, а лишь предполагает их использование для получения неподвижного фона в кадре. Поэтому, в используемых видео кадрах реальный динамический объект из видео фильма был искусственно перемещен на статичный неподвижный фон. Предполагается, что подобный статичный фон можно получить после использования параметрических моделей, однако в рамках данной выпускной квалификационной работы алгоритмы параметрических моделей не рассматриваются.

Использование подобного совмещения реального объекта с полностью статичным фоном предполагает полное отсутствие случайных шумов, возникающих при работе с параметрическими моделями. Поэтому весь присущий видео картинке случайный шум был искусственно создан для максимального приближения данного видео к реальному.

Также, другим недостатком является то, что внутри основного контура может присутствовать множество многосвязанных контуров. Их появление связано с неоднородностью текстуры объекта,

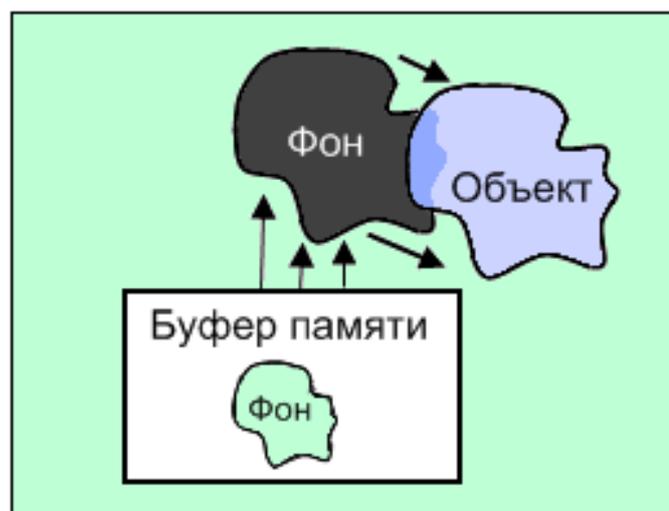
резкими изменениями цветности и яркости от одной части объекта к другой. Передвижение подобных областей объекта, как единого целого вызывает появление контента малых контуров.

Для устранения подобных дефектов применяются два фильтра. После пороговой обработки межкадровой разницы полученный кадр обрабатывается первым фильтром – Фильтр удаления случайных шумов. С помощью данного фильтра все единичные пиксели приравниваются к нулю (становятся черными). Обход всех пикселей совершается по горизонтали и по вертикали. Затем, обработанный фильтром удаления случайных шумов, кадр поступает на обработку вторым фильтром – Фильтр формирования односвязанного контура, построенного на нестационарном уравнении теплопроводности. С точки зрения данного фильтра получившееся видео картинка является тепловой картой среды, где белый цвет соответствует сильно нагретым областям, а черный цвет – холодным областям. После этого, относительно введенных пользователем параметров “нагрева” и порога “охлаждения” симулируется процесс охлаждения и нагрев среды согласно нестационарному уравнению теплопроводности из области термодинамики. В результате процесса “охлаждения” пиксели начинают приравниваться к нулю. Скорость процесса зависит от концентрации пикселей. Малые группы пикселей быстро исчезают (“охлаждаются”). Большие группы пикселей исчезают намного медленнее. Затем, по достижению некоторого порогового значения, включается процесс “нагрева”, и вокруг групп пикселей, которые не успели исчезнуть, за счет процесса аппроксимации черные пиксели приравниваются к значению соседних белых пикселей. Постепенно, до достижения определенного порогового значения, концентрация пикселей увеличивается. Пороговые значения выбираются с точки зрения наибольшей эффективности. В

результате получается четкий единственный контур, заполненный белым цветом – бинарная альфа карта объекта.

Затем, начинается обход пикселей по горизонтали. Как только встречается белый пиксель, программа запоминает место положения точки (пикселя) входа в контур. После того, как часть стоки с белыми пикселями, принадлежащими объекту, пройдены, и начинаются черные пиксели, программа запоминает точку (пиксель) выхода из контура. В результате, данного обхода производится сегментирование контура объекта. Выделенный контур заносится в буфер памяти, где по выделенным точкам входа и выхода из контура выделяется динамический объект во втором кадре.

Выделенный объект представляет собой цветное видео картинка. Помимо самого объекта выделяются и некоторые участки фона. Эти участки фона также нуждаются в передаче, так как, во время своего передвижения объект закроет новую область фона, но освободит участки фона, которые он до этого собою закрывал, как это показано на рисункунок.2.3.



Рисункунок.3.3. Соотношение движущегося объекта и статичного фона видео картинка.

Именно эти пустующие зоны и будут заполнены участками фона, которые передаются вместе с контуром динамического объекта одним файлом и запоминаются в буфере кадровой памяти.

Следующим шагом обработки двух кадров является поступление в формирователь выходного потока ключевого кадра, а также сегментированного динамического объекта с локальными координатами его размещения в кадре. Таким образом, для видео картинки с неизменным фоном и перемещающимися объектами переменной формы и цвета в предельном случае (1 объект) достаточно полностью передавать ключевой кадр и выделенный динамический объект смежного кадра с координатами его локального размещения в кадре. Все зоны второго кадра, которые не переданы, остаются стационарными и могут быть скопированы из первого кадра. А зоны, подлежащие изменению, передаются одним файлом. Таким образом, кадровый поток, обработанный кодером, становится готов к отправке по каналу связи.

После этого, декодер на приемной стороне восстанавливает с помощью ключевого кадра стационарный фон и вставляет сегментированный объект, находящийся в его буфере памяти в восстановленный кадр, ориентируясь по переданным координатам его перемещения. Это позволяет получить довольно большие коэффициенты сокращения с использованием относительно простых алгоритмов сегментации динамического объекта.

В данной выпускной квалификационной работе производится оценка эффективности применения объектно-ориентированного сокращения для реальных динамических объектов со сложной формой и текстурой. При этом, в рамках данного исследования рассматривается случай выделения одного единственного движущегося объекта на статичном фоне. Однако, в дальнейшем возможны разработки по

выделению сразу нескольких динамических объектов с последующей их обработкой по отдельности.

3.2. Алгоритм изменения формы цветного видео картинка в яркостное

Преобразование цветного видео картинка в яркостное необходимо для более легкого выделения объекта. Это связано с тем, что работать с черно-белом видео картинким в случае применения межкадровой разницы значительно легче. При произведении вычитании цветных видео картинки можно получить межкадровую разницу различных цветов, так как, в этом случае, от значений пикселей одного цвета будут отниматься значения пикселей другого цвета. Поэтому, целесообразно перед произведением межкадровой разницы осуществлять перевод цветных видео картинки в яркостные.

Каждый пиксель цветного видео картинка выражен тремя цветными компонентами: красной (Red), зеленой (Green) и синей (Blue) составляющей.

Различные сочетания этих трех компонент представляют все цвета видео картинка.

В цветовом пространстве YUV: Y — яркостная составляющая. Вместе с сигналом яркости Y введены ещё так называемые сигналы цветности U и V: $U = B(\text{Blue}) - Y$, $V = R(\text{Red}) - Y$. Таким образом, U, V — компоненты, отвечающие за цвет (хроматический красный и хроматический синий). В цветовой модели YUV эти величины рассматриваются как три составляющие цветового оттенка.

