



Бесплатная электронная книга

УЧУСЬ video

Free unaffiliated eBook created from
Stack Overflow contributors.

#video

.....	1
1:	2
.....	2
Examples.....	2
.....	2
.....	3
.....	4
.....	5
.....	5
.....	6
2:	7
.....	7
Examples.....	7
.....	7
.....	8
.....	9
.....	10
.....	10
.....	11
.....	11
DASH.....	14
3:	19
.....	19
Examples.....	19
(DAR).....	19
(PAR).....	20
(SAR).....	20
.....	21
.....	22

Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [video](#)

It is an unofficial and free video ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official video.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с видео

замечания

Воспроизведение видео является фронтом и центром в широком диапазоне современных решений, при этом программное обеспечение и стандарты быстро развиваются. Чтобы понять это поле, вы должны сначала понять несколько аспектов, связанных с работой с видео:

- Информация о необработанном цвете, полученная из физического мира, обычно *кодируется* с использованием кодека - алгоритма, целью которого является представление этих данных в сжатой форме, часто жертвуя некоторыми визуальными деталями в пользу большего сжатия.
- Для воспроизведения выполняется обратный алгоритм - данные *декодируются*, чтобы снова стать необработанной информацией о цвете, которая может быть подана на устройство вывода (например, монитор).
- Между кодированием и декодированием сжатые данные *упаковываются* для хранения, что может включать объединение треков разных типов в один файл или сегментацию содержимого в большое количество небольших сегментов.
- Видео *доставляется* на устройство конечного пользователя с использованием технологии доставки, которая может быть такой же простой, как загрузка файла по протоколу HTTP или значительно сложнее, включая обратную связь с сетевой инфраструктурой и автоматическую адаптацию уровней качества.
- Премиум-контент обычно *зашифровывается* перед упаковкой и может воспроизводиться только в проигрывателе, оснащенный технологией DRM, которая обеспечивает безопасность ключа дешифрования во время использования и активно защищает от захвата вывода.

Хотя визуальная часть, очевидно, доминирует в видимости, аудио и текст также играют ключевую роль в мультимедийных презентациях, предоставляя многоязычные функции, которые делают контент доступным для широкой аудитории. В большинстве рабочих процессов звуковые и текстовые треки обрабатываются способом, эквивалентным видеодорожкам, кодируются, декодируются, упаковываются и передаются по тем же линиям.

Все эти аспекты - и многое другое - должны быть учтены в современном решении, обеспечивая приятный опыт для конечных пользователей.

Examples

Понимание автономных медиафайлов

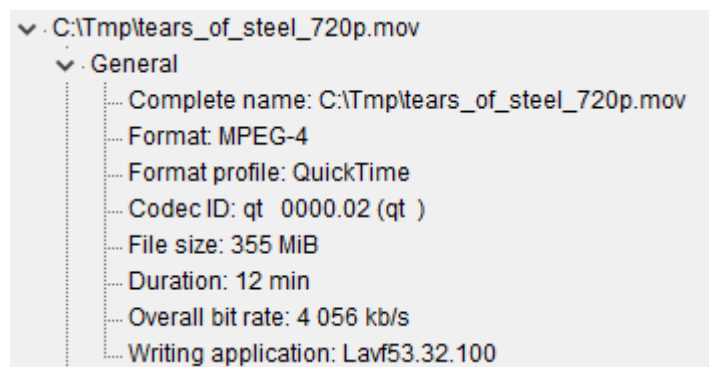
Используемый здесь образец - Tears of Steel, Blender Foundation. В частности, мы будем использовать загрузку под названием «HD 720p (~ 365MB, mov, 2.0)». Это единственный файл, который заканчивается расширением «mov» и будет воспроизводиться практически в любом современном медиаплеере.

Обратите внимание, что страница загрузки предлагает субтитры в виде отдельных загрузок файлов SRT. В этом примере контент не содержит субтитров, поставленных вместе в одном файле. Поэтому мы оставляем анализ субтитров вне сферы действия этого примера.

Простым способом анализа различных медиафайлов является использование инструмента / библиотеки [MediaInfo](#). Хотя представленная здесь функция анализа использует GUI для простоты, все функции также доступны через MediaInfo API.

Открыв этот файл в графическом интерфейсе MediaInfo и переключившись на древовидную структуру, вы увидите три раздела: «Общие», «Видео» и «Аудио». Первая содержит основную информацию о файле, тогда как остальные два описывают медиа-трек, найденный в этом файле. Давайте рассмотрим наиболее релевантную информацию в каждом разделе вывода.

генеральный



Первыми представляющими интерес параметрами являются *профиль Format* и *Format*. Первое указывает, что **формат упаковки** соответствует стандарту MPEG-4. MPEG-4 определяет *формат базового медиафайла ISO* и *формат упаковки MP4*. Кроме того, Apple разработала свою собственную спецификацию, которая вытекает из них, названная в MediaInfo как профиль «QuickTime».

Примечание. Будьте осторожны, чтобы не путать MP4 и MPEG-4 - первый относится к определенному формату упаковки в стандарте MPEG-4 международных стандартов, который также включает видео и аудио кодеки. Это может привести к путанице, поэтому избегайте использования термина MPEG-4 при обращении к чему-либо, кроме полного набора стандартов.

Все форматы упаковки, основанные на формате базового медиафайла ISO, определенные

в семействе стандартов MPEG-4, очень похожи и часто могут обрабатываться одними и теми же инструментами, причем их отличия во многом зависят от пользовательских расширений поставщиков, которые часто могут быть безопасно игнорируются. Таким образом, мы можем ожидать, что образец видео здесь будет очень совместим со всеми современными видеоплеерами.

