

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

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

Физический факультет
Кафедра автоматизации физико-технических исследований
Направление подготовки: 03.03.02 «Физика»
Образовательная программа: бакалавриат

Выпускная квалификационная работа

Дмитриенко Михаил Юрьевич

***Исследование свойств и применимости алгоритма
сжатия видео JPEG XS для передачи с заданной пропускной
способностью и минимальной задержкой.***

«К защите допущена»
Заведующий кафедрой
Лысаков Константин Фёдорович,
к. т. н.,
Институт автоматики и
электрометрии СО РАН

Научный руководитель:

Лысаков Константин Фёдорович,
к. т. н.,
Институт автоматики и
электрометрии СО РАН

15 июня 2020 г.

« » июня 2020 г.

Дата защиты: 15 июня 2020 г.

Оглавление

1. Введение	3
2. Литературный обзор	5
2.1. Системы реального времени	5
2.2. Обзор JPEG XS.....	7
2.3. Профилирование программ.	11
2.4. Получение цифрового изображения.....	14
2.5. Типы каналов связи	16
3. Материалы и методы.	18
3.1. JPEG XS Test Model.....	18
3.2. Получение данных для анализа.....	20
4. Результаты и обсуждения	22
3.3. Исследование компромисса между качеством видео и битрейтом	22
3.4. Исследование потери качества видео при множественном процессе кодирования и декодирования.	26
3.5. Статистический анализ всех наборов изображений	28
3.6. Профилирование Jpeg XS test model (XSM)	29
3.7. Jpeg XS как система реального времени	31
5. Выводы.....	34
6. Список литературы	35

1. Введение

Существует немалое количество областей, где чрезвычайно важным параметром является временная задержка получения видео. К таким областям относятся управления БПЛА и видеомонтаж съёмок в реальном времени. При ограничениях на пропускную способность каналов передачи возникает необходимость использования видеокодеков для уменьшения размера передаваемой информации. Особенностью такого применения становится задержка в отображении видео на приёмнике минимум в 1 кадр плюс время на передачу данных и сжатие-разжатие.

Также актуальной проблемой при передаче сжатого видео по каналам передачи данных является подстройка в реальном времени под параметры пропускной способности канала передачи, которые могут динамически меняться во времени в зависимости от внешних условий. Это позволит обеспечить лучшее возможное качество видео в любой момент времени в зависимости от условий пропускной способности.

В задачах автоматического распознавания крайне актуальны системы передачи видео без межкадровой зависимости, что гарантирует непрерывность и корректность видеопотока даже при потере некоторых кадров. JPEG XS позволяет гибко регулировать задержку передачи, что делает возможным его применение в системах реального времени. Регулирование задержки происходит за счёт подбора таких параметров как количество бит на пиксель, ширина кадра, время получения кадра с камеры.

Сегодня существуют различные кодеки для сжатия видео, однако только JPEG XS позволяет решать задачу битрейта с точностью до байта, т.е. позволяет полностью использовать возможности сети. Также JPEG XS позволяет добиться задержки меньше кадра средствами, описанными в стандарте этого кодека, что облегчает его использование для поставленных задач.

Целью работы является исследование свойств и применимости алгоритма сжатия видео JPEG XS для передачи с заданной скоростью и минимальной задержкой. Для достижения поставленной цели необходимо решить следующие задачи:

1. Исследовать области применимости кодека JPEG XS при различных пропускных способностях сети и требованиях к качеству восстановления.
2. Реализовать сжатие и передачу с задержкой меньше кадра.
3. Выявить узкие места в готовой реализации кодека.
4. Оценить применимость JPEG XS в системах реального времени.

2. Литературный обзор

2.1. Системы реального времени

Необходимость минимизации задержки передачи видео в задаче видеоконтроля имеет целесообразность в случае её использования в режиме реального времени. Специфика задачи видеоконтроля определяет требование к задержке передачи видео. Например, в задаче управления беспилотным летательным аппаратом (БПЛА) существенное значение имеет величина задержки видео, из-за того, что необходимо вовремя реагировать на препятствия. Поэтому в таких системах чем меньше задержка, тем лучше. Тогда как, например, в случае передачи изображения с камер видеонаблюдения на пост охраны редко возникает потребность в задержке менее кадра. Таким образом, специфика задачи определяет параметры системы реального времени.

Система реального времени (СРВ) – это любая система, в которой момент времени вывода информации имеет важное значение. Обычно такая значимость определяется тем, что источником входной информации являются физические процессы, протекающие во внешней среде, и реакция системы должна соотноситься с теми же процессами. Временная задержка между моментами ввода и вывода должна быть ограниченной достаточно мала чтобы обеспечить своевременность реакции системы на внешнее воздействие. Требования к времени реакции обычно разделяют на три категории:

Жесткое реальное время – увеличение задержки превышающее некоторое значение приводит к аварийным (возможно катастрофическим) последствиям и является недопустимым.

Мягкое реальное время – увеличение задержки превышающее некоторое значение допускается (если такое событие происходит *«не слишком часто»* и увеличение задержки происходит на *«небольшой промежуток времени»*).

Твердое реальное время – определены два значения задержки, первое из которых (меньшее) рассматривается как ограничение для мягкого реального времени, а второе (большее) как ограничение для жёсткого реального времени.

Реальное время не означает «быстро», но «своевременно»; каждая задача должна закончиться в пределах заданного для нее времени; в этом смысле говорят о «предсказуемости» поведения системы.

Характеристики задачи реального времени:

- длительность выполнения,
- период активизации,
- допустимое время завершения,
- время выполнения в наихудшем случае,
- приоритет,
- время реакции,
- задержка выполнения процедуры обработки события.

Каждая задача реального времени определяется постоянным набором процедур, повторяющихся в течении времени функционирования системы (от единиц до сотен и тысяч часов).

По строгости требований на выполнения задачи (системы) относят к классам «жесткого реального времени», «мягкого реального времени» и «твердого реального времени».

2.2. Обзор JPEG XS

JPEG XS - это новый международный стандарт от комитета JPEG [6-7]. Этот кодек позволяет сжимать изображения без визуальных потерь с контролем битрейта с точностью до байта. Одним из целевых вариантов использования кодека является передача видео по различным каналам связи.

На Рис. 1 показана общая блок-схема кодека JPEG XS. В случае ввода RGB цветовые компоненты декоррелированы посредством цветового преобразования без потерь, идентичного тому, которое используется в JPEG 2000. Затем применяется целочисленное необратимое вейвлет-преобразование. Чтобы соответствовать ограничениям по времени ожидания и избежать чрезмерных требований к памяти, предусмотрены только два вертикальных вейвлет-разложения. В горизонтальном направлении допускается до пяти последовательных разложений.

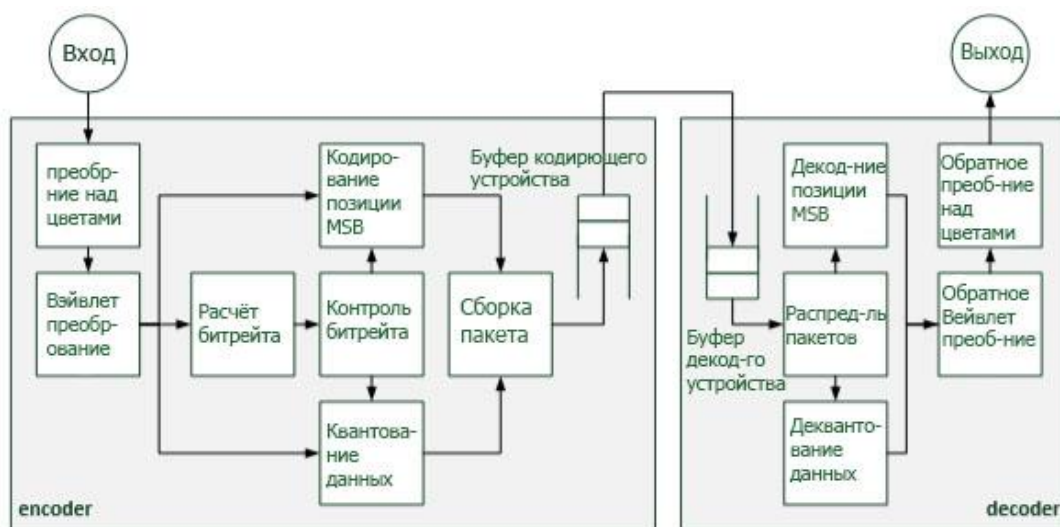


Рис. 1. Архитектура кодера и декодера JPEG XS.

