

(19)



Евразийское
патентное
ведомство

(21) 201791565 (13) A1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ

(43) Дата публикации заявки
2017.12.29(22) Дата подачи заявки
2016.02.10(51) Int. Cl. H04N 19/70 (2014.01)
H04N 21/2343 (2011.01)
H04N 21/845 (2011.01)
H04N 21/854 (2011.01)

(54) СТРУКТУРА ЗАПИСИ ВЫБОРКИ И СИГНАЛИЗАЦИИ РАБОЧЕЙ ТОЧКИ В ФОРМАТЕ ФАЙЛА МНОГОСЛОЙНОГО ВИДЕО

(31) 62/115,075; 15/019,634

(32) 2015.02.11; 2016.02.09

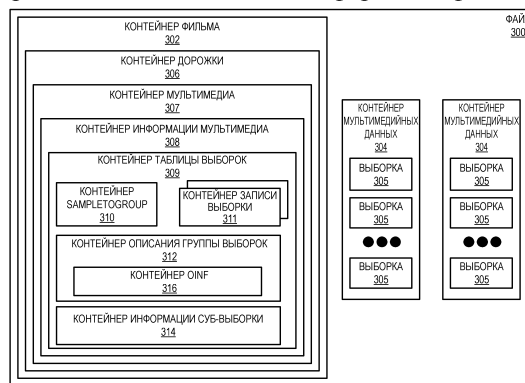
(33) US

(86) PCT/US2016/017280

(87) WO 2016/130635 2016.08.18

(71) Заявитель:
КВЭЛКОММ ИНКОРПОРЕЙТЕД
(US)(72) Изобретатель:
Хендри Фну, Ван Е-Куй (US)(74) Представитель:
Медведев В.Н. (RU)

(57) Видеоустройство для обработки многослойных видеоданных включает в себя носитель информации для хранения данных, выполненный с возможностью сохранения многослойных видеоданных, и один или более процессоров, выполненных с возможностью: получения многослойных видеоданных; сохранения многослойных видеоданных в формате файла; сохранения информации формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла и генерирования файла видеоданных, отформатированного в соответствии с форматом файла.



A1

201791565

201791565

A1

ОПИСАНИЕ ИЗОБРЕТЕНИЯ

2420-543803RU/052

СТРУКТУРА ЗАПИСИ ВЫБОРКИ И СИГНАЛИЗАЦИИ РАБОЧЕЙ ТОЧКИ В ФОРМАТЕ ФАЙЛА МНОГОСЛОЙНОГО ВИДЕО

[0001] По данной заявке испрашивается приоритет Предварительной Патентной Заявки США № 62/115,075, поданной 11 февраля 2015г., которая во всей своей полноте включена в настоящее описание посредством ссылки.

ОБЛАСТЬ ТЕХНИКИ, К КОТОРОЙ ОТНОСИТСЯ ИЗОБРЕТЕНИЕ

[0002] Данное раскрытие относится к кодированию видео.

ПРЕДПОСЫЛКИ СОЗДАНИЯ ИЗОБРЕТЕНИЯ

[0003] Возможности цифрового видео могут быть встроены в широкий диапазон устройств, включая цифровые телевизоры, системы цифрового непосредственного вещания, системы беспроводного вещания, персональные цифровые помощники (PDA), компьютеры класса лэптоп и настольные компьютеры, планшетные компьютеры, устройства для чтения электронных книг, цифровые камеры, цифровые записывающие устройства, цифровые мультимедийные проигрыватели, видеоигровые устройства, видеоигровые консоли, сотовые или спутниковые радиотелефоны, так называемые «интеллектуальные телефоны», устройства для организации видеоконференций, устройства потоковой передачи видео, и подобные. Устройства цифрового видео реализуют методики сжатия видео, такие как те, что описываются в стандартах, определяемых MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Часть 10, Усовершенствованное Кодирование Видео (AVC), стандарте Высокоэффективного Кодирования Видео (HEVC), который разрабатывается в настоящее время, и расширениях таких стандартов. Видеоустройства могут передавать, принимать, кодировать, декодировать, и/или сохранять цифровую видеoinформацию более эффективно посредством реализации таких методик сжатия видео.

[0004] Методики сжатия видео выполняют пространственное (внутри-картинки) предсказание и/или временное (внешнее для картинки) предсказание, чтобы сократить или удалить избыточность присущую видеопоследовательностям. Применительно к основанному

на блоке кодированию видео, слайс видео (т.е., кадр видео или фрагмент кадра видео) может быть разбит на блоки видео, которые также могут именоваться блоками дерева, единицами кодирования (CU) и/или узлами кодирования. Блоки видео во внутренне-кодированном (intra-coded) (I) слайсе картинки кодируются, используя пространственное предсказание по отношению к опорным выборкам в соседних блоках в той же самой картинке (изображении). Блоки видео во внешне-кодированном (inter-coded) (P или B) слайсе картинки могут использовать пространственное предсказание по отношению к опорным выборкам в соседних блоках в той же самой картинке или временное предсказание по отношению к опорным выборкам в других опорных картинках. Картинки могут именоваться кадрами, а опорные картинки могут именоваться опорными кадрами.

[0005] После того, как видеоданные были закодированы, видеоданные могут быть объединены в пакеты для передачи и сохранения. Видеоданные могут быть собраны в видеофайл, согласующийся с любым из многообразия стандартов, таким как базовый формат мультимедийного файла Международной Организации по Стандартизации (ISO) и его расширений, таких как AVC.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

[0006] В целом, данное раскрытие относится к сохранению видеоконтента в файле. В некоторых примерах, методики раскрытия основаны на базовом формате мультимедийного файла Международной Организации по Стандартизации (ISO) (ISOBMFF). Некоторые примеры данного раскрытия относятся к способам для сохранения видеопотоков, содержащих несколько кодированных слоев, где каждый слой может быть масштабируемым слоем, видом текстуры, видом глубины, и т.д., и способы могут применяться к сохранению видеоданных Многовидового Высокоэффективного Кодирования Видео (MV-HEVC), Масштабируемого HEVC (SHVC), 3-мерному HEVC (3D-HEVC), и другим типам видеоданных.