ВИДЕО

```
Video
  ID: 1
  Format: AVC
  Format/Info: Advanced Video Codec
  Format profile: Main@L3.1
  Format settings, CABAC: No
  Format settings, ReFrames: 2 frames
  Format settings, GOP: M=4, N=18
  Codec ID: avc1
  Codec ID/Info: Advanced Video Coding
  Duration: 12 min
  Bit rate: 4 000 kb/s
  Width: 1 280 pixels
  Height: 534 pixels
  Display aspect ratio: 2.40:1
  Frame rate mode: Constant
  Frame rate: 24.000 FPS
  Color space: YUV
  Chroma subsampling: 4:2:0
  Bit depth: 8 bits
  Scan type: Progressive
  Bits/(Pixel*Frame): 0.244
  Stream size: 338 MiB (95%)
  Writing library: x264 core 122
  Encoding settings: cabac=0 / ref=2 / deblock=1:0:0 / ar
  Language: English
```

Наиболее важной деталью видео трека является кодек, используемый для преобразования необработанных цветовых данных в сжатую форму. Название кодека предоставляется параметром *Format*.

AVC также известен как H.264, и именно видеокodeк, который сегодня является самым распространенным, поддерживается практически всеми современными устройствами и программными платформами. Видео трек, закодированный с помощью AVC, обязательно будет воспроизводиться практически для любого игрока.

Кодеки часто имеют несколько *профилей*, которые позволяют разделить функциональность кодека на уровни, что позволяет эволюционировать технологии контролируемым образом. Параметр *профиля* указывает, что это видео использует основной профиль. Этот профиль относительно необычен, так как почти все современные устройства поддерживают высокий профиль, что обеспечивает большую

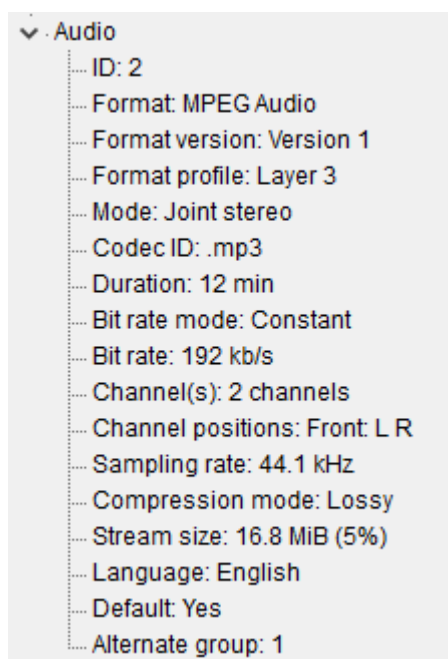
эффективность сжатия.

Качество видеодорожки часто имеет первостепенное значение. Здесь мы видим критические факторы, выраженные параметрами *скорости передачи* , *ширины* и *высоты* . Последние два намека на то, что это означает, что это видео трек 720p, который считается более низким качеством HD. Изображение на самом деле короче по вертикали, чем стандартный кадр 720p с разрешением 1280x720 пикселей.

Битовая скорость измеряет объем данных, которые в среднем приходится на сжатую форму видеопотока, в каждую секунду воспроизведения. Это ключевой параметр для оптимизации, поскольку объем доставленных данных является основным источником затрат в крупномасштабных видеорешениях.

Вышеуказанные данные о качестве видео - это просто факты, которые мы получаем из анализа - любые суждения о целесообразности этих параметров являются темой, которая потребует гораздо большего анализа и будет рассмотрена отдельными темами в этой категории документации, а также многими другими точками работа с видеодорожками.

аудио



Еще раз, знание кодека, используемого для кодирования аудиоданных, имеет решающее значение. Это выражается параметрами *профиля Format* и *Format* . «MPEG Audio Layer 3» более известен как MP3, и это универсальный аудиоформат, который можно ожидать повсюду.

Как и видео, параметры качества звука являются вторыми наиболее важными точками данных, выражаемыми в основном параметром *скорости передачи* .

Резюме анализа

Содержимое упаковано с использованием очень популярного формата упаковки, построенного на стандартном стандарте MPEG-4. Он кодируется с использованием общепринятых видео и аудио кодеков. Из этого ясно, что видео должно быть легко доступно для каждого зрителя - совместимость и доступность были ключевыми для его авторов.

Использование MP3 показывает возраст примерного контента, поскольку он больше не рассматривается в соответствии с современными конкурентами - вместо этого AAC (Advanced Audio Coding) является кормильцем в области аудиокодеков.

То же самое можно сказать об использовании основного профиля H.264. Очень редко используется любой профиль H.264, кроме High, учитывая, что почти все декодеры поддерживают его, позволяя каждому воспользоваться улучшенной эффективностью, предоставляемой функциями High profile.

Используемые битрейты немного выше, чем ожидалось для сегодняшней среды. Это может быть объяснено стремлением авторов к качеству или просто ограничениями кодеров, которые были доступны при создании контента.

Больше

Другими полезными инструментами для анализа [медиафайлов](#) являются [FFprobe](#), который является частью программного пакета FFmpeg и [инструментов Bento4](#) для работы с файлами MP4. Оба они также доступны в библиотечной форме. Они способны к более низкоуровневому анализу, чем MediaInfo, в ситуациях, когда вам нужно изучить отдельные элементы, составляющие медиафайлы.