Перевод видео картинка из цветового пространства RGB в цветовое пространство YUV можно представить с помощью формул (3.1), (3.2), (3.3) для каждой составляющей [13]:

$$Y_{ij} = 0,258 * R_{ij} + 0,504 * G_{ij} + 0,098 * B_{ij}; \quad (3.1)$$

$$U_{ij} = -0,148 * R_{ij} - 0,29 * G_{ij} + 0,437 * B_{ij}; \quad (3.2)$$

$$V_{ij} = 0,437 * R_{ij} - 0,367 * G_{ij} - 0,07 * B_{ij}; \quad (3.3)$$

Блок-схема данного алгоритма представлена на рисунке 3.4, где:

i, j – координаты пикселей по горизонтали и вертикали соответственно;

h – высота обрабатываемого видео картинка;

w – ширина обрабатываемого видео картинка;

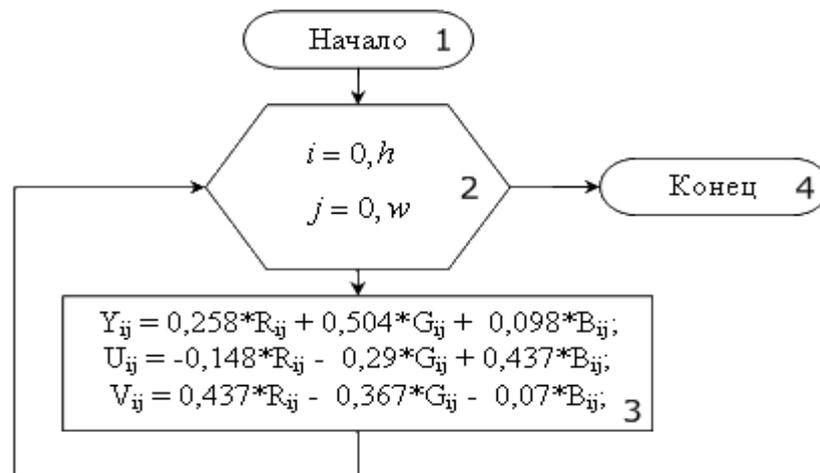


Рисунок.3.4. Алгоритм изменения формы цветного видео картинка RGB в яркостное YUV.

Каждый загруженный кадр подвергается обработке данного алгоритма. Начинается обход пикселей сначала по вертикали, а потом и по горизонтали (Блок 2). Таким образом, каждый пиксель сканируется и подвергается преобразованию по формулам 3.1, 3.2, 3.3 (Блок 3). В результате, мы получаем яркостное видео картинки из цветного, как это показано на рисунке 3.12 для первого (опорного) кадра и на рисунке 3.13. для второго (смежного кадра).

3.3. Алгоритм межкадровой разницы

Для определения контура и осуществления сегментации динамического объекта со сложным, изменяющимся в процессе движения, контуром, было решено

использовать метод межкадровой разницы. Это связано с тем, что вычислить контур динамического объекта, меняющего свою форму во время своего передвижения, крайне сложно. Однако, метод межкадровой разницы позволяет осуществить данную трудоемкую задачу. Одним из основных преимуществ данного метода является достаточно простая программная реализация и достижение хороших коэффициентов сокращения ($K_{сж}=34$ раза со сокращённым объемам без потерь каждого кадра, как смежного).

Также, одним из условий применения метода межкадровой разницы к реальному видео, является первичная обработка этого видео посредством параметрических моделей (обработка параметров движения снимающей видео камеры) для получения относительно статичного фона. То есть, осуществляется “привязка” локальных координат фона в кадре к каким либо постоянным значениям. Без использования параметрических моделей метод межкадровой разницы станет нецелесообразным в применении.

Реализации метода межкадровой разницы осуществляется с помощью формул 3.4, 3.5, 3.6:

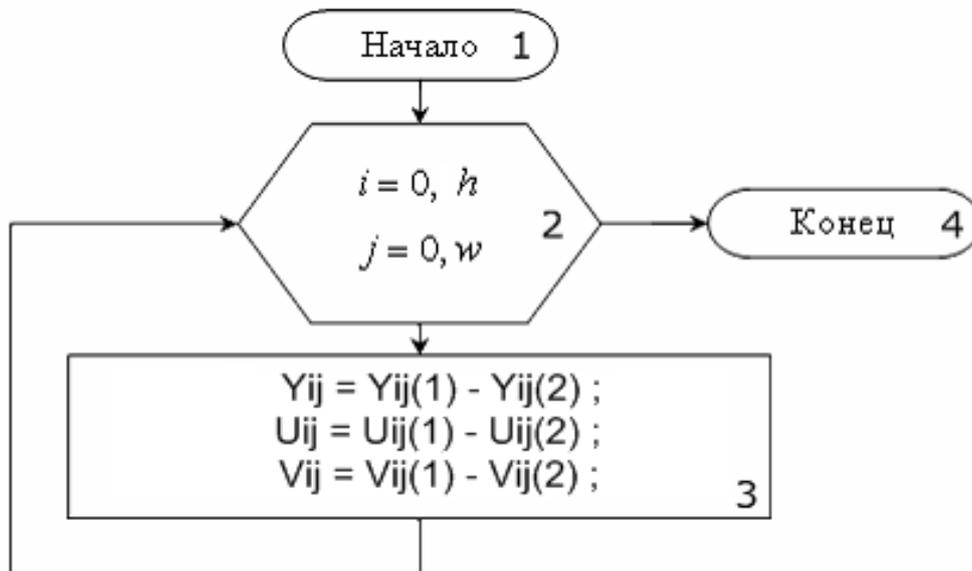
$$Y_{ij} = Y_{ij} (1) - Y_{ij} (2); \quad (3.4)$$

$$U_{ij} = U_{ij} (1) - U_{ij} (2); \quad (3.5)$$

$$V_{ij} = V_{ij} (1) - V_{ij} (2); \quad (3.6)$$

В результате этого, из каждого значения яркостной составляющей YUV спектра пикселя одного кадра вычиталось значение яркостной составляющих YUV спектра пикселя из другого кадра с точно такими же локальными координатами размещения в кадре. Данный алгоритм продемонстрирован в виде структурной схемы на рисунке 3.5.

На структурной схеме показано, что алгоритм начинает свою работу с обхода пикселей видео картинка по вертикали и горизонтали обоих загруженных кадров. Начинается обход пикселей видео картинка по вертикали (от 0 до h) и обход пикселей по горизонтали (от 0 до w), как это показано во втором блоке структурной схемы на рисунке 3.5. Каждый пиксель выражен значениями трех YUV составляющих. От каждой из YUV составляющих пикселя первого кадра отнимается YUV составляющая пикселя второго кадра по формулам 3.4, 3.5, 3.6.



Рисунокунок. 3.5. Алгоритм межкадровой разницы.

Затем, процесс повторяется пока обход всех пикселей видео картинка по вертикали и горизонтали не будет завершен. В итоге, на результирующем окне программы появится полученная межкадровая разница, отображающая только движущиеся области кадра.

3.4. Алгоритм пороговой обработки яркостного сигнала

Результатом работы алгоритма межкадровой разницы является полутонового видео картинка движущихся областей (объект и частичная область фона, соотносящаяся с передвигающимся объектом), очень близкое по форме к контуру движущегося динамического объекта. Для получения двухградационного видео картинка, производится пороговая обработка сигнала, которая заключается в том, что пикселям, имеющим значения яркости выше пороговой величины (которую можно задавать в ручную), присунокунокваивается максимальное значение равное 255, а меньше «порога» – значение нуля. В результате этого, формируется бинарное видео картинка, имеющее только два значения яркости – 0 или 1. Как было сказано ранее, использование бинарного видео картинка

позволяет простым и быстродействующим способом произвести выделение контура объекта. При использовании полутоновых видео картинки сегментация контура объекта стала бы крайне затруднительной с точки зрения программной реализации. Структурная схема алгоритма пороговой обработки яркостного сигнала изображена на рисунке 3.6, где:

Y_n – пороговое значение яркости;

Y_{ij} – значение яркости каждого пикселя;

В процессе бинаризации пользователь может вручную выбрать пороговое значение яркости (Блок 2) для межкадровой разницы. Затем, производится обработка всех пикселей видео картинка по вертикали и горизонтали (Блок 3), в процессе которой яркостное значение каждого выбранного пикселя сравнивается с выбранным пороговым значением в программе (Блок 4). Пороговое значение яркости выбирается вручную с точки зрения наибольшей эффективности.