[0007] В одном примере, способ обработки многослойных видеоданных включает в себя этапы, на которых: получают многослойные видеоданные; сохраняют многослойные видеоданные в формате файла; сохраняют информацию формата представления для

каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и генерируют файл видеоданных, отформатированный в соответствии с форматом файла.

[0008] В другом примере, способ для обработки многослойных видеоданных включает в себя этапы, на которых: получают файл многослойных видеоданных, отформатированный в соответствии с форматом файла; применительно к формату файла, определяют информацию формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и декодируют многослойные видеоданные на основании определенной информации формата представления.

[0009] В другом примере, видеоустройство для обработки многослойных видеоданных, включает в себя: носитель информации для хранения данных, выполненный с возможностью сохранения многослойных видеоданных, и один или более процессоров, выполненных с возможностью: получения многослойных видеоданных; сохранения многослойных видеоданных в формате файла; сохранения информации формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и генерирования файла видеоданных, отформатированного в соответствии с форматом файла.

[0010] В другом примере, видеоустройство для обработки многослойных видеоданных включает в себя: носитель информации для хранения данных, выполненный с возможностью сохранения многослойных видеоданных, и один или более процессоров, выполненных с возможностью: получения файла многослойных видеоданных, отформатированного в соответствии с форматом файла; применительно к формату файла, определения информации формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и декодирования многослойных видеоданных на основании определенной информации формата представления.

[0011] В другом примере, видеоустройство для обработки многослойных видеоданных включает в себя: средство для получения многослойных видеоданных; средство для сохранения многослойных

видеоданных в формате файла; средство для сохранения информации формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и средство для генерирования файла видеоданных, отформатированного в соответствии с форматом файла.

[0012] В другом примере, машиночитаемый запоминающий носитель информации хранит инструкции, которые, когда исполняются, предписывают одному или более процессорам: получать многослойные видеоданные; сохранять многослойные видеоданные в формате файла; сохранять информацию формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и генерировать файл видеоданных, отформатированный в соответствии с форматом файла.

[0013] Подробности одного или более примеров раскрытия излагаются на сопроводительных чертежах и в описании ниже. Другие признаки, цели, и преимущества будут очевидны из описания, чертежей, и формулы изобретения.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

[0014] Фиг. 1 является структурной схемой, иллюстрирующей примерную систему кодирования и декодирования видео, которая может использовать методики, описанные в данном раскрытии.

[0015] Фиг. 2 является структурной схемой, иллюстрирующей примерный кодер видео, который может реализовывать методики, описанные в данном раскрытии.

[0016] Фиг. 3 является структурной схемой, иллюстрирующей примерный декодер видео, который может реализовывать методики, описанные в данном раскрытии.

[0017] Фиг. 4 является структурной схемой, иллюстрирующей примерный набор устройств, которые формируют часть сети.

[0018] Фиг. 5A является концептуальной схемой, иллюстрирующей примерную структуру файла в соответствии с одной или более методик данного раскрытия.

[0019] Фиг. 5B является концептуальной схемой, иллюстрирующей примерную структуру файла в соответствии с одной или более методик данного раскрытия.

[0020] Фиг. 6 является концептуальной схемой, иллюстрирующей примерную структуру файла в соответствии с одной или более методик данного раскрытия.

[0021] Фиг. 7 является блок-схемой, иллюстрирующей примерное функционирование устройства генерирования файла в соответствии с одной или более методик данного раскрытия.

[0022] Фиг. 8 является блок-схемой, иллюстрирующей примерное функционирование устройства чтения файла в соответствии с одной или более методик данного раскрытия.

ПОДРОБНОЕ ОПИСАНИЕ

[0023] Базовый формат мультимедийного файла ISO (ISOBMFF) является форматом файла для сохранения мультимедийных данных. ISOBMFF является расширяемым, чтобы поддерживать сохранение видеоданных, согласующихся с конкретными стандартами кодирования видео. Например, ISOBMFF ранее был расширен, чтобы поддерживать сохранение видеоданных, согласующихся со стандартами кодирования видео H.264/AVC и Высокоэффективного Кодирования Видео (HEVC). Кроме того, ISOBMFF ранее был расширен, чтобы поддерживать сохранение видеоданных, согласующихся с расширениями многовидового кодирования (MVC) и масштабируемого кодирования видео (SVC) у H.264/AVC. MV-HEVC, 3D-HEVC, и SHVC являются расширениями стандарта кодирования HEVC, которые поддерживают многослойные видеоданные. Свойства, добавленные в ISOBMFF для сохранения видеоданных, согласующихся с расширениями MVC и SVC у H.264/AVC, не являются достаточными для эффективного сохранения видеоданных, согласующихся с MV-HEVC, 3D-HEVC, и SHVC. Другими словами, разнообразные проблемы могут возникать, при попытке использования расширения ISOBMFF для сохранения видеоданных, согласующихся с расширениями MVC и SVC у H.264/AVC, для сохранения видеоданных, согласующихся с MV-HEVC, 3D-HEVC, и SHVC.

[0024] Например, в отличие от битового потока, который согласуется с расширениями MVC и SVC у H.264/AVC, битовый поток, который согласуется с MV-HEVC, 3D-HEVC, и SHVC, может включать в себя единицы доступа, которые содержат картинки Внутренней Точки Произвольного Доступ (IRAP) (далее IRAP картинка) и не-IRAP

картинки. Единица доступа, содержащая IRAP картинки и не-IRAP картинки может быть использована для произвольного доступа в MV-HEVC, 3D-HEVC, и SHVC. Тем не менее, ISOBMFF и существующие его расширения не предоставляют способа идентификации таких единиц доступа. Это может помешать возможности вычислительного устройства выполнять произвольный доступ, переключение слоев, и другие функции, ассоциированные с многослойными видеоданными.