Прочитайте [Начало работы с видео онлайн](#): <https://riptutorial.com/ru/video/topic/5690/начало-работы-с-видео>

глава 2: Понимание презентации СМІ

замечания

Воспроизведение видео является фронтом и центром в широком диапазоне современных решений, при этом программное обеспечение и стандарты быстро развиваются. Чтобы понять, что составляет медиа-презентацию, вы должны сначала понять несколько аспектов, связанных с работой с видео:

- Информация о необработанном цвете, полученная из физического мира, обычно *кодируется* с использованием кодека - алгоритма, целью которого является представление этих данных в сжатой форме, часто жертвуя некоторыми визуальными деталями в пользу большего сжатия.
- Для воспроизведения выполняется обратный алгоритм - данные *декодируются*, чтобы снова стать необработанной информацией о цвете, которая может быть подана на устройство вывода (например, монитор).
- Между кодированием и декодированием сжатые данные *упаковываются* для хранения, что может включать объединение треков разных типов в один файл или сегментацию содержимого в большое количество небольших сегментов.
- Видео *доставляется* на устройство конечного пользователя с использованием технологии доставки, которая может быть такой же простой, как загрузка файла по протоколу HTTP или значительно сложнее, включая обратную связь с сетевой инфраструктурой и автоматическую адаптацию уровней качества.
- Премиум-контент обычно *зашифровывается* перед упаковкой и может воспроизводиться только в проигрывателе, оснащенный технологией DRM, которая обеспечивает безопасность ключа дешифрования во время использования и активно защищает от захвата вывода.

Хотя визуальная часть, очевидно, доминирует в видимости, аудио и текст также играют ключевую роль в мультимедийных презентациях, предоставляя многоязычные функции, которые делают контент доступным для широкой аудитории. В большинстве рабочих процессов звуковые и текстовые треки обрабатываются способом, эквивалентным видеодорожкам, кодируются, декодируются, упаковываются и передаются по тем же линиям.

Все эти аспекты - и многое другое - влияют на состав, форматирование и использование медиапрезентации. Их необходимо понимать как целое, чтобы эффективно использовать медиатехнологии.

Examples

Понимание автономных медиафайлов

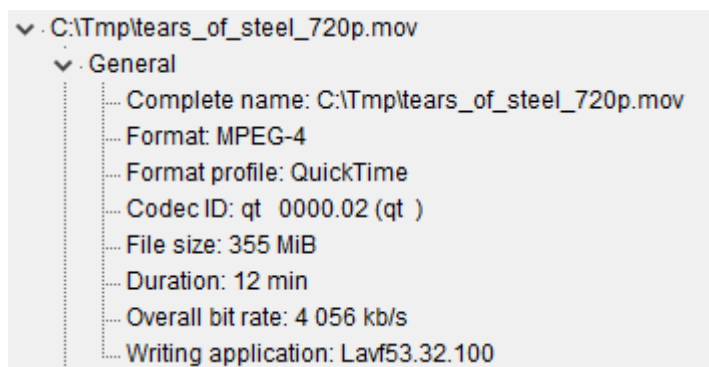
Используемый здесь образец - [Tears of Steel](#) , Blender Foundation. В частности, мы будем использовать загрузку под названием «HD 720p (~ 365MB, mov, 2.0)». Это единственный файл, который заканчивается расширением «mov» и будет воспроизводиться практически в любом современном медиаплеере.

Обратите внимание, что страница загрузки предлагает субтитры в виде отдельных загрузок файлов SRT. В этом примере контент не содержит субтитров, поставленных вместе в одном файле. Поэтому мы оставляем анализ субтитров вне сферы действия этого примера.

Простым способом анализа различных медиафайлов является использование инструмента / библиотеки [MediaInfo](#) . Хотя представленная здесь функция анализа использует GUI для простоты, все функции также доступны через MediaInfo API.

Открыв этот файл в графическом интерфейсе MediaInfo и переключившись на древовидную структуру, вы увидите три раздела: «Общие», «Видео» и «Аудио». Первая содержит основную информацию о файле, тогда как остальные два описывают медиа-трек, найденный в этом файле. Давайте рассмотрим наиболее релевантную информацию в каждом разделе вывода.

генеральный



Первыми представляющими интерес параметрами являются *профиль Format* и *Format* . Первое указывает, что **формат упаковки** соответствует стандарту MPEG-4. MPEG-4 определяет *формат базового медиафайла ISO* и *формат упаковки MP4* . Кроме того, Apple разработала свою собственную спецификацию, которая вытекает из них, названная в MediaInfo как профиль «QuickTime».

Примечание. Будьте осторожны, чтобы не путать MP4 и MPEG-4 - первый относится к определенному формату упаковки в стандарте MPEG-4 международных стандартов, который также включает видео и аудио кодеки. Это может привести к путанице, поэтому избегайте использования термина MPEG-4 при обращении к чему-либо, кроме полного набора стандартов.

Все форматы упаковки, основанные на формате базового медиафайла ISO, определенные

в семействе стандартов MPEG-4, очень похожи и часто могут обрабатываться одними и теми же инструментами, причем их отличия во многом зависят от пользовательских расширений поставщиков, которые часто могут быть безопасно игнорируются. Таким образом, мы можем ожидать, что образец видео здесь будет очень совместим со всеми современными видеоплеерами.

ВИДЕО

```
Video
  ID: 1
  Format: AVC
  Format/Info: Advanced Video Codec
  Format profile: Main@L3.1
  Format settings, CABAC: No
  Format settings, ReFrames: 2 frames
  Format settings, GOP: M=4, N=18
  Codec ID: avc1
  Codec ID/Info: Advanced Video Coding
  Duration: 12 min
  Bit rate: 4 000 kb/s
  Width: 1 280 pixels
  Height: 534 pixels
  Display aspect ratio: 2.40:1
  Frame rate mode: Constant
  Frame rate: 24.000 FPS
  Color space: YUV
  Chroma subsampling: 4:2:0
  Bit depth: 8 bits
  Scan type: Progressive
  Bits/(Pixel*Frame): 0.244
  Stream size: 338 MiB (95%)
  Writing library: x264 core 122
  Encoding settings: cabac=0 / ref=2 / deblock=1:0:0 / ar
  Language: English
```

Наиболее важной деталью видео трека является кодек, используемый для преобразования необработанных цветовых данных в сжатую форму. Название кодека предоставляется параметром *Format*.