Результирующие вейвлет-коэффициенты анализируются для вычисления бюджета, прогнозируется количество битов, необходимых для каждого возможного квантования. Поскольку большее квантование означает более сильное искажение сигнала, алгоритм управления скоростью вычисляет наименьший коэффициент квантования, который не превышает бюджет битов, доступный для кодирования вейвлет-коэффициентов. Затем вейвлет-

коэффициенты энтропийно кодируются, как описано далее. Наконец, все секции данных объединяются в структуру пакета и отправляются в канал передачи. Буфер сглаживания обеспечивает постоянную скорость передачи битов на выходе кодера, хотя входное изображение может состоять из областей ввода, которые легче сжимать, и других областей, требующих больше битов на пиксель. Учитывая, что декодер должен иметь возможность обрабатывать пиксели с постоянной тактовой частотой, количество считываемых битов за единицу времени зависит от того, хорошо ли сжимаются текущие вейвлет-коэффициенты или нет. Эти изменения скорости снова компенсируются сглаживающим буфером на входе декодера. Анализатор пакетов разделяет поток битов на отдельные порции данных, представляющие части поддиапазона, прежде чем коэффициенты вейвлета декодируются и преобразуются обратно в область пространственных пикселей.

Чтобы иметь возможность представлять изображение с как можно меньшим количеством битов, важно представлять часто встречающиеся значения пикселей короткими кодовыми словами, в то время как редкие значения пикселей могут быть представлены большими кодовыми словами. Этот процесс называется энтропийным кодированием. Кодирование и декодирование таких слов переменной длины требует значительных аппаратных и программных ресурсов. Таким образом, для обеспечения возможности реализации с низкой сложностью было решено выполнять кодирование с переменной длиной по группе из четырех коэффициентов. На Рис. 2 показана такая группа коэффициентов. Каждый коэффициент представлен знаковым битом и фиксированным числом битов величины. Затем можно выполнить энтропийное кодирование, исключив начальные нулевые битовые линии каждой группы коэффициентов (простые серые линии на Рис. 2). С этой целью кодер определяет для каждой группы коэффициентов так называемое положение MSB (Most Significant Bit).

-8	22	0	6
=	=	=	=
1	0	0	0
Биты, отвечающие за знак			
0	0	0	0
0	0	0	0
0	1	0	0
1	0	0	0
0	1	0	1
0	1	0	1
0	0	0	0
Позиция MSB			

Рис. 2. Группа коэффициентов для энтропийного кодирования.

Он соответствует самой значимой битовой линии в группе коэффициентов, где по крайней мере один бит коэффициента равен единице. Чтобы закодировать это значение положения MSB, его сначала вычитают из значения в соседней группе коэффициентов по горизонтали или вертикали, а затем эту разницу кодируют с помощью кода переменной длины. Этот механизм позволяет встраивать только битовые строки с меньшей значимостью, чем ведущие нулевые битовые линии, в поток битов, что приводит к сокращению данных. В случае, если доступного бюджета битов недостаточно для включения всех битовых линий, их количество может быть уменьшено путем квантования. [1]

JPEG XS обладает следующими ключевыми особенностями:

1. Зрительное отсутствие потерь между оригинальным и сжатым изображением (как определено в [2]). Типичные коэффициенты сжатия до 10:1 для изображений 4:4:4 и 4:2:2 с точностью до 12-бит на компоненту, но также может быть выше в зависимости от характера изображения или требования целевого приложения.
2. Надежность восстановления после множественного пересжатия, то есть без существенного ухудшения качества при 10 циклах декодирования-кодирования [3].

3. Низкая сложность, как в аппаратном, так и в программном обеспечении.
Процессор способен обрабатывать видеопоток с разрешением 4k, цветовой субдискретизацией - 4:4:4, 60 кадров в секунду в режиме реального времени. С аппаратной стороны, FPGA реализации не должны требовать внешней памяти и не должны занимать более 50% Artix7 XC7A200T или 25% Cyclon5 5CEA9 при применении к видео с разрешением 4k, цветовой субдискретизацией - 4:4:4, 60 кадров в секунду.[3]
4. Низкая задержка передачи данных за счёт разбития кадра на части.
Таким образом, JPEG XS является стандартизированным кодеком, способным получать задержку менее кадра и гибко регулировать бирейт.

2.3. Профилирование программ.

Профилирование — это сбор характеристик работы программы, таких как время выполнения отдельных фрагментов (обычно подпрограмм), число верно предсказанных условных переходов, число кэш-промахов и т. д. Инструмент, используемый для анализа работы, называют профилировщиком или профайлером (англ. profiler). Обычно выполняется совместно с оптимизацией программы.

Характеристики, собираемые профилировщиком, могут быть аппаратными (время) или вызванные программным обеспечением (функциональный запрос). Инструментальные средства анализа программы чрезвычайно важны для того, чтобы понять поведение программы. Проектировщики ПО нуждаются в таких инструментальных средствах, чтобы оценить, насколько хорошо выполнена работа. Программисты нуждаются в инструментальных средствах, чтобы проанализировать их программы и идентифицировать критические участки программы, т.е. участки повторяющиеся наибольшее количество раз.

В зависимости от степени детализации данных и способа сбора информации профилировщики подразделяются на статистические и профилировщики с инструментированием. Профилировщики с инструментированием добавляют инструкции к целевой программе для сбора необходимой информации, причем это может быть выполнено на уровне исходного кода, ассемблерного кода или на двоичном уровне [4]. Этот метод изменяет исходный код программы путем вставки профилирующего кода, после чего измененный код выполняется на целевой платформе. Однако инструментирование программы может вызвать изменения производительности и привести к некорректным результатам.

Статистические профилировщики работают посредством сэмплирования, т. е. профилировщик регулярно проверяет стек вызовов целевой программы, используя прерывания операционной системы. Полученные профили, как

правило, менее точны, чем полученные от профилировщиков с инструментированием, однако такой подход позволяет целевой программе работать почти на полной скорости. Итоговый результат профилирования программы данным инструментом содержит различные показатели, как например, время выполнения каждой подпрограммы, которые могут быть детализированы на уровне инструкций.

Также имеется возможность анализа производительности многопоточных программ. Профилировщик имеет возможность построения дерева вызовов, которое визуализирует поток исполнения приложения и помогает выявить критические функции и временные затраты в приложении. Посредством семплирования на основе временных характеристик и событий профилировщик позволяет находить критические участки программ, потребляющие значительную долю ресурсов процессора, а также участки кода, в которых происходят промахи кэширования, ошибки предсказания ветвления и другие проблемы производительности. JPEG XS Test Model (см. раздел 2.1) представляет из себя исходный код кодека, оформленный как Visual Studio проект. Средства профилирования и диагностики Visual Studio позволяют диагностировать использование памяти и ЦП, и выявлять другие проблемы на уровне приложения. С их помощью можно накапливать данные о производительности во время выполнения приложения. Имеется возможность получить анализ использования ресурсов ЦП и памяти, а также просматривать события, отображающие сведения, связанные с производительностью. [5]

Профилировщик так же имеет возможность построения дерева вызовов. Кроме того, инструмент предоставляет возможность просматривать сведения о производительности непосредственно при взаимодействии с кодом, в частности, можно получить такие сведения, как длительность события (измеряется с момента последней приостановки отладчика или с момента запуска приложения) и длительность выполнения блока кода или времени, необходимого для завершения работы одной функции. Наконец, Visual Studio

позволяет оценить использование памяти в приложении, например, можно узнать число и размер объектов в куче.