[0025] Несмотря на то, что значительная часть описания методик данного раскрытия описывает MV-HEVC, 3D-HEVC, и SHVC, читателю следует понимать, что методики данного раскрытия могут быть применены к другим стандартам кодирования видео и/или их расширениям.

[0026] Как будет объяснено более подробно ниже, файл, согласующийся с форматом файла HEVC, может включать в себя ряд объектов, именуемых контейнерами. Контейнер может быть объектно-ориентированным строительным блоком, который определяется уникальным идентификатором типа и длиной. Данное раскрытие описывает методики, которые относятся к генерированию файлов, в соответствии с форматом файла, и, в частности, описывают методики для расположения определенных типов информации в определенных контейнерах, чтобы потенциально улучшать возможность устройства воспроизведения по обработке файлов, которые включают в себя несколько рабочих точек.

[0027] Несмотря на то, что значительная часть описания методик данного раскрытия описывает MV-HEVC, 3D-HEVC, и SHVC, читателю следует понимать, что методики данного раскрытия могут быть применены к другим стандартам кодирования видео и/или их расширениям.

[0028] Фиг. 1 является структурной схемой, иллюстрирующей пример системы 10 кодирования и декодирования видео, которая может использовать методики, описанные в данном раскрытии. Как показано на Фиг. 1, система 10 включает в себя устройство-источник 12, которое генерирует закодированные видеоданные, которые должны быть декодированы позже устройством-получателем 14. Устройство-источник 12 и устройство-получатель 14 могут быть выполнены в виде широкого диапазона устройств, включая

настольные компьютеры, компьютеры класса ноутбук (т.е., лэптоп), планшетные компьютеры, телевизионные абонентские приставки, телефонные трубки, такие как так называемые «интеллектуальные» телефоны, так называемые «интеллектуальные» панели, телевизоры, камеры, устройства отображения, цифровые мультимедийные проигрыватели, видеоигровые консоли, устройство потоковой передачи видео, или подобное. В некоторых случаях устройство-источник 12 и устройство-получатель 14 могут быть оборудованы для беспроводной связи. Устройство-источник 12 и устройство-получатель 14 могут рассматриваться в качестве видеоустройств.

[0029] В примере на Фиг. 1, устройство-источник 12 включает в себя источник 18 видео, кодер 20 видео и интерфейс 22 вывода. В некоторых случаях, интерфейс 22 вывода может включать в себя модулятор/демодулятор (модем) и/или приемопередатчик. В устройстве-источнике 12, источник 18 видео может включать в себя источник, такой как устройство захвата видео, например, видеокамеру, видеоархив, содержащий ранее захваченное видео, интерфейс видеоканала, чтобы принимать видео от поставщика видеоконтента, и/или систему компьютерной графики для генерирования данных компьютерной графики в качестве исходного видео, или сочетание таких источников. Тем не менее, методики, описываемые в данном раскрытии, могут быть применены к кодированию видео в целом, и могут быть применены к беспроводным и/или проводным приложениям.

[0030] Кодер 20 видео может кодировать захваченное, предварительно захваченное, или сгенерированное компьютером видео. Устройство-источник 12 может передавать закодированные видеоданные непосредственно устройству-получателю 14 через интерфейс 22 вывода устройства-источника 12. Закодированные видеоданные также могут быть (или альтернативно) сохранены на запоминающем устройстве 33 для осуществления доступа в дальнейшем посредством устройства-получателя 14 или других устройств, для декодирования и/или воспроизведения.

[0031] Устройство-получатель 14 включает в себя интерфейс 28 ввода, декодер 30 видео, и устройство 32 отображения. В некоторых случаях, интерфейс 28 ввода может включать в себя

приемник и/или модем. Интерфейс 28 ввода устройства-получателя 14 принимает закодированные видеоданные через линию 16 связи. Закодированные видеоданные, которые сообщаются через линию 16 связи, или предоставляются на запоминающем устройстве 33, могут включать в себя многообразие элементов синтаксиса, сгенерированных кодером 20 видео для использования декодером видео, таким как декодер 30 видео, при декодировании видеоданных. Такие элементы синтаксиса могут быть включены с закодированными видеоданными, которые передаются по средству связи, сохраняются на запоминающем носителе информации, или сохраняются на файловом сервере.

[0032] Устройство 32 отображения может быть интегрировано с, или может быть внешним для, устройством-получателем 14. В некоторых примерах, устройство-получатель 14 может включать в себя интегрированное устройство отображения и также может быть выполнено с возможностью создания интерфейса с внешним устройством отображения. В других примерах, устройство-получатель 14 может быть устройством отображения. В целом, устройство 32 отображения отображает декодированные видеоданные пользователю, и может быть выполнено в виде многообразия устройств отображения, таких как жидкокристаллический дисплей (LCD), плазменный дисплей, дисплей на органических светоизлучающих диодах (OLED), или другой тип устройства отображения.

[0033] Кодер 20 видео и декодер 30 видео каждый может быть реализован в качестве любой из многообразия пригодной схемы кодера, такой как один или более микропроцессоров, цифровые сигнальные процессоры (DSP), проблемно-ориентированные интегральные микросхемы (ASIC), программируемые вентильные матрицы (FPGA), дискретная логика, программное обеспечение, аппаратное обеспечение, встроенное программное обеспечение или любые их сочетания. Когда методики реализуются частично в программном обеспечении, устройство может хранить инструкции для программного обеспечения на пригодном, не временном машиночитаемом носителе информации и исполнять инструкции в аппаратном обеспечении, используя один или более процессоров,

чтобы выполнять методики данного раскрытия. Каждый из кодера 20 видео и декодера 30 видео может быть включен в один или более кодеров или декодеров, любой из которых может быть интегрирован как часть объединенного кодера/декодера (КОДЕКА) в соответствующем устройстве.