AVC также известен как H.264, и именно видеокодек, который сегодня является самым распространенным, поддерживается практически всеми современными устройствами и программными платформами. Видео трек, закодированный с помощью AVC, обязательно будет воспроизводиться практически для любого игрока.

Кодеки часто имеют несколько *профилей*, которые позволяют разделить функциональность кодека на уровни, что позволяет эволюционировать технологии контролируемым образом. Параметр *профиля* указывает, что это видео использует основной профиль. Этот профиль относительно необычен, так как почти все современные устройства поддерживают высокий профиль, что обеспечивает большую

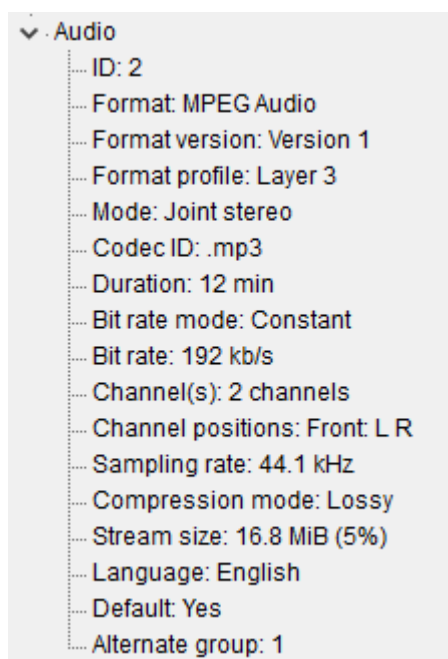
эффективность сжатия.

Качество видеодорожки часто имеет первостепенное значение. Здесь мы видим критические факторы, выраженные параметрами *скорости передачи* , *ширины* и *высоты* . Последние два намека на то, что это означает, что это видео трек 720p, который считается более низким качеством HD. Изображение на самом деле короче по вертикали, чем стандартный кадр 720p с разрешением 1280x720 пикселей.

Битовая скорость измеряет объем данных, которые в среднем приходится на сжатую форму видеопотока, в каждую секунду воспроизведения. Это ключевой параметр для оптимизации, поскольку объем доставленных данных является основным источником затрат в крупномасштабных видеорешениях.

Вышеуказанные данные о качестве видео - это просто факты, которые мы получаем из анализа - любые суждения о целесообразности этих параметров являются темой, которая потребует гораздо большего анализа и будет рассмотрена отдельными темами в этой категории документации, а также многими другими точками работа с видеодорожками.

аудио



Еще раз, знание кодека, используемого для кодирования аудиоданных, имеет решающее значение. Это выражается параметрами *профиля Format* и *Format* . «MPEG Audio Layer 3» более известен как MP3, и это универсальный аудиоформат, который можно ожидать повсюду.

Как и видео, параметры качества звука являются вторыми наиболее важными точками данных, выражаемыми в основном параметром *скорости передачи* .

Резюме анализа

Содержимое упаковано с использованием очень популярного формата упаковки, построенного на стандартном стандарте MPEG-4. Он кодируется с использованием общепринятых видео и аудио кодеков. Из этого ясно, что видео должно быть легко доступно для каждого зрителя - совместимость и доступность были ключевыми для его авторов.

Использование MP3 показывает возраст примерного контента, поскольку он больше не рассматривается в соответствии с современными конкурентами - вместо этого AAC (Advanced Audio Coding) является кормильцем в области аудиокодеков.

То же самое можно сказать об использовании основного профиля H.264. Очень редко используется любой профиль H.264, кроме High, учитывая, что почти все декодеры поддерживают его, позволяя каждому воспользоваться улучшенной эффективностью, предоставляемой функциями High profile.

Используемые битрейты немного выше, чем ожидалось для сегодняшней среды. Это может быть объяснено стремлением авторов к качеству или просто ограничениями кодеров, которые были доступны при создании контента.

Больше

Другими полезными инструментами для анализа [медиафайлов](#) являются [FFprobe](#), который является частью программного пакета FFmpeg и [инструментов Bento4](#) для работы с файлами MP4. Оба они также доступны в библиотечной форме. Они способны к более низкоуровневому анализу, чем MediaInfo, в ситуациях, когда вам нужно изучить отдельные элементы, составляющие медиафайлы.

Понимание компонентов видеодорожек

В этом примере мы рассмотрим, как увидеть компоновку видеодорожки и как извлечь отдельные изображения внутри нее.

Используемый здесь образец - [Tears of Steel](#), Blender Foundation. В частности, мы будем использовать загрузку под названием «HD 720p (~ 365MB, mov, 2.0)». Это единственный файл, который заканчивается расширением «mov» и будет воспроизводиться практически в любом современном медиаплеере.

Мы будем использовать инструменты mp4info и mp4dump из [набора Bento4](#) для отслеживания компоновки и анализа структуры и [FFmpeg](#) для извлечения отдельных изображений, составляющих видеодорожку.

В образце фильма используется формат упаковки «QuickTime» (MOV), основанный на формате ISO Media File Format - международный стандарт, который лежит в основе всех форматов упаковки в семействе форматов файлов MP4. Это делает его очень совместимым с большинством доступных инструментов и позволяет легко анализировать.