Если значение яркости пикселя будет меньше порогового значения, то оно приравнивается к минимальному значению (0, черный цвет), и в результате, данный пиксель считается частью статичного фона видео картинка, а не контура динамического объекта (Блок 5).

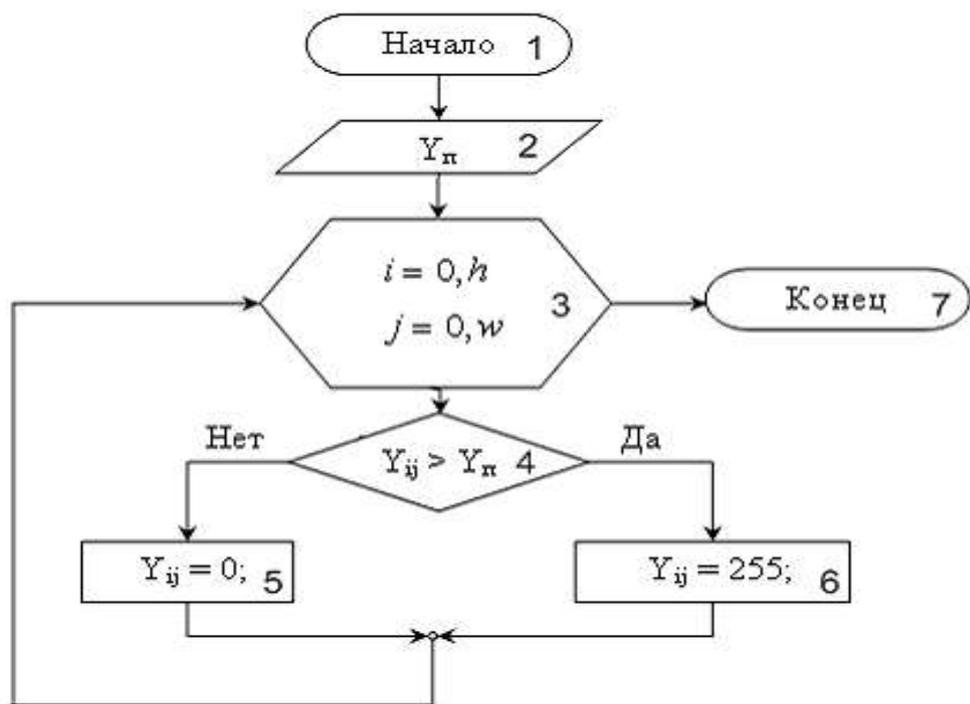


Рисунок.3.6. Структурная схема алгоритма пороговой обработки яркостного сигнала.

В противном случае, когда значение яркости пикселя больше порогового значения, то оно приравнивается к максимальному значению (255, белый цвет), и данный пиксель считается частью контура объекта (Блок 6). Цикл заканчивается, когда проверяется последний пиксель видео картинка и ему присуждается определенное значение.

Таким образом, получается бинарное видео картинка с двумя значениями яркости (0 и 1), представленное на рисунке. 3.16.

3.5. Алгоритм фильтра удаления случайных шумов

Результат пороговой обработки представляет собой бинарное видео картинка, очень близкое по форме к реальному контуру объекта. Однако, в кадре обычно присутствует случайный шум, в виде всплесков яркости, отображающихся в качестве случайных одиночных или малых групп пикселей. Случайный шум появляется из-за некоторых недостатков работы параметрических моделей. Метод параметрических

моделей как один из способов механизма компенсации движения позволяет стабилизировать движение снимающей видео камеры и осуществить привязку статичного фона к каким-либо локальным координатам в кадре.

Однако, не смотря на обработку данным методом, обычно присуще некоторому дрожанию фона, которое отображается при применении метода межкадровой разницы в качестве случайного шума.

Для удаления подобных одиночных пикселей используется фильтр удаления случайных шумов. Основным принцип работы данного фильтра заключается в следующем: выбирается группа из трех пикселей, находящихся в одной строке или одном столбце. Затем, проверяются первый и последний пиксель в группе. Если они оба имеют черный цвет, то средний пиксель относится к шуму и представляет собой случайный единичный пиксель. Это значит, что средний пиксель надо также приравнять к нулю (черный цвет), чтобы исключить случайный шум. Более подробно принцип работы данного фильтра отображен в виде структурной схемы на рисунке 3.7, где:

Y_{ij} - яркость пикселя i строки и j столбца бинарного видео картинка;

$Y_{i-1,j}$ - яркость пикселя $i-1$ строки и j столбца бинарного видео картинка;

$Y_{i+1,j}$ - яркость пикселя $i+1$ строки и j столбца бинарного видео картинка;

$Y_{i,j-1}$ - яркость пикселя i строки и $j-1$ столбца исходного видео картинка;

$Y_{i,j+1}$ - яркость пикселя i строки и $j+1$ столбца исходного видео картинка;

Y_n - пороговая яркость пикселей;

Работа данного фильтра осуществляется следующим образом. Сначала совершается обход пикселей по вертикали. Для обработки в массив вводятся значения яркости трех пикселей $Y_{i-1,j}$, $Y_{i,j}$, $Y_{i+1,j}$, находящихся в одном столбце j (Блок 2). Затем, проверяются значения

первого и последнего пикселя ($Y_{i-1,j}$ и $Y_{i+1,j}$). Если их значения меньше порогового значения $Y_{\text{п}}$ (Блок 3 и Блок 4), то значение яркости среднего пикселя $Y_{i,j}$ принимается равным нулю (Блок 5). Затем, весь процесс повторяется со следующими тремя пикселями в том же столбце.