[0034] Устройство-получатель 14 может принимать закодированные видеоданные, которые должны быть декодированы, через линию 16 связи. Линия 16 связи может быть выполнена в виде любого типа средства или устройства, выполненного с возможностью перемещения закодированных видеоданных от устройства-источника 12 к устройству-получателю 14. В одном примере, линия 16 связи может содержать средство связи, чтобы позволить устройству-источнику 12 передавать закодированные видеоданные непосредственно устройству-получателю 14 в режиме реального времени. Закодированные видеоданные могут быть модулированы в соответствии со стандартом связи, таким как протокол беспроводной связи, и переданы устройству-получателю 14. Средство связи может быть выполнено в виде любого средства беспроводной или проводной связи, такого как радиочастотный (RF) спектр или одна или более физических линий передачи. Средство связи может формировать часть пакетной сети, такой как локальная сеть, сеть широкого охвата, или глобальная сеть, такая как Интернет. Средство связи может включать в себя маршрутизаторы, коммутаторы, базовые станции, или любое другое оборудование, которое может быть полезным, чтобы способствовать осуществлению связи от устройства-источника 12 к устройству-получателю 14.

[0035] В качестве альтернативы, интерфейс 22 вывода может выводить закодированные данные на запоминающее устройство 33. Сходным образом, интерфейс 28 ввода может осуществлять доступ к закодированным данным запоминающего устройства 33. Запоминающее устройство 33 может включать в себя любой из многообразия носителей информации для хранения данных с распределенным или локальным доступом, такой как накопитель на жестком диске, диски Blu-ray, DVD, CD-ROM, флэш-память, энергозависимую или энергонезависимую память, или любые другие пригодные цифровые запоминающие носители информации для хранения закодированных

видеоданных. В дополнительном примере, запоминающее устройство 33 может соответствовать файловому серверу или другому промежуточному запоминающему устройству, которое может удерживать закодированное видео, сгенерированное устройством-источником 12. Устройство-получатель 14 может осуществлять доступ к сохраненным видеоданным от запоминающего устройства 33 через потоковую передачу или загрузку. Файловый сервер может быть сервером любого типа, выполненным с возможностью сохранения закодированных видеоданных и передачи тех закодированных видеоданных устройству-получателю 14. Примерные файловые серверы включают в себя web-сервер (например, для web-сайта), FTP сервер, устройства подключаемого к сети накопителя (NAS), или локальный накопитель на диске. Устройство-получатель 14 может осуществлять доступ к закодированным видеоданным посредством любого стандартного соединения для передачи данных, включая Интернет соединение. Это может включать в себя беспроводной канал (например, Wi-Fi соединение), проводное соединение (например, DSL, кабельный модем, и т.д.), или сочетание двух видов, которое является пригодным для осуществления доступа к закодированным видеоданным, хранящимся на файловом сервере. Передача закодированных видеоданных от запоминающего устройства 33 может быть потоковой передачей, передачей загрузки, или сочетанием двух видов.

[0036] Методики данного раскрытия не обязательно ограничиваются беспроводными приложениями или установками. Методики могут быть применены к кодированию видео в поддержку любых из многообразия мультимедийных приложений, таких как эфирное телевизионное вещание, передачи кабельного телевидения, передачи спутникового телевидения, передачи потокового видео, например, через Интернет, кодирование цифрового видео для сохранения на носителе информации для хранения данных, декодирование цифрового видео, хранящегося на носителе информации для хранения данных, или других приложений. В некоторых примерах, система 10 может быть выполнена с возможностью обеспечения односторонней или двусторонней передачи видео, чтобы поддерживать приложения, такие как потоковая

передача видео, воспроизведение видео, вещание видео, и/или видеотелефония.

[0037] Кроме того, в примере на Фиг. 1, система 10 кодирования видео может включать в себя устройство 34 генерирования файла. Устройство 34 генерирования файла может принимать закодированные видеоданные, сгенерированные устройством-источником 12, и генерировать файл, который включает в себя закодированные видеоданные. Устройство-получатель 14 может принимать, либо напрямую, либо через запоминающее устройство 33, файл, сгенерированный устройством 34 генерирования файла. В разнообразных примерах, устройство 34 генерирования файла может включать в себя разнообразные типы вычислительных устройств. Например, устройство 34 генерирования файла может быть выполнено в виде Ориентированного на Мультимедиа Сетевого Элемента (MANE), серверного вычислительного устройства, персонального вычислительного устройства, вычислительного устройства особого назначения, коммерческого вычислительного устройства, или другого типа вычислительного устройства. В некоторых примерах, устройство 34 генерирования файла является частью сети доставки контента. Устройство 34 генерирования файла может принимать закодированные видеоданные от устройства-источника 12 через канал, такой как линия 16 связи. Кроме того, устройство-получатель 14 может принимать файл от устройства 34 генерирования файла через канал, такой как линия 16 связи.

[0038] В некоторых конфигурациях, устройство 34 генерирования файла может быть отдельным видеоустройством от устройства-источника 12 и устройства-получателя 14, тогда как в других конфигурациях, устройство 34 генерирования файла может быть реализовано в качестве компонента устройства-источника 12 или устройства-получателя 14. В реализациях, где устройство 34 генерирования файла является компонентом устройства-источника 12 или устройства-получателя 14, тогда устройство 34 генерирования файла может совместно использовать одни и те же ресурсы, такие как памяти, процессоры, и другое программное обеспечение, используемое кодером 20 видео и декодером 30 видео. В

реализациях, где устройство 34 генерирования файла является отдельным устройством, тогда устройство генерирования файла может включать в себя свою собственную память, процессоры, и другие единицы аппаратного обеспечения.