Давайте сначала рассмотрим общую структуру файла. Все медиафайлы, основанные на ISO Base Media File Format, структурированы как иерархия ящиков - мини-файловая система. Используйте утилиту `mp4dump`, чтобы извлечь структуру ящика, выполнив следующую команду:

```
mp4dump tears_of_steel_720p.mov
```

Вывод будет похож на следующий:

```
[ftyp] size=8+12
  major_brand = qt
  minor_version = 200
  compatible_brand = qt
[wide] size=8+0
[mdat] size=8+371579623
[moov] size=8+598972
  [mvhd] size=12+96
    timescale = 1000
    duration = 734167
    duration(ms) = 734167
  [trak] size=8+244250
    [tkhd] size=12+80, flags=f
      enabled = 1
      id = 1
      duration = 734167
      width = 1280.000000
      height = 534.000000
  ...
```

Это представляет собой внутреннюю структуру файла. Например, вы видите здесь поле `moov` с 8-байтовым заголовком и 598972 байтами содержимого. Этот ящик является контейнером для различных блоков метаданных, которые описывают содержимое файла. Для получения дополнительной информации о значении различных ящиков и их свойств см. [ИСО / МЭК 14496-12](#).

Фактические образцы самих медиа - сжатые изображения и звуковые сигналы - хранятся в поле `mdat`, содержимое которого непрозрачно утилите `mp4dump`.

Чтобы исключить нерелевантные данные и упростить рабочий процесс анализа - этот пример фокусируется на видеодорожке - теперь мы удаляем звуковую дорожку из нашего образца фильма. Выполните следующую команду:

```
ffmpeg -i tears_of_steel_720p.mov -an -vcodec copy video_track.mov
```

Обратите внимание, что вышеупомянутый шаг также удалит различные

пользовательские элементы расширения из входного видео, упаковывая суть визуального контента в новый файл контейнера и отбрасывая что-либо еще. Если вы сделаете это в производственном сценарии, убедитесь, что вы действительно свободны отбрасывать все остальные элементы во входном файле!

Закодированные видеодорожки представляют собой последовательность изображений. С используемым здесь кодеком H.264 - и всеми другими обычно используемыми современными кодеками - изображения могут быть разных типов:

- I-изображения - это независимые снимки, декодируемые исключительно с использованием данных, содержащихся в изображении.
- P-изображения - они берут другое изображение в качестве базовой линии и применяют преобразование к этому изображению (например, «переместите эти конкретные пиксели вправо на 5 пикселей»).
- B-изображения - похожие на P-кадры, но двунаправленные - они также могут ссылаться на изображения из будущего и определять преобразования, такие как «эти конкретные пиксели, которые будут полностью видимы в 5 кадрах, теперь на 10% видимы».

Точная комбинация типов изображений может быть свободно выбрана с помощью рабочего процесса кодирования, создавая множество возможностей оптимизации, хотя некоторые варианты использования могут ограничить доступную гибкость, например, требуя, чтобы I-кадр присутствовал ровно через 2 секунды.

Выполните следующую команду, чтобы увидеть структуру изображения видеодорожки:

```
mp4info --show-layout video_track.mov
```

Помимо представления общедоступной формы общих метаданных файла, вы увидите подробную распечатку макета изображения видеодорожки.

```
...
00000959 [V] (1) size= 7615, offset=15483377, dts=491008 (39958 ms)
00000960 [V] (1) * size=104133, offset=15490992, dts=491520 (40000 ms)
00000961 [V] (1) size= 16168, offset=15595125, dts=492032 (40042 ms)
00000962 [V] (1) size= 4029, offset=15611293, dts=492544 (40083 ms)
00000963 [V] (1) size= 24615, offset=15615322, dts=493056 (40125 ms)
00000964 [V] (1) size= 4674, offset=15639937, dts=493568 (40167 ms)
00000965 [V] (1) size= 18451, offset=15644611, dts=494080 (40208 ms)
00000966 [V] (1) size= 95800, offset=15663062, dts=494592 (40250 ms)
00000967 [V] (1) size= 30271, offset=15758862, dts=495104 (40292 ms)
00000968 [V] (1) size= 10997, offset=15789133, dts=495616 (40333 ms)
00000969 [V] (1) size= 28458, offset=15800130, dts=496128 (40375 ms)
00000970 [V] (1) size= 9593, offset=15828588, dts=496640 (40417 ms)
00000971 [V] (1) size= 24548, offset=15838181, dts=497152 (40458 ms)
00000972 [V] (1) size= 6853, offset=15862729, dts=497664 (40500 ms)
00000973 [V] (1) size= 27698, offset=15869582, dts=498176 (40542 ms)
00000974 [V] (1) size= 7565, offset=15897280, dts=498688 (40583 ms)
```



```
00000975 [V] (1) size= 24682, offset=15904845, dts=499200 (40625 ms)
00000976 [V] (1) size= 5535, offset=15929527, dts=499712 (40667 ms)
00000977 [V] (1) size= 38360, offset=15935062, dts=500224 (40708 ms)
00000978 [V] (1)* size= 82466, offset=15973422, dts=500736 (40750 ms)
00000979 [V] (1) size= 13388, offset=16055888, dts=501248 (40792 ms)
00000980 [V] (1) size= 2315, offset=16069276, dts=501760 (40833 ms)
00000981 [V] (1) size= 21983, offset=16071591, dts=502272 (40875 ms)
00000982 [V] (1) size= 3384, offset=16093574, dts=502784 (40917 ms)
00000983 [V] (1) size= 22225, offset=16096958, dts=503296 (40958 ms)
...
```

Каждая строка в этой распечатке представляет собой изображение, содержащееся в видеодорожке. Те, которые отмечены звездочкой как (1)* являются I-картинками. Вы можете видеть, как они являются самыми большими по размеру, а другие обеспечивают большее сжатие, ссылаясь на существующие изображения и описывая только различия.

В листинге также содержится смещение данных изображения в видеофайле и временная метка декодирования изображения, что обеспечивает дополнительную корреляцию и анализ. Обратите внимание, что порядок / время декодирования изображений не обязательно совпадают с порядком / выбором представления! Если на видео есть B-изображения, они могут быть декодированы только *после* любых изображений, которые они ссылаются, даже если они представлены *перед* ссылками на изображения!