2.4. Получение цифрового изображения

Для получения цифрового изображения применяются фотоматрицы различных типов. По технологии изготовления фотоматрицы можно разделить на два типа:

- ПЗС (CCD) матрица (прибор с зарядовой связью, англ. CCD, charge-coupled device)
- КМОП (CMOS) матрица (комплементарная структура металл-оксид-полупроводник; англ. CMOS, complementary metal-oxide-semiconductor)

По функционированию главное отличие в том, что в ПЗС матрице считывание сигнала происходит строками с последующей обработкой, а в КМОП матрице существует произвольный доступ к каждому элементу, в следствии чего возможно сканирование части изображения для более быстрой и экономичной работы. ПЗС матрица требует для своей работы больше энергии. Ранние образцы КМОП матриц были менее чувствительны из-за большего количества полупроводниковых элементов в ячейке. Современные технологии позволили уменьшить размеры полупроводниковых элементов, что положительно сказалось на качестве КМОП матриц.

Для получения цветного изображения с матриц цифровых фотоаппаратов, видеокамер и сканеров применяются различные цветные фильтры. Самый простой и распространённый это фильтр Байера (шаблон Байера).

Фильтр Байера состоит из 25 % красных элементов, 25 % синих и 50 % зелёных элементов, расположенных перед фото матрицей как показано на Рис. 3.

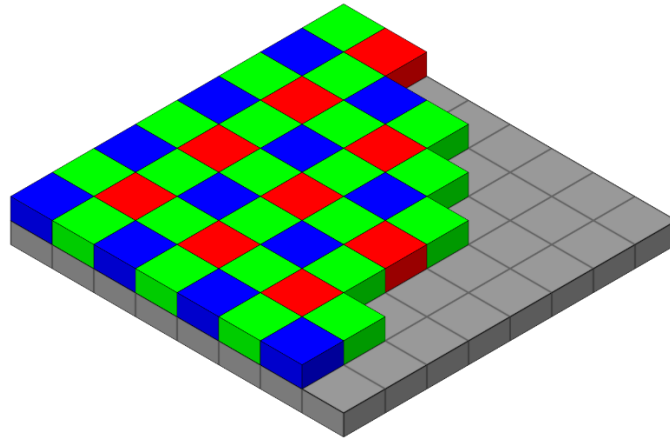


Рис. 3. Схематическое изображение фильтра Байера.

Формирование цифрового изображения разбито на 5 этапов (Рис. 4). На первом этапе мы имеем свет на матрице камеры, который не имеет такого понятия как пиксель. На втором этапе измеряется интенсивность света на каждом пикселе. На третьем этапе изображение находится в своем 'сыром' виде, т.е. каждый пиксель имеет свой цвет и свою интенсивность. На четвёртом этапе мы добиваемся того, чтобы каждый пиксель имел три компоненты (RGB), это достигается при помощи интерполяции. В итоге мы имеем конечное изображение для сжатия.

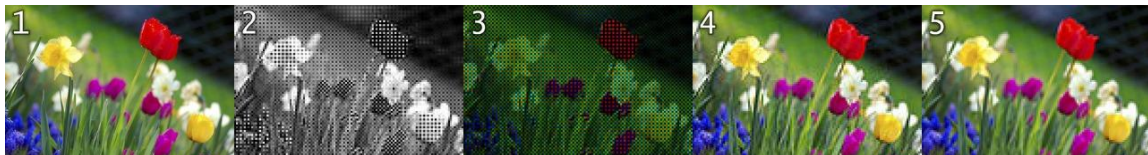


Рис. 4. Этапы формирования цифрового изображения.

2.5. Типы каналов связи

Под каналом связи можно понимать всё, что может передавать сигнал. Сигнал можно передавать при помощи проводов либо радиосигналов. В соответствии с этим можно поделить каналы связи на две группы: проводные и беспроводные. Все передаваемые сигналы делятся на три вида: цифровые, аналоговые и импульсные. В последнее время наибольшее распространение получили цифровые сигналы. К типичным беспроводным технологиям связи можно отнести Wi-Fi, Радио, Bluetooth. К типичным проводным технологиям связи можно отнести Ethernet (проводной или оптоволоконный) и просто провода. Каждый канал связи имеет следующие характеристики:

- максимальная задержка передачи сообщения в одну сторону,
- пропускная способность (постоянная или вариативная, и её типичные значения).

Цифровой канал связи передает сигналы, которые характеризуются использованием дискретных амплитуд. Двоичный цифровой сигнал, например, имеет лишь два разрешенных значения, представляющих двоичные цифры “ВКЛЮЧЕНО” и “ВЫКЛЮЧЕНО”.

Стандарты Ethernet определяют проводные соединения и электрические сигналы на физическом уровне. Формат кадров и протоколы управления доступом к среде определяются на канальном уровне модели OSI. Всё, передаваемое одним узлом, одновременно принимается всеми остальными.

Wi-Fi — технология беспроводной локальной сети с устройствами на основе стандартов IEEE 802.11. Обычно схема сети Wi-Fi содержит не менее одной точки доступа и не менее одного клиента. Также возможно подключение двух клиентов в режиме точка-точка (Ad-hoc), когда точка доступа не используется, а клиенты соединяются посредством сетевых адаптеров «напрямую». Точка доступа передаёт свой идентификатор сети (SSID) с помощью специальных сигнальных пакетов на скорости 0,1 Мбит/с

каждые 100 мс. Поэтому 0,1 Мбит/с — наименьшая скорость передачи данных для Wi-Fi.

На Таблица 1 представлены типичные битрейты для различных каналов передач.

Технология передачи	Битрейт
Мобильная связь	1,2 КБ/с - 40 750 КБ/с
Беспроводная сеть	0,25 МБ/с – 866 МБ/с (на каждого абонента с МІМО не более 0.125 или 0.25 ГБ/с)
Глобальная сеть	77,76 МБ/с - 4976 МБ/с
Локальная сеть	<1,25 МБ/с – 25 ГБ/с
Шина у компонентов компьютера	425 КБ/с - 51,20 ГБ/с
Обмен данными с внешними устройствами компьютера	28,8 КБ/с - 5 ГБ/с

Таблица 1. Типичные битрейты для различных технологий передачи информации.

3. Материалы и методы.

3.1. JPEG XS Test Model

Стандарт ISO IEC 21122-1 предоставляет исходный код реализации JPEG XS. Для того, чтобы пользоваться этой реализацией, необходимо знать, какие параметры принимают кодер и декодер. Пример использования кодировщика выглядит следующим образом:

```
jxs_encoder <command line options> <input file> <output file>,
```

где <input file> - это имя файла, оканчивающееся на “. ppm”, “. pgx” или “. yuv”, а <output file > - это имя файла заканчивающееся на “.jxs”.

jxs_encoder поддерживает следующие параметры:

-p <profile> – определяет профиль с которым будет сжато исходное изображение. Существуют следующие профили: 3 – main profile, 4 – high profile, 5 – light-subline profile, 6 – light profile. Подробное описание профилей приведено в [3].

-b <bitrate> – определяет количество бит на пиксель в закодированном изображении. Возможны дробные значения.

-s <bytesize> – определяет размер сжатого файла в байтах.

При этом совместное использование флагов –b и –s запрещено.

-o <mode> – определяет приоритет потерь информации при сжатии. Существуют следующие возможные режимы: Psnr – минимизирует среднеквадратичную ошибку, Visual – минимизирует визуальную потерю качества изображения.