[0039] В других примерах, устройство-источник 12 или другое вычислительное устройство может генерировать файл, который включает в себя закодированные видеоданные. Тем не менее, для простоты объяснения, данное раскрытие описывает устройство 34 генерирования файла как генерирующее файл. Однако следует понимать, что такие описания применимы к вычислительным устройствам в целом.

[0040] Кодер 20 видео и декодер 30 видео могут функционировать в соответствии со стандартом сжатия видео, таким как стандарт Высокоэффективного Кодирования Видео (HEVC) или его расширение. Стандарт HEVC также может именоваться ISO/IEC 23008-2. Недавно, разработка HEVC была завершена Объединенной Совместной Командой по Кодированию Видео (JCT-VC) из Экспертной Группы по Кодированию Видео ITU-T (VCEG) и Экспертной Группы по Кинематографии (MPEG). Последний проект технического описания HEVC, и именуемый далее HEVC WD, доступен по адресу http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip. Многовидовое расширение HEVC, а именно MV-HEVC, также было разработано JCT-3V. Последний Рабочий Проект (WD) MV-HEVC, озаглавленный «MV-HEVC Draft Text 5» и именуемый далее MV-HEVC WD5, доступен по адресу http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1004-v6.zip. Масштабируемое расширение HEVC, а именно SHVC, также разрабатывается JCT-VC. Последний Рабочий Проект (WD) SHVC, озаглавленный «High efficiency video coding (HEVC) scalable extension draft 3» и именуемый далее SHVC WD3, доступен по адресу http://phenix.it-sudparis.eu/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1008-v3.zip. Последний рабочий проект (WD) расширения диапазона HEVC, доступен по адресу

v3.zip. Последний рабочий проект (WD) 3D расширения HEVC, а именно 3D-HEVC, озаглавленный «3D-HEVC Draft Text 1» доступен по адресу http://phenix.int-evry.fr/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip. Кодер 20 видео и декодер 30 видео могут функционировать в соответствии с одним из этих стандартов.

[0041] В качестве альтернативы, кодер 20 видео и декодер 30 видео могут функционировать в соответствии с другими собственными или промышленными стандартами, такими как стандарт ITU-T H.264, альтернативно именуемый MPEG-4, Часть 10, Расширенное Кодирование Видео (AVC), или расширения таких стандартов. Методики данного раскрытия, тем не менее, не ограничиваются каким-либо конкретным стандартом кодирования. Другие примеры стандартов сжатия видео включают в себя ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 или ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual и ITU-T H.264 (также известный как ISO/IEC MPEG-4 AVC), включая его расширения Масштабируемое Кодирование Видео (SVC) и Многовидовое Кодирование Видео (MVC).

[0042] Несмотря на то, что не показано на Фиг. 1, в некоторых аспектах, кодер 20 видео и декодер 30 видео каждый может быть интегрирован с кодером и декодером аудио, чтобы обрабатывать кодирование как аудио, так и видео в общем потоке данных или в отдельных потоках данных. Если применимо, в некоторых примерах, блоки MUX-DEMUX могут быть согласованы с протоколом мультиплексора ITU H.223, или другими протоколами, такими как протокол пользовательских дейтаграмм (UDP).

[0043] JCT-VC разрабатывается стандарт HEVC. Усилия по стандартизации основаны на развивающейся модели устройства кодирования видео, именуемой Тестовая Модель HEVC (НМ). НМ предполагает некоторое количество дополнительных возможностей устройств кодирования видео относительно существующих устройств в соответствии с, например, ITU-T H.264/AVC. Например, тогда как H.264/AVC предоставляет девять режимов кодирования с внутренним предсказанием, НМ может обеспечивать вплоть до тридцати трех режимов кодирования с внутренним предсказанием.

[0044] В целом, рабочая модель НМ описывает, что кадр или картинка видео могут быть разделены на последовательность блоков дерева или Наибольшие Единицы Кодирования (LCU), которые включают в себя выборки как яркости, так и цветности. Блоки дерева также могут именоваться Единицами Дерева Кодирования (CTU). Блок дерева имеет цель сходную с макроблоком в стандарте H.264/AVC. Слайс включает в себя некоторое количество последовательных блоков дерева в очередности кодирования. Кадр или картинка видео могут быть разбиты на один или более слайсов. Каждый блок дерева может быть разбит на единицы кодирования (CU) в соответствии с квадродеревом. Например, блок дерева, как корневой узел квадродерева, может быть разбит на четыре узла-потомка, и каждый узел-потомок в свою очередь может быть узлом-родителем и разбит на другие четыре узла-потомка. Итоговый, неразбитый узел-потомок, как концевой узел квадродера, содержит узел кодирования, т.е., кодированный блок видео. Данные синтаксиса, ассоциированные с кодированным битовым потоком, могут определять максимальное количество раз, которое может быть разбит блок дерева, и также могут определять минимальный размер узлов кодирования.

[0045] CU включает в себя узел кодирования и единицы предсказания (PU) и единицы преобразования (TU), ассоциированные с узлом кодирования. Размер CU соответствует размеру узла кодирования и должен быть квадратным по форме. Размер CU может находиться в диапазоне от 8×8 пикселей вплоть до размера блока дерева с максимум 64×64 пикселей или больше. Каждая CU может содержать одну или более PU и одну или более TU. Данные синтаксиса, ассоциированные с CU, могут описывать, например, разбиение CU на одну или более PU. Режимы разбиения могут отличаться в зависимости от того, является ли CU кодируемой в режиме пропуска или непосредственном режиме, кодируемой в режиме внутреннего предсказания, или кодируемой в режиме внешнего предсказания. PU могут быть разбиты так, чтобы быть неквадратными по форме. Данные синтаксиса, ассоциированные с CU, также могут описывать, например, разбиение CU на одну или более

TU в соответствии с квадродеревом. TU может быть квадратной или неквадратной по форме.