Получив некоторое представление о структуре видеодорожки, выполните следующую команду, чтобы извлечь 30 изображений, начиная с отметки 40 секунд, в виде файлов PNG:

```
ffmpeg -i video_track.mov -ss 00:00:40 -vframes 30 picture%04d.png
```

Выбранные снимки будут полностью декодированы, так как они появятся в видеоплеере - невозможно (без особо специализированного инструментария) получить визуальное представление необработанных данных в P-кадрах или B-кадрах.

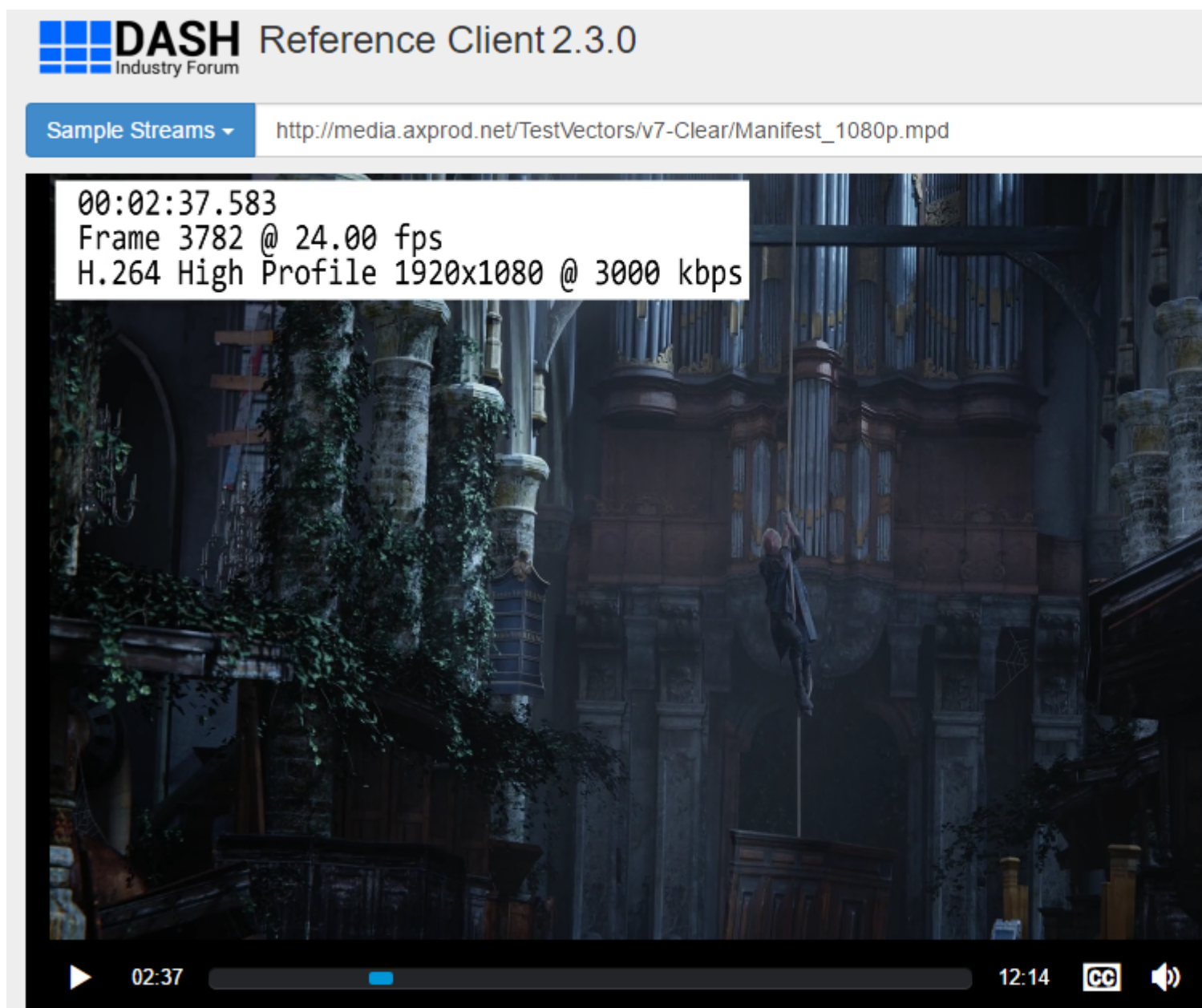
Наблюдайте за тем, как 7-го сгенерированное изображение является полной сменой сцены в видео. Вы можете легко соотнести это с выходом `tr4info` выше - седьмое изображение, начиная с отметки 40 секунд (номер 00000966), намного больше размера, чем те, что рядом с ним. Сцены трудно кодировать, поскольку они обновляют всю картину и содержат много новых данных. Если кодери не дано достаточной снисходительности для оптимизации изменений сцены (т. Е. Не разрешено создавать большое изображение), визуальный вывод будет низкого качества или «блочным» до следующего I-изображения. Изучая распределение полосы пропускания (байтов) на различные изображения, вы можете получить представление о таких визуальных артефактах, которые могут внезапно появиться в видео.

Понимание адаптивных потоковых презентаций DASH

DASH является наиболее широко используемой адаптивной потоковой технологией в современных решениях, используемых для доставки видео в самых разных сценариях. Лучший способ понять презентации DASH - наблюдать за сетевой активностью, которая происходит во время воспроизведения.

В этом примере используется [Fiddler](#) для захвата и анализа сетевого трафика браузера, хотя любой подобный инструмент также будет достаточным. Мы будем использовать проигрыватель с открытым исходным кодом [dash.js](#) для воспроизведения видео.