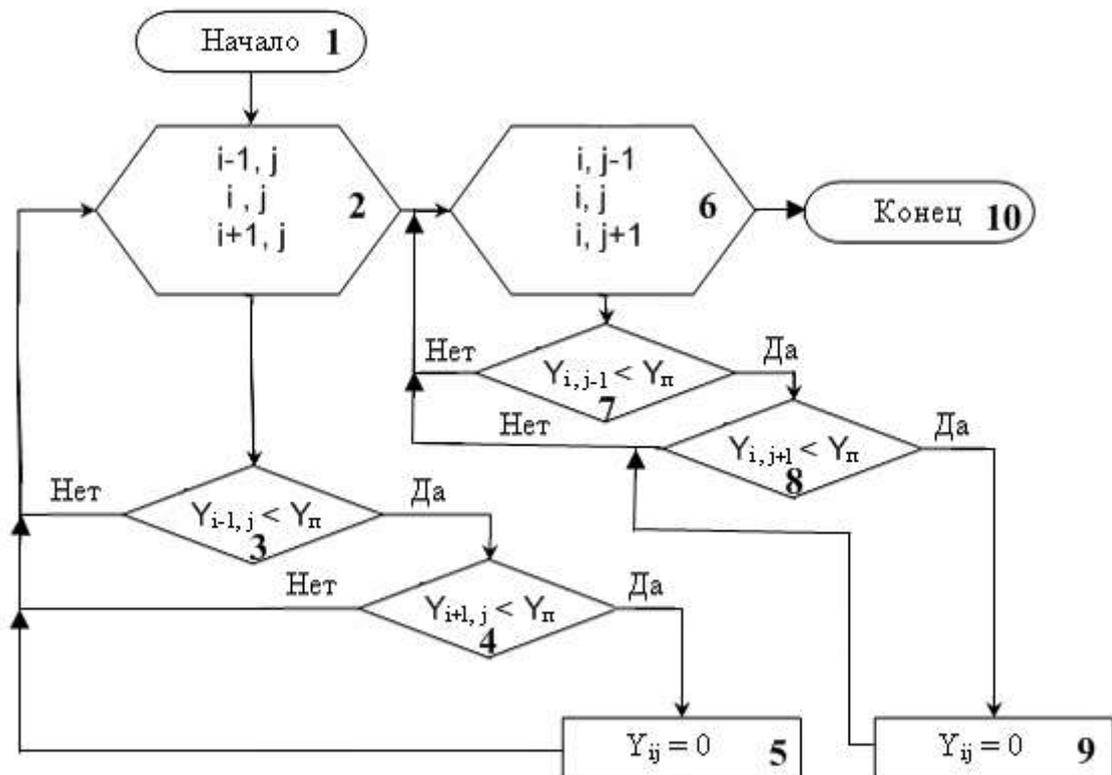


Рисунок 3.7. Структурная схема алгоритма фильтра удаления случайных шумов.

После завершения обхода пикселей по вертикали начинается обход пикселей по горизонтали (Блок 6) аналогичным образом. Также, выбирается группа из трех пикселей $Y_{i,j-1}$, $Y_{i,j}$, $Y_{i,j+1}$, и если значения яркости первого и последнего пикселя ($Y_{i,j-1}$ и $Y_{i,j+1}$) в выбранной группе меньше порогового значения $Y_{\text{п}}$ (Блок 7 и Блок 8), то значение яркости среднего пикселя $Y_{i,j}$ выбирается равным нулю (Блок 9). Затем, выбираются следующие три пикселя в той же строке, и весь цикл повторяется до тех пор, пока обход пикселей по горизонтали не будет завершен.

Пороговое значение может быть равным любому числу от 0 до 255. Его значение не влияет на работу алгоритма, так как в обработке участвует бинарное видео картинки, где присунокунокутствует лишь два значения яркости пикселей (0 – черный цвет и 255 – белый цвет). Пороговое значение необходимо для указания разницы между черными пикселями и белыми пикселями при выполнении условий в Блоке 3,4,7,8.

Наличие двух обходов (обход по строкам и по столбцам) вдвое повышает эффективность данного фильтра.

После завершения работы фильтра можно получить очищенное от случайного шума в виде случайных одиночных пикселей или их групп малой концентрации бинарное видео картинки, представленное на рисунокунок. 3.17.

3.6. Алгоритм фильтра формирования односвязанного контура

После удаления случайных шумов на видео картинки, возникает следующая трудность – наличие множества многосвязанных контуров внутри основного контура объекта. Это связано с тем, что обычно реальные динамические объекты содержат области с различной цветностью и яркостью.

При использовании межкадровой разницы подобные области также рассматриваются как движущиеся псевдообъекты внутри реального объекта.

В результате, образуются множество мелких контуров внутри основного контура динамического объекта, найденного посредством межкадровой разницы. Для удаления подобного дефекта псевдоконтуров используется фильтр формирования односвязанного контура. Помимо этого, процесс симуляции “нагрева” среды в виде бинарного видео картинка при использовании данного фильтра позволяет получить бинарную альфу

карту, которая в дальнейшем существенно облегчает сегментацию динамического объекта.

Работа данного фильтра состоит в том, что все бинарное видео картинки представляется как температурная карта среды, где белые пиксели элементизирует нагретые области, а черные пиксели – охлажденные области.

Затем, с использованием нестационарного уравнения теплопроводности среды производится симуляция теплового обмена в среде по формуле 3.7.

$$\frac{\partial T(x, y)}{\partial t} = D \frac{\partial^2 T(x, y)}{\partial x^2} + \phi(x, T(x, y), t)$$

(3.7)

где

D – коэффициент диффузии (скорость интерполяции яркостей пикселей);

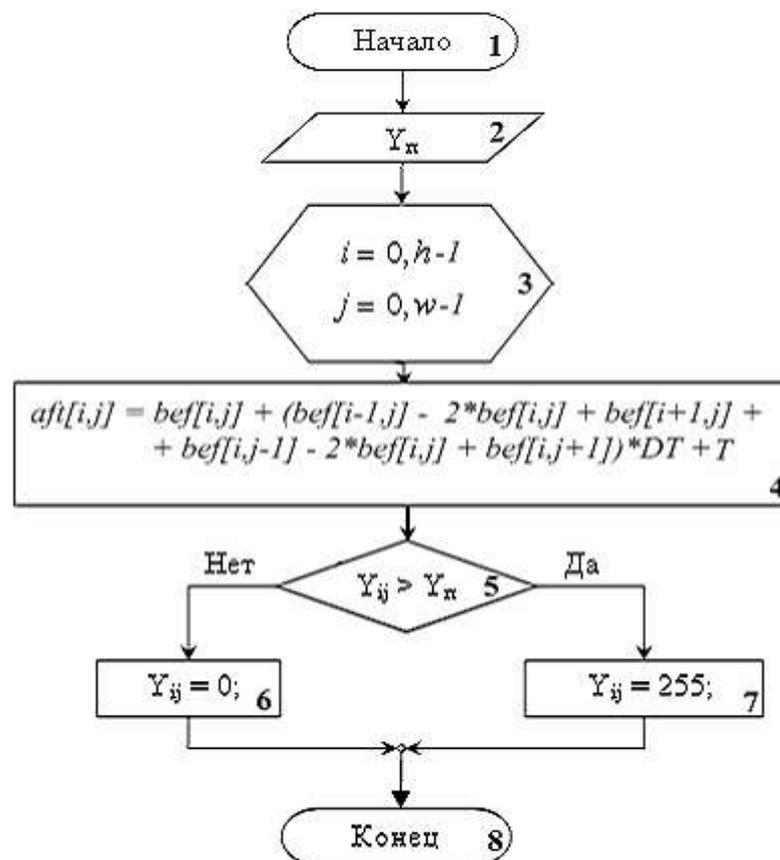
ϕ – источники тепла (все пиксели белого цвета);

$T(x,t)$ – температурный коэффициент (пороговое значение для процесса интерполяции между черными и белыми пикселями).

Сначала, реализуется процесс “охлаждения” среды, в результате которого, “нагретые” области (все белые пиксели) отдают тепло в среду. Данный процесс продолжается до определенного порогового значения, введенного пользователем. В данном случае с видео картинким происходит следующее: концентрации белых пикселей начинают уменьшаться в размерах. Малые концентрации быстро исчезают и все их пиксели становятся черными. Крупные группы пикселей основного контура значительно уменьшаются в размерах, но не успевают полностью исчезнуть до прекращения процесса. В результате все мелкие псевдоконтура полностью исчезают, и остаются лишь небольшие группы пикселей основного контура.

Следующим шагом обработки видео картинка данным фильтром является “нагрев” среды. В результате, концентрации белых пикселей

начинают с течением времени расширяться в размерах, и черные пиксели, соседствующие с белыми пикселями, тоже становятся белыми. Процесс продолжается до достижения некоторого порогового значения, введенного пользователем. Пороговое значение выбирается с точки наибольшей эффективности. Результатом становится расширение групп белых пикселей и заполнение ими всего внутреннего пространства контура. В итоге получается бинарная альфа карта объекта, где все пиксели внутри контура объекта являются белыми, а все пиксели вне его – черными. Принцип работы алгоритма данного фильтра изображен на рисункунок. 3.8.



Рисунокунок.3.8. Структурная схема алгоритма фильтра охлаждения и нагрева среды.