-w <width> – определяет ширину изображения, используется только для .yuv файлов.

-h <height> – определяет высоту изображения, используется только для .yuv файлов.

-d <bitprecision> – определяет количество бит на пиксель, используется только для .yuv файлов.

Пример использования декодировщика выглядит следующим образом:

jxs_decoder <command line options> <input file> <output file>,

где <output file> - это имя файла, оканчивающееся на “.ppm”, “.pgx” или “.yuv”,
а <input file > - это имя файла заканчивающееся на “.jxs”.[6]

3.1. Получение данных для анализа

Для получения данных использовался персональный компьютер с центральным процессором Intel® Core™ i5-8250U. Для работы с JPEG XS было разработано ПО для исследовательской работы на языке программирования python (примеры кода см Рис. 5).

```
def draw(typ):
    for quality in res[typ]:
        fig, (ax1) = plt.subplots(nrows=1, ncols=1, figsize=(8, 16))
        for file_name in res[typ][quality]:
            x = [res[typ][quality][file_name][i][0] for i in range(len(res[typ][quality][file_name]))]
            y = [res[typ][quality][file_name][i][1] for i in range(len(res[typ][quality][file_name]))]
            ax1.plot(x,y)
            plt.xlabel('Бит в секунду' + " " + quality + " " + typ, axes=ax1)
            plt.ylabel('Пиковое отношение сигнал-шум', axes=ax1)
        plt.show()

def start(count, step, start_size, stop_size, step_size):
    global res
    while (True):
        global res
        print(count, step)
        if step == 0:
            jxs_encoder.encode()
            step += 1
            print(count, step)
        if step == 1:
            decode_ppm.decode()
            step += 1
            print(count, step)
        if step == 2:
            res = metrics(start_size, stop_size, step_size)
            step += 1
            print(count, step)
        if step == 3:
            for typ in res:
                draw(typ)
            print(count, step)
            break
```

Рис. 5. Основные функции модуля тестирования JPEG XS.

На Рис. 5 представлен исходный код 2 функций, которые покрывают все 4 этапа тестирования. На первом этапе тестирования исходные изображения кодируются для имитации различных битрейтов. Стартовое значения битрейта, конечное значение битрейта, а также шаг битрейта можно посмотреть на Рис. 6.

```

count = 3
step = 0
fps = 30
birate_start = 1*10e5
bitrate_stop = 1*10e8*2
birate_step = 10e4*5

# bitrate_interval_count = 100
start_size = int(birate_start/fps/8)
stop_size = int(bitrate_stop/fps/8)
step_size = int(birate_step/8)
jxs_encoder = Encoder(step_size, stop_size, step_size)
res = {}

```

Рис. 6. Раздел программы для исследований, отвечающий за определение констант.

На втором этапе тестирования закодированные кадры раскодируются, восстановленные изображения сохраняются под уникальной комбинацией имя-путь. На третьем этапе считается метрика качества восстановленных изображений PSNR (Пиковое отношение сигнал-шум) и результаты сохраняются в переменную `res` (Рис. 6). На четвёртом этапе тестирования происходит построение графиков, которые визуализируют результаты тестов. Анализ данных производится в разделе 3 Результаты и обсуждения. Все эксперименты проводились с помощью JPEG XS Test Model. В разделе 2.1 подробно описаны параметры тестовой реализации.

4. Результаты и обсуждения

3.2. Исследование компромисса между качеством видео и битрейтом

В Таблица 2 приведены результаты тестов JPEG XS на качество восстановления изображений при заданном битрейте. В ячейках таблицы указана ошибка восстановления изображения (пиковое отношение сигнала к шуму). Символ ‘+’ обозначает полное восстановление изображения.

Битрейт (МБит/с)	UHD	FullHD	HD	SD
50	35,57	40,12	46,83	46,25
100	39,2	47,5	50,73	54,17
200	42,3	52	61,71	+
300	43,61	55	85,85	+
400	44,26	58,5	+	+
500	45,08	65	+	+
1000	47,38	+	+	+
2000	53,93	+	+	+

Таблица 2. Зависимость качества восстановления от пропускной способности сети.

Приведенные ниже графики наглядно отображают компромисс между битрейтом и качеством восстановления изображения. Исследовались три типа изображений: “Голова”(7-10 А), “Небо” (7-10 Б), “Мелкие детали ” (7-10 В). Изображения типа “Голова” содержат лица людей, на изображениях типа “Небо” большую часть занимает небо, изображения типа “Мелкие детали” полностью состоят из мелких деталей.

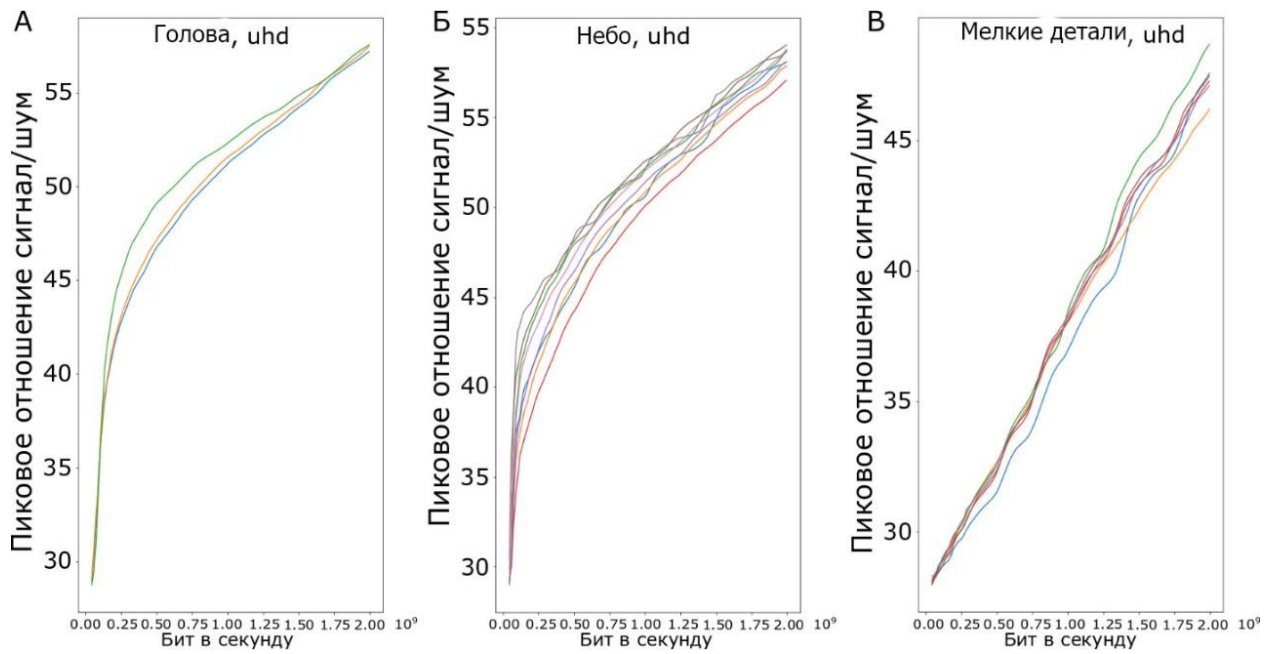


Рис. 7. Зависимость пикового отношения сигнала к шуму от пропускной способности сети (разрешение 4k, цветовая субдискретизация - 4:4:4, 30 кадров в секунду).