Для нашего демонстрационного контента мы будем использовать [тестовые векторы Axinom DASH](#), в частности [однопериодный](#) вариант 1080p тестового вектора «Clear».



The screenshot shows the DASH Reference Client 2.3.0 interface. At the top, there is a logo for the DASH Industry Forum and the text "DASH Reference Client 2.3.0". Below this, there is a "Sample Streams" dropdown menu and a text input field containing the URL "http://media.axprod.net/TestVectors/v7-Clear/Manifest_1080p.mpd". The main area is a video player showing a scene from a movie. A white box in the top left corner of the video player displays the following metadata: "00:02:37.583", "Frame 3782 @ 24.00 fps", and "H.264 High Profile 1920x1080 @ 3000 kbps". The video player has a progress bar at the bottom showing the current time as 02:37 and the total duration as 12:14. There are also icons for closed captions and audio.

С запуском сетевого захвата откройте [dash.js](#) [ночной образец образца](#) в любом современном браузере и введите URL-адрес http://media.axprod.net/TestVectors/v6-Clear/Manifest_1080p.mpd в текстовое поле. Нажмите «Загрузить», чтобы начать воспроизведение. Вы будете следить за загрузкой следующих файлов:

```
http://media.axprod.net/TestVectors/v7-Clear/Manifest_1080p.mpd
http://media.axprod.net/TestVectors/v7-Clear/2/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/15/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/18/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/18/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/2/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/15/0001.m4s
http://media.axprod.net/TestVectors/v7-Clear/15/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/5/init.mp4
http://media.axprod.net/TestVectors/v7-Clear/5/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/18/0002.m4s
http://media.axprod.net/TestVectors/v7-Clear/5/0003.m4s
...
```

Первый файл - манифест презентации - документ XML, формат которого определен в [ISO / IEC 23009-1](#) . Это описывает представление DASH на глубину, достаточную для того, чтобы игрок мог понять, как его воспроизвести.

Если вы посмотрите в манифест, вы увидите различные элементы `AdaptationSet` , каждый из которых описывает одну адаптацию содержимого. Например, для видеодорожки есть один набор адаптации, три набора адаптации для трех языков звукового сопровождения и пять наборов адаптации для пяти языков субтитров.

Внутри наборов адаптации `Representation` элементы `Representation` . Для набора видео адаптации есть несколько из них - каждое представление содержит один и тот же визуальный контент, закодированный с использованием другого уровня качества. Каждый набор адаптации аудио и текста имеет только одно представление.

```
AdaptationSet [ segmentAlignment=true group=1 maxWidth=1920 maxHeight=1080 maxFrameRate=24 par=16:9 ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="1200000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" dur
  <Representation id="8" mimeType="video/mp4" codecs="hev1.2.4.L63.90" width="512" height="288" frameRat
  <Representation id="9" mimeType="video/mp4" codecs="hev1.2.4.L63.90" width="640" height="360" frameRat
  <Representation id="10" mimeType="video/mp4" codecs="hev1.2.4.L90.90" width="852" height="480" frameRa
  <Representation id="11" mimeType="video/mp4" codecs="hev1.2.4.L93.90" width="1280" height="720" frameF
  <Representation id="12" mimeType="video/mp4" codecs="hev1.2.4.L120.90" width="1920" height="1080" fram
AdaptationSet [ segmentAlignment=true group=2 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" dur
  <Representation [ id=15 mimeType=audio/mp4 codecs=mp4a.40.29 audioSamplingRate=48000 startWithSAP=1 b
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" val
AdaptationSet [ segmentAlignment=true group=2 lang=en-AU ]
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" dur
  <Representation [ id=16 mimeType=audio/mp4 codecs=mp4a.40.29 audioSamplingRate=48000 startWithSAP=1 b
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" val
AdaptationSet [ segmentAlignment=true group=2 lang=et-ET ]
  <SegmentTemplate timescale="24000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" dur
  <Representation [ id=17 mimeType=audio/mp4 codecs=mp4a.40.29 audioSamplingRate=48000 startWithSAP=1 b
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" val
AdaptationSet [ segmentAlignment=true group=3 lang=en ]
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="subtitle" xmlns="urn:mpeg:dash:schema:mpd:2011" />
  <SegmentTemplate timescale="1000" media="$RepresentationID$/$Number%04d$.m4s" startNumber="1" durat
  <Representation id="18" mimeType="application/mp4" codecs="wvtt" startWithSAP="1" bandwidth="428" xmlns
```

Чтобы выполнить воспроизведение, игроку необходимо будет решить, какие наборы

адаптации будут представлены для зрителя. Он может принять это решение на основе любой пользовательской или встроенной бизнес-логики, которую он хочет (например, предпочтительный язык). Адаптивные наборы, которые автор содержимого считает первичными, имеют элемент `Role` в манифесте, объявляющий «главную» роль.

Кроме того, игроку необходимо будет решить, какое представление представить для зрителя (если набор адаптации предлагает несколько представлений). Большинство игроков начинают консервативно и применяют эвристический алгоритм, который попытается представить максимальный уровень качества, который может поддерживать сетевое соединение зрителя.

Игрок может свободно изменять активный набор представлений и / или наборов адаптации в любое время, либо в ответ на действие пользователя (выбор другого языка), либо автоматическую логику (эвристика пропускной способности приводит к изменению уровня качества).

Элемент `SegmentTemplate` определяет структуру URL, которую игрок может использовать для доступа к различным представлениям. Ключевым фактором презентаций DASH является то, что контент разбивается на небольшие сегменты в несколько секунд каждый (4 секунды в случае нашего образца фильма), которые загружаются независимо. Каждое представление также имеет сегмент инициализации, названный «init.mp4» для этого образцового фильма, который содержит конфигурацию декодера конкретного представления и поэтому должен быть загружен до того, как любой другой сегмент из этого представления может быть обработан.