[0046] Стандарт HEVC обеспечивает преобразования в соответствии с TU, которые могут быть разными для разных CU. Размер TU, как правило, определяется на основании размера PU внутри заданной CU, которая определена для разбитой LCU, несмотря на то, что это не всегда может быть так. TU, как правило, имеют размер точно такой же или меньше чем у PU. В некоторых примерах, остаточные выборки, соответствующие CU, могут быть подразделены на более мелкие единицы, используя структуру квадродерева, известную как «остаточное квадродерево» (RQT). Концевые узлы RQT могут именоваться TU. Значения пиксельной разности, ассоциированные с TU, могут быть преобразованы, чтобы создавать коэффициенты преобразования, которые могут быть квантованы.

[0047] В целом, PU включает в себя данные, которые относятся к процессу предсказания. Например, когда PU является кодируемой во внутреннем режиме, PU может включать в себя данные, описывающие режим внутреннего предсказания для PU. В качестве другого примера, когда PU является кодируемой во внешнем режиме, PU может включать в себя данные, определяющие вектор движения для PU. Данные, определяющие вектор движения для PU, могут описывать, например, горизонтальную составляющую вектора движения, вертикальную составляющую вектора движения, разрешение для вектора движения (например, точность в одну четвертую пикселя или точность в одну восьмую пикселя), опорную картинку на которую указывает вектор движения, и/или список опорных картинок (т.е., Список 0, Список 1, или Список C) для вектора движения.

[0048] В целом, TU используется для процессов преобразования и квантования. Заданная CU с одной или более PU также может включать в себя одну или более единиц преобразования (TU). Вслед за предсказанием, кодер 20 видео может вычислять остаточные значения, соответствующие PU. Остаточные значения содержат значения пиксельной разности, которые могут быть преобразованы в коэффициенты преобразования, квантованы, и

просканированы, используя TU, чтобы создавать преобразованные в последовательную форму коэффициенты преобразования для энтропийного кодирования. Данное раскрытие, как правило, использует понятие «блок видео», чтобы ссылаться на узел кодирования (т.е., блок кодирования) у CU. В некоторых особых случаях, данное раскрытие также может использовать понятие «блок видео», чтобы ссылаться на блок дерева, т.е., LCU, или CU, которая включает в себя узел кодирования и PU и TU.

[0049] Видеопоследовательность, как правило, включает в себя ряд кадров или картинок видео. Группа картинок (GOP) обычно содержит ряд из одной или более картинок видео. GOP может включать в себя данные синтаксиса в заголовке GOP, заголовке одной или более картинок, или где-либо еще, которые описывают количество картинок, включенных в GOP. Каждый слайс картинки может включать в себя данные синтаксиса слайса, которые описывают режим кодирования для соответствующего слайса. Кодер 20 видео, как правило, оперирует блоками видео в индивидуальных слайсах видео для того, чтобы кодировать видеоданные. Блок видео может соответствовать узлу кодирования внутри CU. Блоки видео могут иметь фиксированные или варьирующиеся размеры, и могут отличаться по размеру в соответствии с указанным стандартом кодирования.

[0050] В качестве примера, HM поддерживает предсказание в разнообразных размерах PU. Предполагая, что размер конкретной CU составляет $2N \times 2N$, HM поддерживает внутреннее предсказание в размерах PU вида $2N \times 2N$ или $N \times N$, и внешнее предсказание в симметричных размерах PU вида $2N \times 2N$, $2N \times N$, $N \times 2N$, или $N \times N$. HM также поддерживает ассиметричное разбиение для внешнего предсказания в размерах PU вида $2N \times nU$, $2N \times nD$, $nL \times 2N$, и $nR \times 2N$. В несимметричном разбиении, одно направление CU не разбивается, тогда как другое направление разбивается на 25% и 75%. Фрагмент CU, соответствующий 25% разделу указывается «n», за которым следует указание «Верх», «Низ», «Лево», или «Право». Таким образом, например, « $2N \times nU$ » относится к $2N \times 2N$ CU, которая разбивается горизонтально с $2N \times 0.5N$ PU вверху и $2N \times 1.5N$ PU внизу.

[0051] В данном раскрытии, «N×N» и «N на N» может быть использовано взаимозаменяемо, чтобы сослаться на размеры в пикселях блока видео в единицах вертикального и горизонтального измерений, например, 16×16 пикселей или 16 на 16 пикселей. В целом, 16×16 блок имеет 16 пикселей в вертикальном направлении ($y=16$) и 16 пикселей в горизонтальном направлении ($x=16$). Подобным образом, N×N блок в целом имеет N пикселей в вертикальном направлении и N пикселей в горизонтальном направлении, где N представляет собой неотрицательное целочисленное значение. Пиксели в блоке могут быть организованы в строках и столбцах. Более того, блоки не обязательно имеют количество пикселей в горизонтальном направлении точно такое же как в вертикальном направлении. Например, блоки могут содержать N×M пикселей, где M не обязательно равно N.

[0052] Вслед за кодированием с внутренним предсказанием или внешним предсказанием, используя PU у CU, кодер 20 видео может вычислять остаточные данные для TU у CU. PU могут содержать данные пикселя в пространственной области (также именуемой область пикселя), а TU могут содержать коэффициенты в области преобразования в след за применением преобразования, например, дискретного косинусного преобразования (DCT), целочисленного преобразования, вейвлет преобразования, или концептуально сходного преобразования к остаточным видеоданным. Остаточные данные могут соответствовать пиксельным разностям между пикселями незакодированной картинки и значениями предсказания, соответствующими PU. Кодер 20 видео может формировать TU, включающие в себя остаточные данные для CU, и затем преобразовывать TU, чтобы создавать коэффициенты преобразования для CU.