где

$aft[i,j]$ -пиксели i строки и j столбца первого слоя видео картинка;

$aft[i-1,j]$ -пиксели $i-1$ строки и j столбца первого слоя видео картинка;

$aft[i+1,j]$ -пиксели $i+1$ строки и j столбца первого слоя видео картинка;
 $bef[i,j]$ -пиксели i строки и j столбца второго слоя видео картинка;
 $bef[i,j-1]$ -пиксели i строки и $j-1$ столбца первого слоя видео картинка;
 $bef[i,j+1]$ -пиксели i строки и $j+1$ столбца первого слоя видео картинка;
 DT -шаг времени действия процесса

T – Пороговое значение для процесса интерполяции между черными и белыми пикселями.

Первым шагом в работе данного алгоритма является введение порогового значения Y_n (Блок 2). Ввод данного значения необходим для пороговой обработки результирующего видео картинка после окончания процесса “нагрева” среды. Это связано с тем, что процесс “нагрева” среды предполагает аппроксимацию соседних с белыми черных пикселей (обмен тепла с окружающей средой), в результате которой некоторые пиксели могут приобрести серый цвет. Для того чтобы получить полностью бинарное видео картинка используется пороговая яркостная обработка.

Затем, начинается обход всех пикселей по вертикали и горизонтали [Блок 3]. Берется полное число строк и столбцов ($i-1, j-1$). После этого, видео картинка условно подразделяется на два слоя [Блок 4]. Первый слой содержит исходное значение яркости для каждого пикселя – $bef[i,j]$. Второй слой содержит результирующее значение яркости пикселей слоя $bef[i,j]$ видео картинка после их обработки – $aft[i,j]$.

Формулы “охлаждения и нагрева” среды приводят к исчезновению многосвязанных контуров и вызывают расширение основного контура во внутрь до заполнения всего контура белыми пикселями. Данный процесс продолжается в пределах определенного времени, выраженного коэффициентом DT , значение которого лежит в пределах от 0 до 1, и представляет из себя условие Куранта – Леви для нестационарного уравнения теплопроводности. Также существует пороговый

температурный коэффициент T , который тоже вводится пользователем. При процессе нагрева среды данный пороговый коэффициент служит ограничением нагрева и соответственно увеличения концентрации пикселей. Значения коэффициентов: DT и T определяются пользователем исходя из наибольшей эффективности работы фильтра.

После окончания данного процесса необходима пороговая обработка, так как аппроксимационный процесс мог вызвать появление серых пикселей на месте черных. Для устранения данной проблемы и получения бинарного видео картинка применяется пороговая обработка получившегося видео картинка, в результате которой все пиксели сравниваются с определенным пороговым яркостным значением $Y_{п}$ [Блок 5]. При этом, все пиксели, яркость которых меньше порогового значения приравниваются к нулю [Блок 6], а те пиксели, яркость которых больше порогового значения приравниваются к 255 – белому цвету [Блок 7]. Пороговое значение $Y_{п}$ выбирается на 20% больше значения температурного коэффициента T для обеспечения эффективности работы алгоритма.

Использование таких терминов, как “охлаждение среды” и “нагрев среды”, объясняется тем, что данный алгоритм реализован с помощью уравнения из области термодинамики.

Результатом работы данного фильтра является исчезновение мелких внутренних контуров внутри объекта, а также формирование бинарной альфы карты, которая существенно облегчает процесс сегментации динамического объекта, как это показано на рисунках. 3.18.

3.7. Алгоритм обхода контура с фиксацией его координат

После окончания работы фильтра формирования односвязанного контура, а также последующей пороговой яркостной обработки видео картинка получается бинарная альфа карта объекта, с помощью которой достаточно легко выделить контур динамического объекта с последующей его фиксацией в памяти программы. Для этого необходимо начать обход всех пикселей и с помощью пороговой обработки отличить белые пиксели от черных в бинарном видео картинке. Там, где черные пиксели сменяются белыми, находится точка входа в контур, а место где белые пиксели в той же строке сменяются черными, отмечается сигнальной точкой выхода из контура. После этого, в указанном месте на диске компьютера создается специальный текстовый файл, куда заносятся координаты всех поочередно выделенных сигнальных точек входа и выхода из контура.

Более подробно схема работы алгоритма выделения и фиксации контура динамического объекта рассмотрена на рисунках 3.9, где:

$X_{\min}[J]$ – сигнальная точка (пиксель) входа в контур J столбца строки, по которой ведется обход ;

$X_{\max}[J]$ – сигнальная точка (пиксель) выхода из контура J столбца строки, по которой ведется обход ;

I – координаты входа в контур для записи в текстовый файл;

K – координаты выхода из контура для записи в текстовый файл;

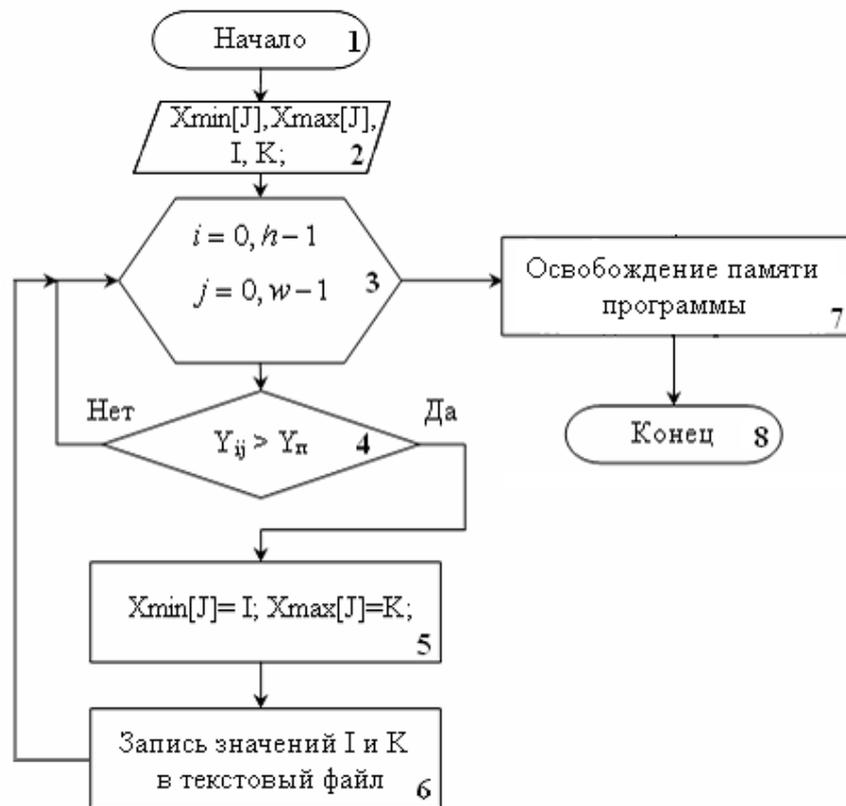


Рисунок. 3.9. Алгоритм обхода контура с фиксацией координат его пикселей.

Работа данного алгоритма начинается с обхода пикселей. Обход производится только по строкам, и когда строка обработана, происходит смещение по вертикали на следующую строку [Блок 3]. Сперва обход строки ведется слева на право. Как только встречается белый пиксель, значение которого больше порогового значения $Y_{п}$ (Блок 4), то данный пиксель отмечается как сигнальный пиксель входа в контур. Это производится следующим образом: значение X_{\min} приравнивается к целой переменной I . Затем, начинается обход той же строки справа налево для нахождения таким же образом сигнальной точки выхода из контура (точка окончания контура на данной строке).