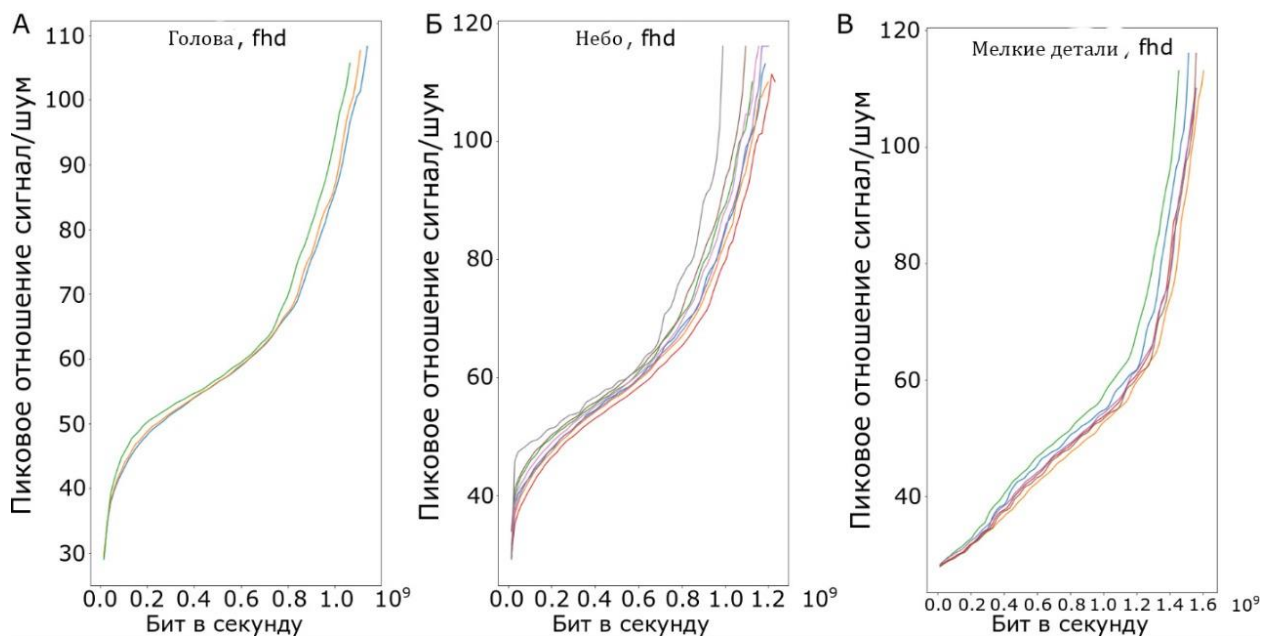


Рис. 8. Зависимость пикового отношения сигнала к шуму от пропускной способности сети (разрешение 2k, цветовая субдискретизация - 4:4:4, 25 кадров в секунду).

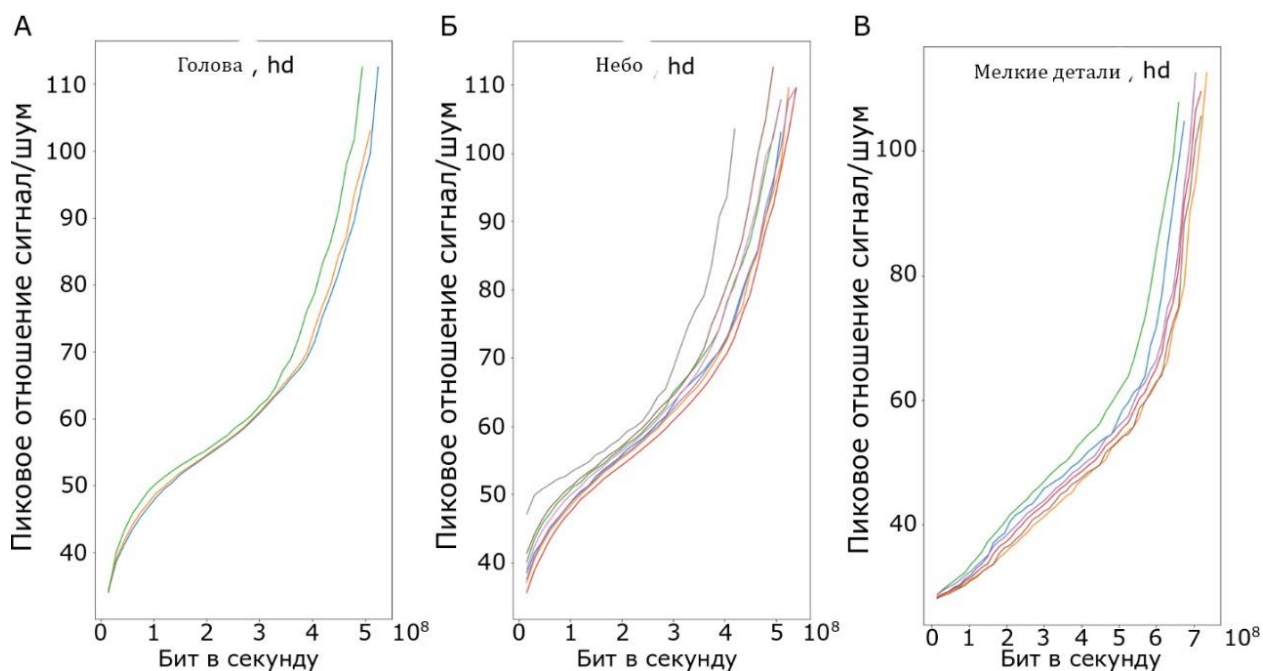


Рис. 9. Зависимость пикового отношения сигнала к шуму от пропускной способности сети (разрешение HD, цветовая субдискретизация - 4:4:4, 25 кадров в секунду).

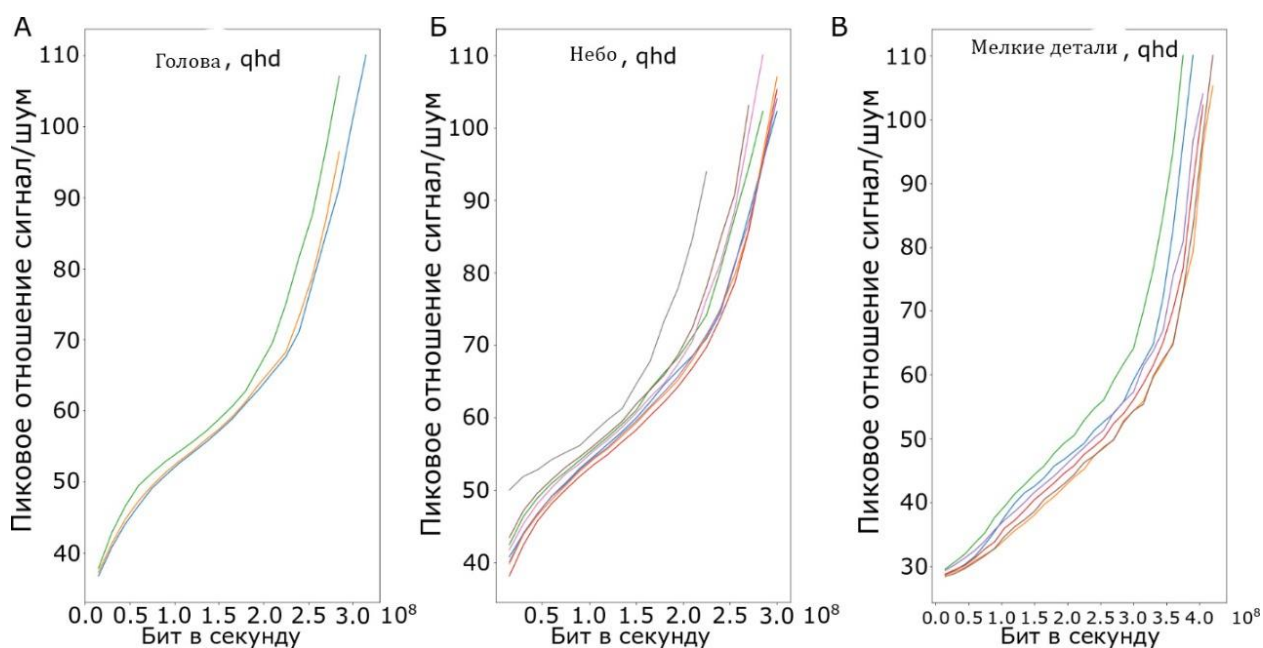


Рис. 10. Зависимость пикового отношения сигнала к шуму от пропускной способности сети (разрешение SD, цветовая субдискретизация - 4:4:4, 25 кадров в секунду).