[0053] Вслед за любыми преобразованиями, чтобы создать коэффициенты преобразования, кодер 20 видео может выполнять квантование коэффициентов преобразования. Квантование, как правило, относится к процессу, при котором коэффициенты преобразования квантуются чтобы возможно сократить объем данных, используемых для представления коэффициентов, обеспечивая

дальнейшее сжатие. Процесс квантования может сокращать битовую глубину, ассоциированную с некоторыми или всеми из коэффициентов. Например, n -битное значение может быть округлено до ближайшего меньшего m -битного значения во время квантования, где n больше m .

[0054] В некоторых примерах, кодер 20 видео может использовать предварительно определенную очередность сканирования, чтобы сканировать квантованные коэффициенты преобразования, чтобы создавать преобразованный в последовательную форму вектор, который может быть энтропийно закодирован. В других примерах, кодер 20 видео может выполнять адаптивное сканирование. После сканирования квантованных коэффициентов преобразования, чтобы сформировать одномерный вектор, кодер 20 видео может энтропийно кодировать одномерный вектор, например, в соответствии с контекстно-зависимым адаптивным кодированием с переменной длиной кодового слова (CAVLC), контекстно-зависимым адаптивным бинарным арифметическим кодированием (CABAC), основанным на синтаксисе контекстно-зависимым адаптивным бинарным арифметическим кодированием (SBAC), энтропийным кодированием с разбиением на интервала вероятности (PIPE) или другой методологией энтропийного кодирования. Кодер 20 видео также может энтропийно кодировать элементы синтаксиса, ассоциированные с закодированными видеоданными для использования декодером 30 видео при декодировании видеоданных.

[0055] Чтобы выполнять CABAC, кодер 20 видео может назначать контекст в рамках модели контекста символу, который должен быть передан. Контекст может относиться, например, к тому, являются ли соседние значения символа ненулевыми или нет. Чтобы выполнить CAVLC, кодер 20 видео может выбирать код переменной длины для символа, который должен быть передан. Кодовые слова в кодировании с переменной длиной (VLC) могут быть сконструированы таким образом, что относительно более короткие коды соответствуют более вероятным символам, тогда как более длинные коды соответствуют менее вероятным символам. Таким образом, использование VLC может достигать экономии битов в

сравнении с, например, использованием кодовых слов равной длины для каждого символа, который должен быть передан. Определение вероятности может быть основано на контексте, назначенном символу.

[0056] Кодер 20 видео может выводить битовый поток, который включает в себя последовательность битов, которые формируют представление кодированных картинок и ассоциированных данных. Понятие «битовый поток» может быть собирательным понятием, которое используется, чтобы ссылаться на либо поток единиц Слоя Сетевой Абстракции (NAL) (например, последовательность единиц NAL), либо байтовый поток (например, инкапсуляция потока единиц NAL, содержащая префиксы начального кода и единицы NAL, как указано в Приложении В стандарта HEVC). Единица NAL является структурой синтаксиса, содержащей указание типа данных в единице NAL и байты, содержащие данные в форме полезной нагрузки необработанной байтовой последовательности (RBSP) чередующейся при необходимости с битами предотвращения эмуляции. Каждая из единиц NAL может включать в себя заголовок единицы NAL и может инкапсулировать RBSP. Заголовок единицы NAL может включать в себя элемент синтаксиса, который указывает код типа единицы NAL. Код типа единицы NAL, указываемый заголовком единицы NAL у единицы NAL, указывает тип единицы NAL. RBSP может быть структурой синтаксиса, содержащей целое число байтов, которые инкапсулированы внутри единицы NAL. В некоторых примерах, RBSP включает в себя нулевые биты.

[0057] Разные типы единиц NAL могут инкапсулировать разные типы RBSP. Например, первый тип единицы NAL может инкапсулировать RBSP для PPS, второй типа единицы NAL может инкапсулировать RBSP для сегмента слайса, третий тип единицы NAL может инкапсулировать RBSP для SEI, и т.п. Единицы NAL, которые инкапсулируют RBSP для данных кодирования видео (в противоположность RBSP для наборов параметров и сообщений SEI) могут именоваться единицами NAL слоя кодирования видео (VLC). Единицы NAL, которые содержат наборы параметров (например, VPS, SPS, PPS, и т.д.) могут именоваться единицами NAL набора параметров.

[0058] Данное раскрытие может относиться к единице NAL, которая инкапсулирует RBSP для сегмента слайса, как к единице NAL кодированного слайса. Как определено в HEVC WD, сегмент слайса является целочисленным количеством CTU, упорядоченных последовательно в сканировании тайла и которые содержатся в одной единице NAL. В противоположность, в HEVC WD, слайс может быть целочисленным количеством CTU, которые содержатся в одном независимом сегменте слайса и всех последующих зависимых сегментах слайса (если есть), которые предшествуют следующему независимому сегменту слайса (если есть) внутри одной и той же единицы доступа. Независимый сегмент слайса является сегментом слайса, для которого значения элементов синтаксиса заголовка сегмента слайса не вытекают из значений для предшествующего сегмента слайса. Зависимый сегмент слайса является сегментом слайса, для которого значения некоторых элементов синтаксиса заголовка сегмента слайса вытекают из значений для предшествующего независимого сегмента слайса в очередности декодирования. RBSP единицы NAL кодированного слайса может включать в себя заголовок сегмента слайса и данные слайса. Заголовок сегмента слайса является частью кодированного сегмента слайса, содержащей элементы данных принадлежащие к первой или всем CTU, представленным в сегменте слайса. Заголовок слайса является заголовком сегмента слайса независимого сегмента слайса, т.е. текущего сегмента слайса или самого последнего независимого сегмента слайса, который предшествует текущему зависимому сегменту слайса в очередности декодирования.

[0059] VPS является структурой синтаксиса, содержащей элементы синтаксиса, которые применяются к нулю или более полным кодированным видеопоследовательностям (CVS). SPS является структурой синтаксиса, содержащей элементы синтаксиса, которые применяются к нулю или более полным CVS. SPS может включать в себя элемент синтаксиса, который идентифицирует VPS, которая является активной, когда активна SPS. Таким образом, элементы синтаксиса VPS могут быть более в целом применимы, чем элементы синтаксиса SPS.