Как только черные пиксели сменяются белыми, значение яркости первого белого пикселя при данном обходе превышает пороговое и он отмечается как сигнальная точка (пиксель) выхода из контура. Теперь уже значение X_{\max} приравнивается к целой переменной K [Блок 5].

Следующим шагом является создание специального текстового файла для записи контента. Найденные значения I и K сигнальных пикселей входа и выхода из контура соответственно записываются в данный текстовый файл.

После этого производится смещение по вертикали на один пиксель и начинается обход следующей строки таким же образом. Весь процесс повторяется заново до тех пор, пока не закончится обход всех строк видео картинка.

Таким образом, совершается обход всех строк с целью нахождения всех сигнальных точек входа и выхода из контура. Если на какой либо строке присунокунокутствует лишь один пиксель, принадлежащий объекту, то точка входа в пиксель будет иметь такие же координаты, как и точка выхода из контура ($X_{\min} = X_{\max}$).

Все сигнальные точки являются частью объекта и так же воспроизводятся в восстановленном видео картинки, как и остальные пиксели, принадлежащие динамическому объекту.

После того, как координаты всех сигнальных пикселей записываются в текстовый файл, производится очистка памяти программы [Блок 7]. На данном этапе фиксация (сегментация) контура динамического объекта завершается.

3.8. Алгоритм сегментации динамического объекта

После окончания процесса обхода контура в памяти сохраняется текстовый файл с координатами всех пикселей (сигнальных точек входа/выхода из контура), принадлежащих контуру динамического объекта.

Следующим шагом обработки видео картинка является сегментация уже не контура, а самого объекта с его текстурой и участком фона, который перекрывался объектом в предыдущем кадре. Данный процесс

происходит в несколько этапов. Прежде всего, загружается смежный кадр. Затем, программа обращается к сохраненному текстовому файлу, и начинает обход всех строк смежного кадра. Как только встречается пиксель с координатами аналогичными координатам пикселей контура, сохраненных в текстовом файле, то данный пиксель отмечается сигнальной точкой и считается пикселем входа в контур. Все последующие пиксели, включая пиксель входа в контур, сохраняются в память. Данный процесс происходит до тех пор, пока не встречается пиксель с координатами, аналогичными координатам сигнального пикселя выхода из контура на этой строке в текстовом файле, к которому производится обращение. Найденный пиксель отмечается как сигнальный пиксель выхода из контура, и процесс сохранения пикселей объекта на данной строке завершается.

После окончания обхода строки происходит смещение на следующую строку, и процесс повторяется заново. В ходе обработки пикселей происходит постоянно сравнение координат каждого пикселя с координатами выделенных сигнальных пикселей в текстовом файле.

Работа данного алгоритма более подробно проиллюстрирована на рисункунок. 3.10.

Согласно данному алгоритму процесс сегментации динамического объекта начинается с ввода значений сигнальных пикселей входа и выхода из контура [Блок 2]. После этого, загружается смежный кадр, который необходимо сжать посредством метода межкадровой разницы [Блок 3].

Затем, начинается обход пикселей всех строк смежного кадра [Блок 4], в ходе которого координаты каждого пикселя одной строки сравниваются с координатами сигнальных пикселей входа/выхода из контура той же строки в текстовом документе. То есть, в результате обхода происходит обращение к текстовому файлу [Блок 5] с целью

нахождения области динамического объекта в смежном кадре. Начинается последовательная проверка каждого пикселя, и если, координаты какого либо пикселя оказываются равными координатам сигнального пикселя входа в контур той же строки [Блок 6], то данный пиксель сохраняется [Блок 8]. Все последующие пиксели данной строки смежного кадра также сохраняются в память.

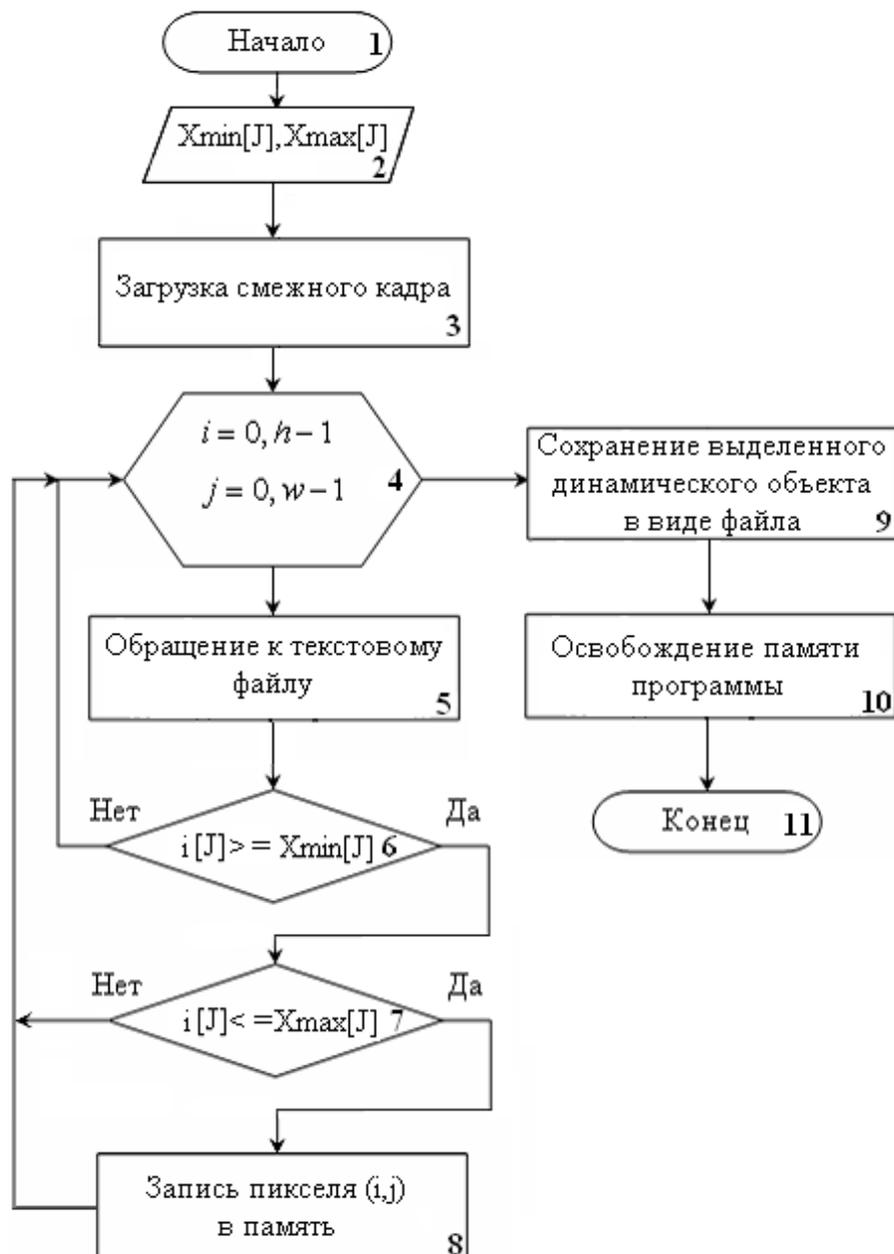


Рисунок. 3.10. Алгоритм сегментации объекта.

Наряду с этим, координаты каждого сохраняемого в память пикселя смежного кадра сравниваются с координатами сигнального пикселя выхода из контура той же строки в текстовом файле [Блок 7]. Как только встречается пиксель с координатами, равными координатам сигнального пикселя выхода из контура, то занесение пикселей данной строки смежного кадра в память заканчивается, и все последующие пиксели той же строки не сохраняются в память.