Описанное здесь поведение является точным для профиля DASH Live, который является наиболее часто используемым вариантом DASH. Существуют также другие профили с немного отличающимся поведением, которые здесь не рассматриваются. Обратите внимание на атрибут «profile» в корневом элементе DASH-манифеста, чтобы убедиться, что это описание относится к вашим видео!

Когда вы просматриваете список URL-адресов, полученных из захвата сетевого трафика и сравнивающих с информацией, предоставленной манифестом, вы сделаете вывод, что после загрузки манифеста игрок выполнил следующие действия:

1. Загрузите сегменты инициализации для представлений 2 (360p видео), 15 (английское аудио) и 18 (английские субтитры).
2. Загрузите первые сегменты из трех представленных представлений (0001.m4s).
3. Загрузите второй сегмент аудио представления.
4. Со второго сегмента видео, переключитесь на видеопоток 1080p! Это указывается путем загрузки сегмента инициализации и второго сегмента представления 5 (видео 1080p).
5. Продолжайте загружать больше сегментов активных представлений.

Наблюдая за сетевой активностью, становится легко наблюдать за решениями, которые

использует адаптивный потоковый проигрыватель DASH. Такой проигрыватель - это просто механизм, который загружает сегменты различных дорожек и последовательно передает их в механизм воспроизведения мультимедиа, соответственно переключая дорожки.

Тест- [тестеры Axinom DASH](#) также содержат архивные файлы, которые позволяют загружать всю презентацию для анализа на уровне файловой системы. Вы обнаружите, что файлы на диске точно такие же, как на сетевом уровне. Это означает, что DASH-презентации могут обслуживаться произвольными HTTP-серверами без необходимости какой-либо пользовательской логики на стороне сервера.

Аспект Live DASH, который усложняет анализ, заключается в том, что образцы СМИ распространяются по большому числу сегментов. Большинство средств анализа средств массовой информации не могут обрабатывать отдельные сегменты, работающие только на всех дорожках. Вы можете часто преодолевать это ограничение, просто конкатенируя сегменты одного представления, начиная с сегмента инициализации. Например, в Windows вы можете использовать следующую команду:

```
copy init.mp4 /b + 0001.mp4 /b + 0002.mp4 /b + 0003.mp4 /b track.mp4
```

Это создаст файл track.mp4, содержащий первые три сегмента мультимедиа из представления. Хотя они не идентичны по структуре автономному файлу MP4, такой файл может быть проанализирован большинством инструментов (например, mp4info и FFmpeg) без существенной потери функциональности.

Прочитайте [Понимание презентации СМИ онлайн](https://riptutorial.com/ru/video/topic/5879/): <https://riptutorial.com/ru/video/topic/5879/>
[понимание-презентации-сми](#)

глава 3: Соотношение сторон видео

замечания

Соотношения сторон часто выражаются как отношение ширины и высоты, которое часто, но не всегда, упрощается, а иногда и как простое целое число с плавающей запятой.

Все следующие соотношения сторон - это одно и то же значение, выраженное различными способами:

- 1280: 720
- 16: 9
- 1,7777777777777777777777777777778

Examples

Соотношение сторон экрана (DAR)



1280

Display Aspect Ratio (DAR) 16:9

Это скриншот воспроизведения видео. Вы видите нормальное видео 16: 9, которое вы ожидаете увидеть в любом современном видеорешении. Это - соотношение сторон, которое видит зритель, - это так называемое *соотношение сторон экрана* или DAR.

Из показанных параметров видно, что $DAR = 1280: 720 = 16: 9 = 1,7777777777777777777777777777778$.

Соотношение сторон изображения (PAR)

Внутри, все видео - это просто серия фотографий. Давайте посмотрим на одну такую картину.



Это выглядит странно, не так ли? В самом деле. Изображения, составляющие видео, могут иметь соотношение сторон, которые отличаются от DAR, чаще всего по алгоритмическим причинам (например, только размер, который кратен 16, может быть сжат выбранным алгоритмом). Это называется *соотношением сторон изображения* или PAR.

В этом примере размеры изображения точно меняются между отображаемой формой (16:9) и фактическим изображением, поэтому PAR составляет 9:16. Обычно различия различаются, но этот пример преувеличен для ясности.

Из показанных параметров видно, что $PAR = 720: 1280 = 9:16 = 0,5625$

Пример соотношения сторон (SAR)

Как показывает пример *соотношения сторон изображения*, видеоролики представляют собой серию изображений, которые не обязательно имеют одинаковое соотношение сторон, так как конечный результат должен отображаться пользователю.

Итак, как вы получаете от этих растянутых изображений до нормально отображаемого вывода? Вам нужен коэффициент растяжения! Этот коэффициент растяжения применяется к изображению, чтобы довести его до правильного соотношения сторон для отображения. Это *соотношение сторон выборки* или SAR.



1280

Sample Aspect Ratio (SAR) 81:256



1280

720

Picture Aspect Ratio (PAR) 9:16

Фактор растяжения часто выражается как отношение двух целых чисел. Вы вычисляете его как $SAR = PAR / DAR$.

Из показанных параметров видно, что $SAR = 9:16 / 16:9 = (9/16) / (16/9) = 81/256 = 3.1604938271604938271604938271605$

Соотношение пикселей

Это другое название для *соотношения сторон образца*, и его следует избегать, поскольку естественный акроним (PAR) противоречит *соотношению сторон изображения*.

Прочитайте Соотношение сторон видео онлайн: <https://riptutorial.com/ru/video/topic/5713/соотношение-сторон-видео>

кредиты

S. No	Главы	Contributors
1	Начало работы с видео	Community , Sander
2	Понимание презентации СМИ	Sander
3	Соотношение сторон видео	Sander