На всех графиках (Рис. 7 – 10) чётко видна монотонная зависимость между качеством восстановления и битрейтом. Начальное значение битрейта на каждом графике зафиксировано на определённом начальном значении. Конечное значение определяется значением битрейта, после которого изображение начинает передаваться без потерь либо другим значением, дальше которого считать битрейт не целесообразно. На всех графиках (Рис.7

– Рис.10) стартовый битрейт равняется 1 Мбит/сек., конечный битрейт равняется 2 Гбит/сек., шаг битрейта равен 500 Кбит/сек. Можем наблюдать, что набор изображений “Мелкие детали” сжимается наименее эффективно (в плане качества). Это объясняется тем, что в этом наборе много ценной информации, которую нельзя выкидывать без потери объектов с кадра.

3.3. Исследование потери качества видео при множественном процессе кодирования и декодирования.

На Рис. 11 отображена потеря качества изображения (метрика PSNR) в зависимости от количества циклов пережатия изображения. Первый цикл пережатия использует оригинальное изображение, чтобы получить из неё первое восстановленное изображение. После замера качества восстановления, происходят последующие циклы пережатия восстановленного изображения с последующим измерением качества восстановленного изображения. Измерения происходят между оригиналом, и пережатым изображениях. Всего было 11 экспериментов пережатия для различных значений бит на пиксель (bpp).

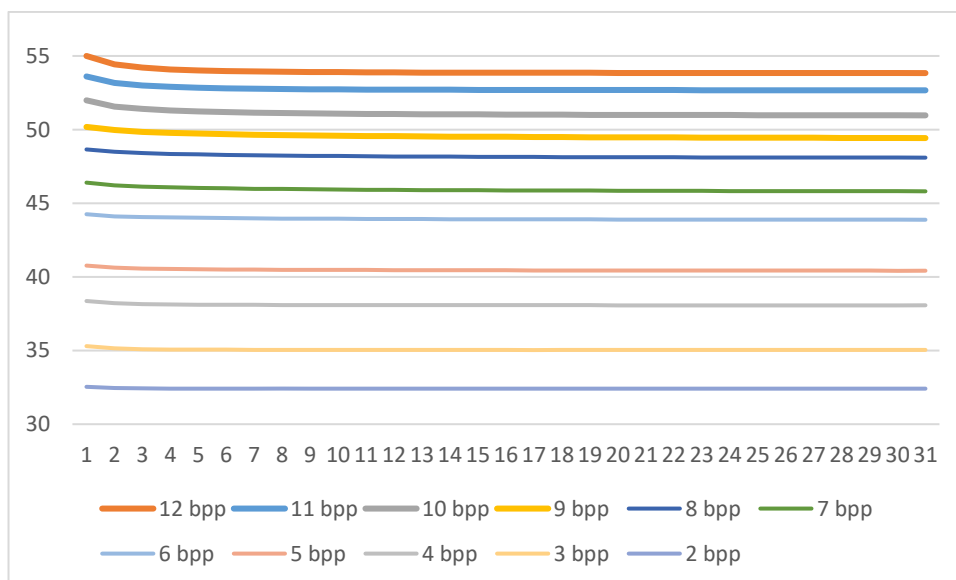


Рис. 11. Зависимость качества изображения от цикла пережатия. Оригинал 4:4:4, восстановленное изображение 4:4:4.

На Рис. 11 можно наблюдать эффект выхода качества восстановленного изображения на “плато”. После определённого значения количества пережатий качество практически перестаёт ухудшаться. Начальные значения качества восстановленных изображений более подробно можно рассмотреть на Рис. 12.

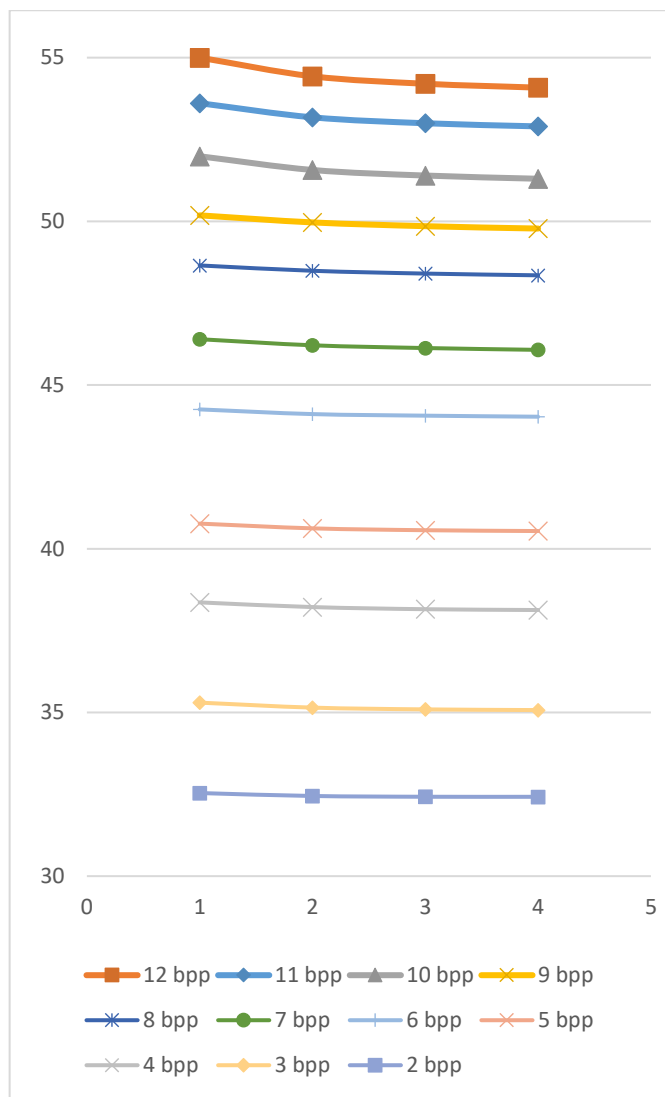


Рис. 12. Зависимость качества изображения от цикла пережатия.

3.4. Статистический анализ всех наборов изображений

На Рис. 13 отображена зависимость качества восстановления изображения от количества бит на пиксель. Красным цветом отображен набор изображений "Голова", синим цветом отображен набор изображений "Небо", зелёным цветом отображен набор изображений "Мелкие детали". Видно, что данные разбились на 2 статистические группы. В первой группе находятся наборы "Небо" и "Голова", во второй группе находится набор "Мелкие детали". Это объясняется тем, что только набор "Мелкие детали" полностью состоит из мелких деталей, а передача мелких деталей требует повышенного битрейта. На Рис. 13 можно наблюдать 3 группы изображений, обрывающихся на битрейте в 8 бит на пиксель. Однако рассмотреть можно 1 такую группу (зелёную), так как остальные не видно из-за наложения линий друг на друга. Эти группы изображений объединяет качество UHD, они имеют такой вид ввиду того, что не хватает битрейта, и линии оканчиваются на 8 бит на пиксель.