Когда обход строки заканчивается, происходит смещение на следующую строку и весь процесс повторяется заново. Результатом данного процесса является выделенный динамический объект, который сохраняется как файл [Блок 9]. Это позволяет очистить память программы с целью подготовки загрузки следующих кадров [Блок 10].

В сохраненном файле хранится не только сам динамический объект, но часть фона, которая была прежде закрыта объектом в предыдущем кадре. Данная область фона будет вставлена в пустующие области во время начальный вид смежных кадров. Результат программного алгоритма сегментации динамического объекта является файл с расширением "IMG", представленный на рисунке 3.19.

3.9. Алгоритм начальный вид смежного кадра на основе опорного кадра

После создания файла, в котором хранятся все пиксели динамического объекта, работа декодера сводится к восстановлению смежного кадра посредством текстового файла, в котором указаны все сигнальные пиксели входа/выхода из контура ($X_{min}[J]$ и $X_{max}[J]$). Происходит это следующим образом: начинается обход строк слева направо. Все пиксели, не принадлежащие контуру динамического объекта (области между точками входа в контур и точками выхода из

контура), принимаются как часть опорного кадра (статическая часть ключевого кадра). Однако, как только встречаются пиксели с координатами, принадлежащими контуру динамического объекта внутри текстового файл, то данные пиксели считаются частью динамического объекта смежного кадра и восстанавливаются из сохраненного файла.

Более подробно данный алгоритм рассмотрен на рисункунок. 3.11, где:

$X_{\min}[J]$ – сигнальная точка (пиксель) входа в контур J столбца строки, по которой ведется обход ;

$X_{\max}[J]$ – сигнальная точка (пиксель) выхода из контура J столбца строки, по которой ведется обход ;

Начинается работа алгоритма с ввода значений $X_{\min}[J]$ и $X_{\max}[J]$ (Блок 2). После этого начинается обход всех строк видео картинка. Когда обход строки заканчивается, происходит смещение на один пиксель по вертикали и процесс обхода строки повторяется заново.

При обходе той или иной строки происходит обращение программы к текстовому файлу и выделение значений $X_{\min}[J]$ и $X_{\max}[J]$ для данной строки. Каждый пиксель сверяется со значениями координат точек входа и выхода из контура в данной строке. Значение каждого обрабатываемого пикселя приравнивается к значению всех пикселей ключевого кадра с такими же координатами. То есть, данные пиксели считаются частью статичного фона ключевого кадра.

Однако, как только значение координат пикселя превышает значение координат сигнальной точки входа в контур (Блок 4), то программа обращается уже к смежному кадру, откуда и берется пиксель с точно такими же координатами. Все последующие пиксели той же строки так же берутся из смежного кадра. То есть, все данные пиксели считаются принадлежащими области динамического объекта в смежном кадре (Блок 9).

Этот процесс продолжается до тех пор, пока значение координат обрабатываемого пикселя в данной строке не превысит значение сигнальной точки выхода из контура $X_{\max}[J]$ (Блок 7). Как только это происходит, вставка пикселей из смежного кадра заканчивается и продолжается вставка пикселей соответствующих локальных координат из опорного кадра. Вставка пикселей продолжается до самого конца строки. Затем, происходит смещение по вертикали, и весь процесс повторяется для следующей строки видео картинки.

В результате получается восстановленное видео картинки, статичная часть которого была восстановлена из опорного кадра, а динамическая часть была восстановлена из смежного кадра посредством выделенного контура внутри текстового файла.

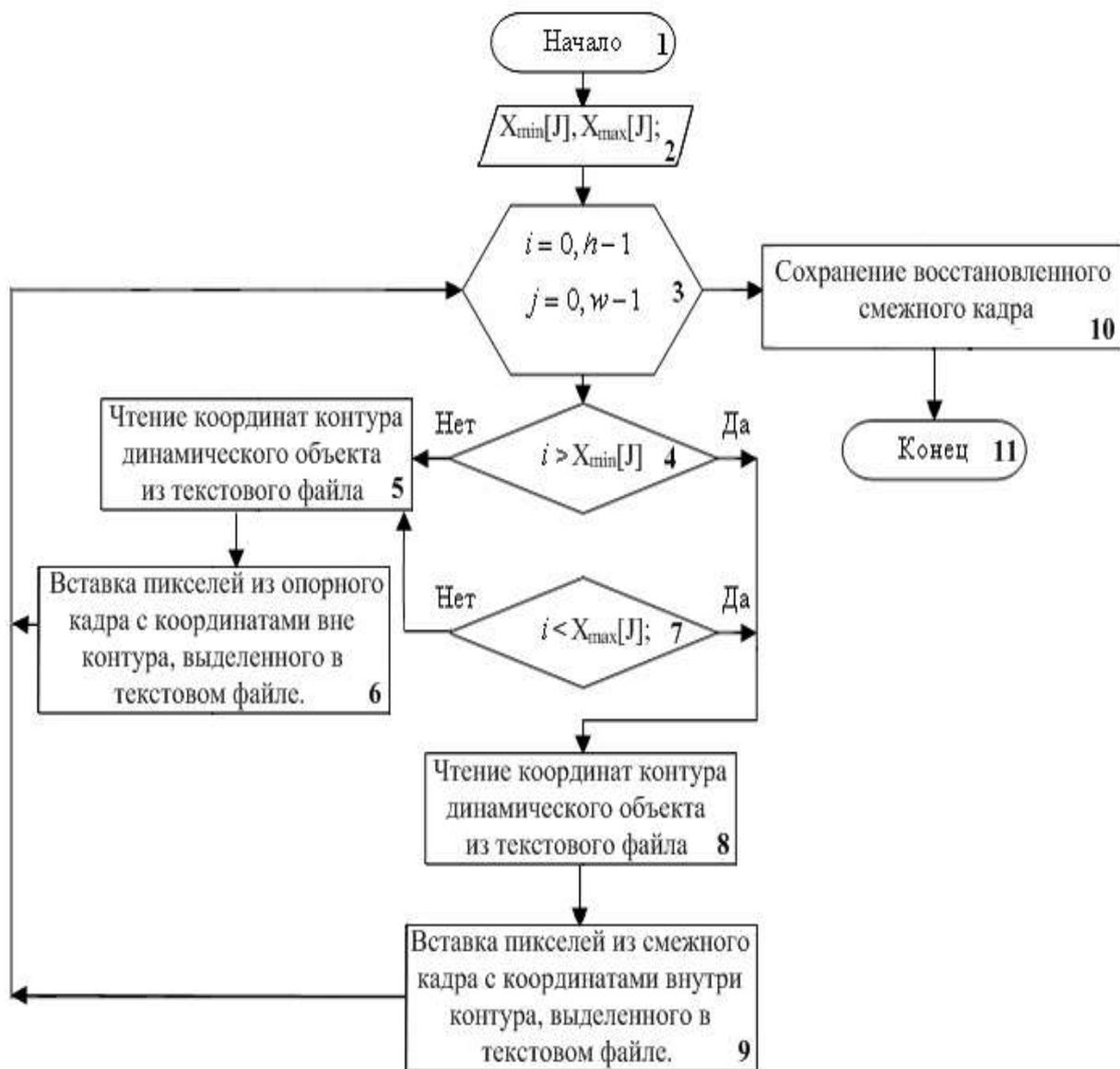


Рисунок. 3.11. Алгоритм начальный вид смежного кадра.

Результаты обработки видео картинки по выше описанным алгоритмам представлены на рисунки 3.12-3.19, а результат работы алгоритма начальный вид смежного кадра представлен на рисунке 3.20.



Рисунок. 3.12. Первый (опорный) кадр.



Рисунок. 3.13. Второй (смежный) кадр.



Рисунокунок. 3.14. Яркостное видео картинки опорного кадра.



Рисунокунок. 3.15. Яркостное видео картинки смежного кадра.



Рисунок. 3.16. Межкадровая разница.

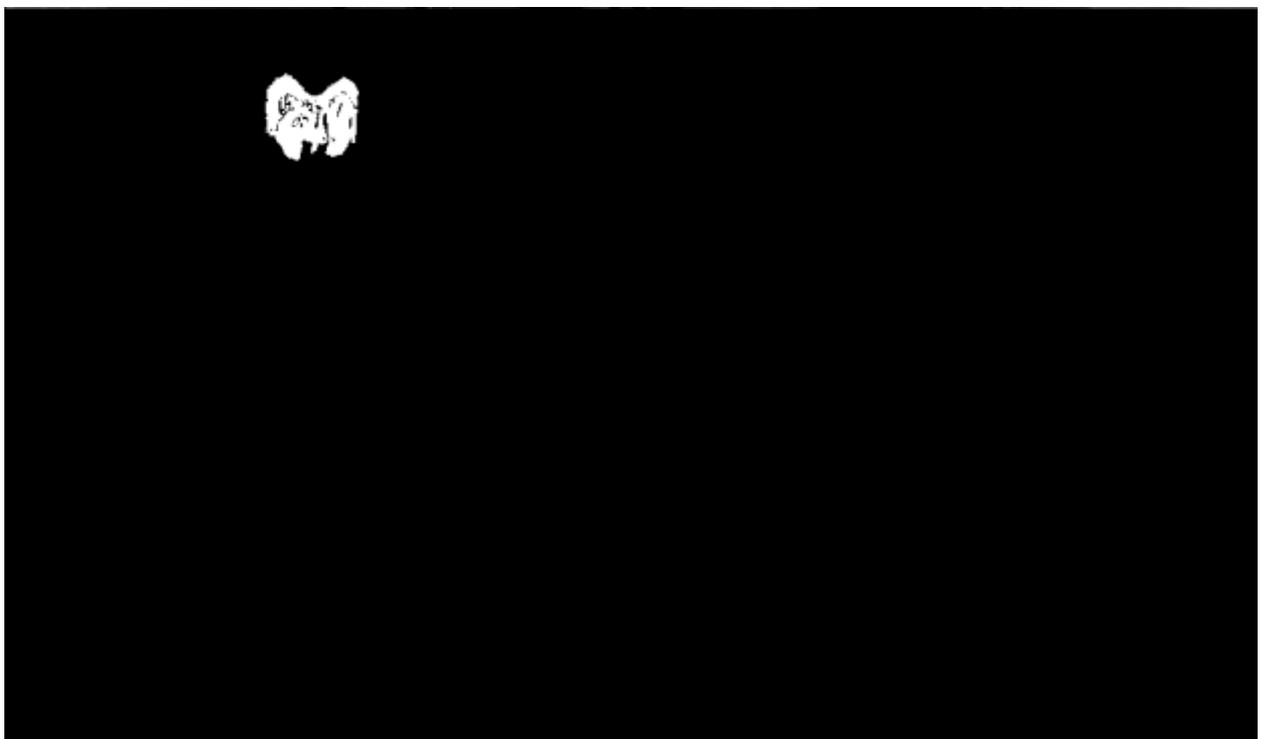
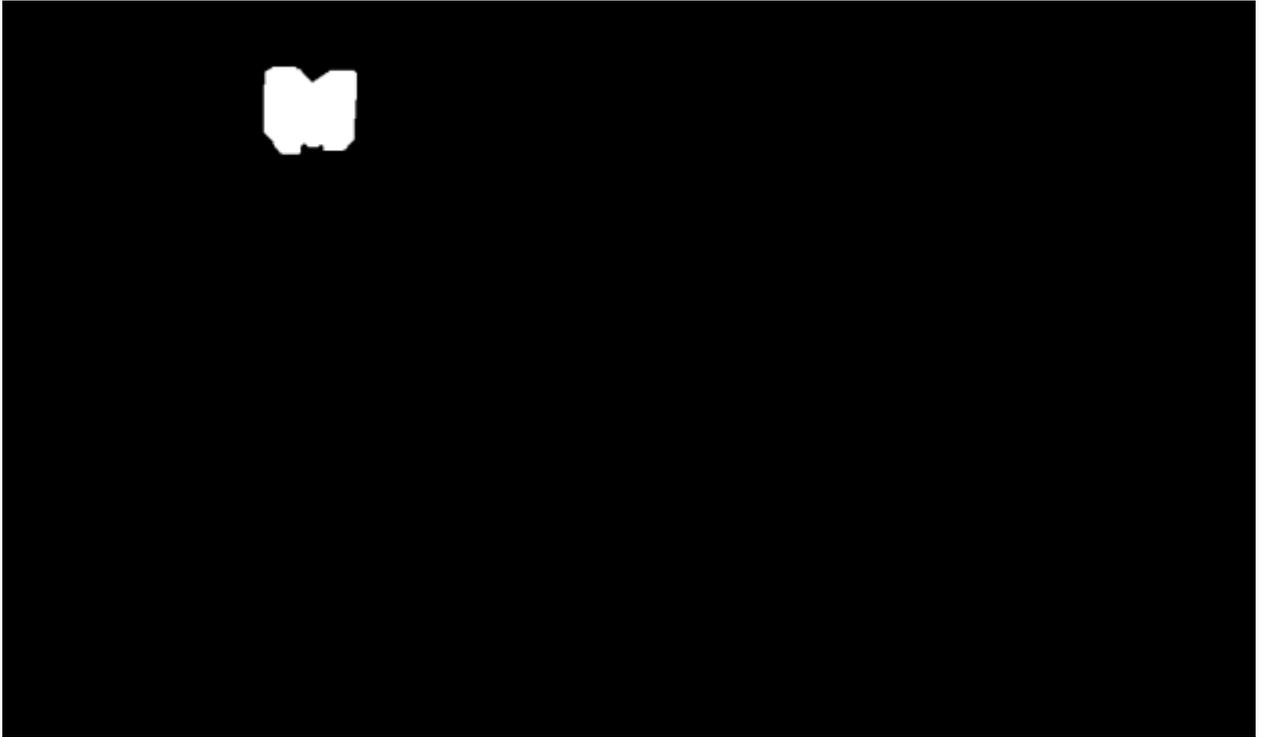


Рисунок. 3.17. Видео картинки после фильтрации случайных шумов.



Рисунокунок. 3.18. Видео картинки после выделения односвязанного контура (бинарная альфа карта).



Рисунокунок. 3.19. Выделенный динамический объект.



Рисунок. 3.20. Восстановленный смежный кадр.

Для оценки эффективности сокращения ТВ видео картинки кодеками были проведены экспериментальные исследования по сжатию тестовых видео картинки из 18 кадров формата 1050x623 и размером 1.87 Мбайт различными видеокодеками, работающими в режиме сокращения без потерь качества. Результаты сравнения параметров сокращения кадров, обработанных различными видео-кодеками представлены в таблице 3.1.

Таблица 3.1.

Результаты сравнения эффективности сокращения тестового видеопотока
из 18 кадров формата 1050x623 размером 1.87 Мбайт

Название используемого видео-кодека	Размер файла, получившегося в результате сокращения 18 кадров (ВМР 24 бит)	Коэффициент сокращения
Без сокращения	33, 660 Мб	Ксж=0;
MPEG-2	5,320 Мб	Ксж=6;
MPEG-4	3,130 Мб	Ксж=11;
MJPEG	11 Мб	Ксж=3;

Как видно из результатов экспериментальных контента из таблицы 3.1. видео-кодек MPEG-4 показывает наилучшие результаты, обеспечивая сокращённого объема в 11 раз.