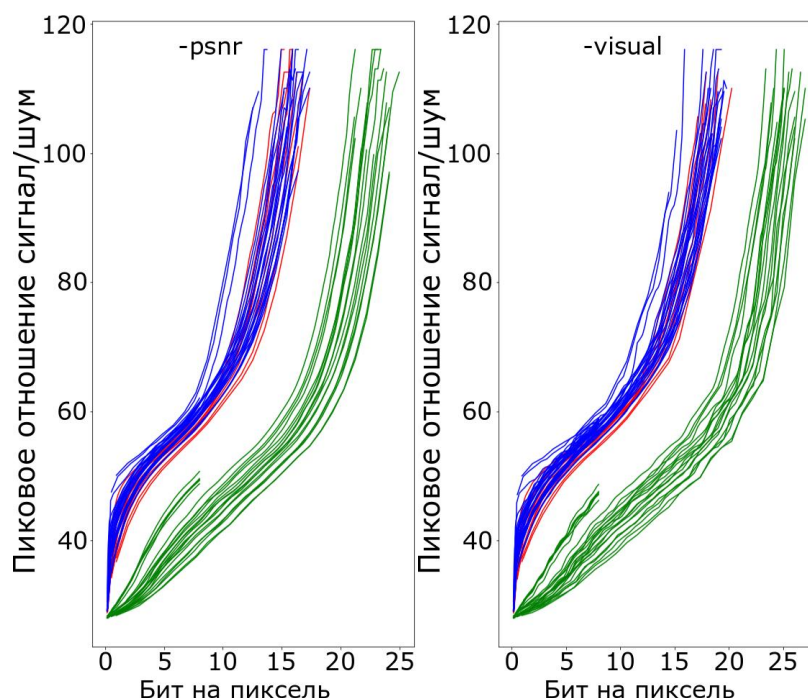


Рис. 13 Зависимость пикового отношения сигнал\шум от количества бит на пиксель изображения. Для графика слева использовался флаг -o psnr, для графика справа использовался флаг -o visual. Красным цветом отображен набор изображений "Голова", синим цветом отображен набор изображений "Небо", зелёным цветом отображен набор изображений "Мелкие детали".

3.5. Профилирование Jpeg XS test model (XSM)


На Рис. 14 изображён результат профилирования Jpeg XS decoder test model. Функция `idwt53` выполняет обратное вейвлет преобразование изображения, к которому было применено деквантование. Функция `bitunpacker_read` занимается чтением данных из не выровненного закодированного потока данных. Функция `unpack_precinct` занимается распаковкой закодированных слайсов. Из Рис. 14 следует, что описанные функции выполняются наибольшее количество времени и именно их следует оптимизировать.

Ведущие функции		
Имя функции	Общее время ЦП [единицы, %]	Собственное время ЦП [единицы, %] 🔥
<code>idwt53</code>	679 (44,21%)	672 (43,75%)
<code>bitunpacker_read</code>	203 (13,22%)	186 (12,11%)
<code>unpack_precinct</code>	449 (29,23%)	150 (9,77%)
[Внешний код]	1536 (100,00%)	134 (8,72%)

Критический путь		
Имя функции	Общее время ЦП [единицы, %]	Собственное время ЦП [единицы, %]
🔥 [Внешний код]	1536 (100,00%)	1 (0,07%)
🔥 <code>__sclr_common_main_seh</code>	1535 (99,93%)	0 (0,00%)
🔥 <code>main</code>	1535 (99,93%)	0 (0,00%)
🔥 <code>tco_dec_bitstream</code>	1535 (99,93%)	49 (3,19%)
🔥 <code>tco_dec_transforms</code>	941 (61,26%)	26 (1,69%)
🔥 <code>unpack_precinct</code>	449 (29,23%)	150 (9,77%)

Рис. 14. Результат профилирования Jpeg XS decoder.

На Рис. 15 изображён результат профилирования Jpeg XS encoder test model. Функции `gcli_budget_compute_generic` и `fill_data_budget_table` занимаются контролем битрейта. Функция `sig_flags_init` участвует в создании флагов значимости. Функция `dwt53` выполняет обратное вейвлет преобразование к исходному изображению. Из Рис. 15 следует, что описанные функции выполняются наибольшее количество времени и именно их следует оптимизировать.

Ведущие функции		
Имя функции	Общее время ЦП [единицы, %]	Собственное время ЦП [единицы, %] 
gcli_budget_compute_generic	5864 (63,67%)	4990 (54,18%)
fill_data_budget_table	1190 (12,92%)	1190 (12,92%)
sig_flags_init	864 (9,38%)	810 (8,79%)
dwt53	586 (6,36%)	552 (5,99%)
compute_residuals	612 (6,64%)	436 (4,73%)










Критический путь		
Имя функции	Общее время ЦП [единицы, %]	Собственное время ЦП [единицы, %]
 [Внешний код]	9209 (99,99%)	12 (0,13%)
 __sclr_common_main_seh	9197 (99,86%)	0 (0,00%)
 main	9197 (99,86%)	0 (0,00%)
 encode	9197 (99,86%)	0 (0,00%)
 tco_enc_image	9197 (99,86%)	109 (1,18%)
 rate_control_process_presinct	7813 (84,83%)	3 (0,03%)
 rate_control_refresh_line0	3538 (38,41%)	3 (0,03%)
 gcli_budget_compute_generic	2740 (29,75%)	2273 (24,68%)
 fill_data_budget_table	1190 (12,92%)	1190 (12,92%)

Рис. 15. Результат профилирования Jpeg XS encoder.

3.6. Jpeg XS как система реального времени

JPEG XS можно использовать для создания системы реального времени. Пример такой системы приведён на Рис. 16.

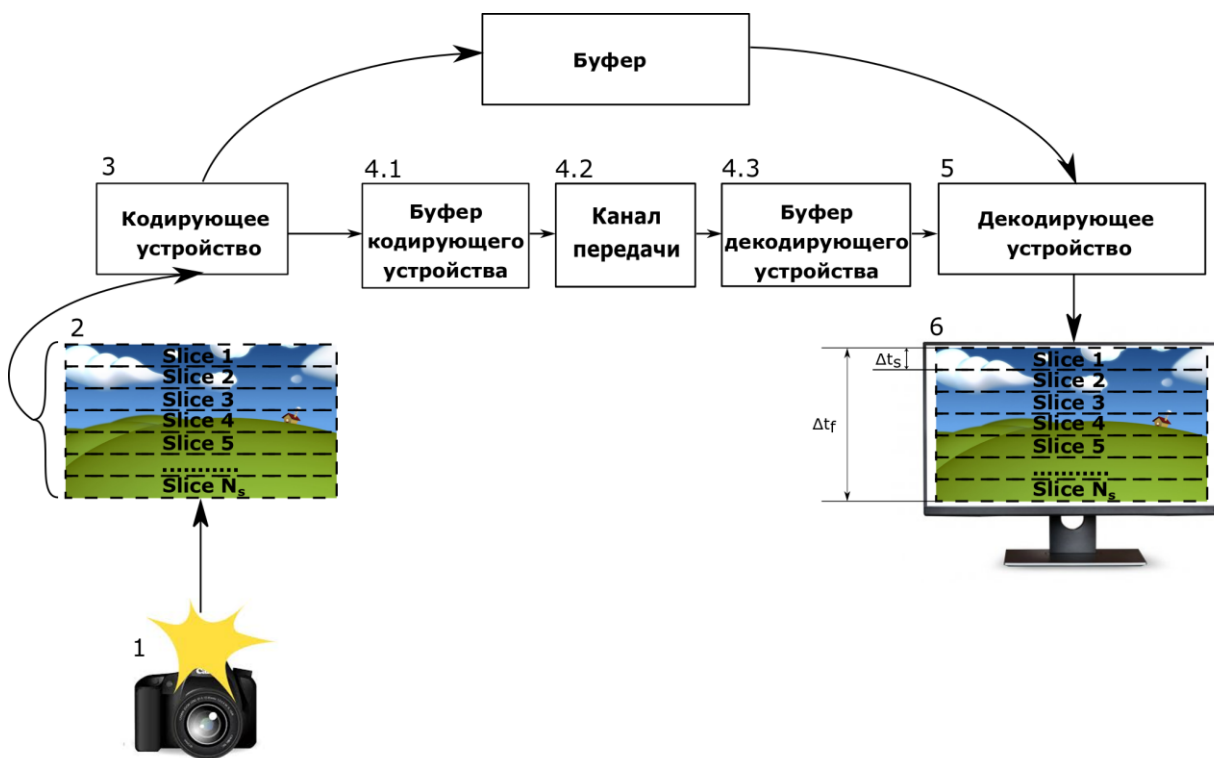


Рис. 16. Схема системы реального времени.

Исходник снимается на камеру. Далее будем работать с сырым изображением с камеры, либо опустим конвертацию в другие форматы. Важно понимать, что “сырое” изображение с камеры не имеет такого понятия как количество компонент на пиксель. Каждый пиксель сырого изображения — это значение с одного элемента матрицы камеры. Обычно матрицы камер представляют из себя фильтр Байера (Раздел 1.4). Изображение делится на слайсы (по 16 строк), каждый слайс может кодироваться независимо от остальных. Далее изображение попадает в кодирующее устройство. Под кодирующим устройством будем понимать encoder JPEG XS. Далее изображение попадает в буфер кодирующего устройства (нижний путь), либо в общий буфер между кодирующим устройством и декодирующим устройством (верхний путь). Из общего буфера изображение сразу попадает в декодирующее устройство, например, таким буфером может являться

оперативная память. Буфер используется из-за того, что кодировщик не способен поддерживать фиксированный темп выдачи данных. В стандарте JPEG XS такой буфер упоминается в виде термина “encoder smoothing buffer”. [7] Из буфера изображение попадает в канал передачи (например, Internet). Будем предполагать, что максимальное время передачи по сети ограничено сверху. Это требование к сети нельзя убрать, так как при этом система больше не будет являться СРВ. Из сети изображение попадает в буфер декодирующего устройства. Такой буфер упоминается в стандарте JPEG XS как “decoder smoothing buffer”. [7] Далее декодер считывает информацию из буфера, после чего мы имеем конечное восстановленное изображение.

Если мы сложим задержки передачи на каждом из этапов нижнего пути, то получим конечное время передачи, которое определяет эту систему как СРВ. Пусть t_1 , t_2 , t_3 и т.д. это задержки передачи изображения на этапах 1,2,3 и т.д на Рис. 16. Тогда $t_1 = 0$, т.к. реальное событие происходит без задержки, т.е. скорость передачи ограничена скоростью света, поэтому временем задержки можно пренебречь, t_2 – максимальное время получения изображения с матрицы камеры, t_3 – максимальное время кодирования изображения, $t_{4.1}$ – максимальное время нахождения изображения в буфере кодирующего устройства, $t_{4.2}$ – максимальное время отправки изображения по сети, $t_{4.3}$ – максимальное время нахождения изображения в буфере декодирующего устройства, t_5 – максимальное время декодирования сжатого изображения. Пусть пропускная способность сети – W_i , пусть пропускная способность декодера – W_d , а пропускная способность энкодера – W_e . Из системы следует требование, что $W_i = W_d = W_e = W$, иначе информация будет накапливаться в одном из узлов сети, что привело бы к потере информации. Например, если $W_e > W_i$, то будет происходить накопление информации в буфере кодирующего устройства. Просуммировав все времена задержек, получим следующую формулу:

$$t_{s2} = t_2 + t_3 + \frac{n_{enc}L}{W} + t_{4.2} + \frac{n_{dec}L}{W} + t_5, \quad (1)$$

где t_{s2} – максимальная суммарная задержка системы (нижний путь), n_{enc} – максимальное количество слайсов в буфере энкодера, n_{dec} – максимальное количество слайсов в буфере декодера, L – размер слайса.

Если мы сложим задержки передачи на каждом из этапов верхнего пути, то получим конечное время передачи, которое определяет эту систему как CPB.

$$t_{s1} = t_2 + t_3 + t_4 + t_5, \quad (2)$$

где t_{s1} – максимальная суммарная задержка системы (верхний путь), t_4 – время нахождения слайса в буфере.

В формуле (1) t_2 можно оценить в 0.01 - 1 мс, t_3 можно оценить в 10-40 мс. $n_{enc} = 2 - 10$ слайсов, $W = 1$ Мбит/сек – 2 Гбит/сек, $L = 16 \times (\text{ширина изображения в пикселях}) \times (\text{количество бит на пиксель})$. Ширина изображения варьируется от 720 (для изображения SD качества) до 3840 (для изображений UHD качества), а количество бит на пиксель варьируется от 2 до 12, таким образом L варьируется от 23 040 до 737 280 бит. Тогда задержка, обусловленная нахождением данных в буфере энкодера, может быть оценена 3 – 10 мс. Задержка t_5 может быть оценена для локальной сети в 1 – 10 мс. Задержка, обусловленная нахождением данных в буфере декодера, аналогично задержке в буфере энкодера, может быть оценена 3 – 10 мс. Задержка t_7 может быть оценена в 5 – 20 мс. Время нахождения в буфере t_4 может быть оценено как <1 мс. Таким образом, можно сделать вывод, что основная задержка набирается во время кодирования и декодирования информации.

5. Выводы

JPEG XS - это новый международный стандарт для кодирования изображений с малыми задержками и низкой потерей качества без видимых артефактов. Простое, но эффективное кодирование позволяет сохранить время ожидания и сложность очень низкими, и в то же время достичь качества восстановления изображения без видимых потерь. Исследование в данной работе кодека JPEG XS показало, что он применим для использования в системах реального времени, в разделе 3.5 подробно была описана схема CPB с использованием JPEG XS. Профилирование кодировщика показало, что наиболее время затратной операцией является контроль битрейта. Исходя из этого, в будущей работе возможна смена подхода к контролю битрейта. Профилирование декодировщика показало, что нужно оптимизировать дискретное вейвлет преобразование и считывание данных из закодированного потока. Было показано, что картинки с большим количеством мелких деталей теряют больше качества, чем картинки с меньшим количеством мелких деталей.

В результате выполненной работы решены следующие задачи:

1. Исследован кодек JPEG XS в условиях разной пропускной способности канала передачи.
2. Получена зависимость среднеквадратичной ошибки от степени сжатия изображения.
3. Создана симуляция программно-аппаратного комплекса для решения поставленной задачи и достижения цели.
4. Отпрофилирована текущая реализация jpegXS.
5. Была изучена применимость JPEG XS в системах реального времени.

6. Список литературы

1. Keinert, J., Lorent, J., Descampe, A., & Föbel, S. (2017). *Introduction To Jpeg Xs – the New Low Complexity Codec Standard for Professional Video Production*.
2. ISO/IEC JTC 1/SC 29/WG1, “Information technology -- Advanced image coding and evaluation methodologies -- Part 2: Evaluation Procedure for nearly lossless coding,” ISO/IEC 29170-2 (2015).
3. Buysschaert, C., Descampe, A., Föbe, S., Keinert, J., Pellegrin, P., Richter, T., & Rouvroy, G. (n.d.). *JPEG White paper : JPEG XS , a new standard for visually lossless low-latency lightweight image coding system*. www.jpeg.org
4. A Survey of Embedded Software Profiling Methodologies, Rajendra Patel, Arvind Rajwat, Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology, Bhopal, India
5. <https://docs.microsoft.com/ru-ru/visualstudio/profiling/profiling-featuretour?view=vs-2019>
6. *DRAFT INTERNATIONAL STANDARD ISO / IEC DIS 21122-5 Information technology — JPEG XS low-latency lightweight image coding system — Part 5 : Reference software*. (2019).
7. *DRAFT INTERNATIONAL STANDARD ISO / IEC DIS 21122-5 Information technology — JPEG XS low-latency lightweight image coding system — Part 2 : Profiles and buffer models*. (2019).