[0060] Набор параметров (например, VPS, SPS, PPS, и т.д.)

могут содержать идентификацию, на которую ссылаются напрямую или косвенно из заголовка слайса у слайса. Процесс ссылки известен как «активация». Следовательно, когда декодер 30 видео декодирует конкретный слайс, набор параметров, на который ссылаются, напрямую или косвенно, посредством элемента синтаксиса в заголовке слайса, называется «активированным». В зависимости от типа набора параметров, активация может происходить на основе картинки или на основе последовательности. Например, заголовок слайса у слайса может включать в себя элемент синтаксиса, который идентифицирует PPS. Следовательно, когда кодер видео кодирует слайс, может быть активирован PPS. Кроме того, PPS может включать в себя элемент синтаксиса, который идентифицирует SPS. Следовательно, когда активируется PPS, который идентифицирует SPS, может быть активирован SPS. SPS может включать в себя элемент синтаксиса, который идентифицирует VPS. Следовательно, когда активируется SPS, который идентифицирует VPS, активируется VPS.

[0061] Декодер 30 видео может принимать битовый поток, сгенерированный кодером 20 видео. В дополнение, декодер 30 видео может осуществлять синтаксический анализ битового потока, чтобы получать элементы синтаксиса из битового потока. Декодер 30 видео может восстанавливать картинки видеоданных на основании, по меньшей мере, частично элементов синтаксиса, полученных из битового потока. Процесс восстановления видеоданных может быть в целом обратным процессу, который выполняется кодером 20 видео. Например, декодер 30 видео может использовать векторы движения PU, чтобы определять блоки предсказания для PU у текущей CU. В дополнение, декодер 30 видео может обратно квантовать блоки коэффициента TU у текущей CU. Декодер 30 видео может выполнять обратные преобразования над блоками коэффициента, чтобы восстанавливать блоки преобразования TU у текущей CU. Декодер 30 видео может восстанавливать блоки кодирования текущей CU посредством сложения выборок блоков предсказания для PU текущей CU с соответствующими выборками блоков преобразования TU текущей CU. Посредством восстановления блоков кодирования для каждой CU картинки, декодер 30 видео может восстанавливать картинку.

[0062] В HEVC WD, CVS может начинаться с картинки Мгновенного Обновления Декодирования (IDR), или картинки доступа с нерабочей ссылкой (BLA), или картинки чистого произвольного доступа (CRA), которая является первой картинкой в битовом потоке, включающем в себя все последующие картинки, которые не являются картинками IDR или BLA. Картинка IDR содержит только I слайсы (т.е., слайсы, в которых используется только внутреннее предсказание). Картинка IDR может быть первой картинкой в битовом потоке в очередности декодирования, или может появляться позже в битовом потоке. Каждая картинка IDR является первой картинкой CVS в очередности декодирования. В HEVC WD картинка IDR может быть картинкой внутренней точки произвольного доступа, для которой единица NAL VCL имеет `nal_unit_type` равный `IDR_W_RADL` или `IDR_N_LP`.

[0063] Картинки IDR могут быть использованы для произвольного доступа. Тем не менее, картинки, следующие за картинкой IDR в очередности декодирования, не могут использовать картинки, декодированные до картинки IDR в качестве ссылки. Соответственно, битовые потоки, которые опираются на картинки IDR для произвольного доступа могут обладать значительно более низкой эффективностью кодирования, чем битовые потоки, которые используют дополнительные типы картинок произвольного доступа. В, по меньшей мере, некоторых примерах, единица доступа IDR является единицей доступа, которая содержит картинку IDR.

[0064] Концепция картинок CRA была введена в HEVC, чтобы позволить картинкам, которые следуют за картинкой CRA в очередности декодирования, но предшествуют картинке CRA в очередности вывода, использовать картинки, декодированные до картинки CRA для ссылки. Картинки, которые следуют за картинкой CRA в очередности декодирования, но предшествуют картинке CRA в очередности вывода, именуются ведущими картинками, ассоциированными с картинкой CRA (или ведущими картинками картинки CRA). Т.е., чтобы улучшить эффективность кодирования, концепция картинок CRA была введена в HEVC, чтобы позволить картинкам, которые следуют за картинкой CRA в очередности декодирования, но предшествуют картинке CRA в очередности

вывода, использовать картинки, декодированные до картинки CRA для ссылки. Единица доступа CRA является единицей доступа, в которой кодированной картинкой является картинка CRA. В HEVC WD, картинка CRA является внутренней картинкой произвольного доступа для которой единица NAL VCL имеет `nal_unit_type` равный `CRA_NUT`.

[0065] Ведущие картинки у картинки CRA являются корректно декодируемыми если декодирование начинается с картинки IDR или картинки CRA, появляющейся перед картинкой CRA в очередности декодирования. Тем не менее, ведущие картинки у картинки CRA могут быть не декодируемыми, когда происходит произвольный доступ из картинки CRA. Следовательно, декодер видео, как правило, декодирует ведущие картинки у картинки CRA во время декодирования с произвольным доступом. Чтобы предотвратить распространение ошибки из опорных картинок, которые могут быть недоступны в зависимости от того, где начинается декодирование, ни одна картинка, которая следует за картинкой CRA как в очередности декодирования, так и очередности вывода, не может использовать любую картинку, которая предшествует картинке CRA либо в очередности декодирования, либо очередности вывода (что включает в себя ведущие картинки) для ссылки.

[0066] Концепция картинки BLA была введена а HEVC после введения картинок CRA и основана на концепции картинок CRA. Картинка BLA, как правило, происходит из сращивания битового потока в позиции картинки CRA, и в сращенном битовом потоке, точка сращивания картинки CRA меняется на картинку BLA. Следовательно, картинки BLA могут быть картинками CRA в исходных битовых потоках и картинка CRA меняется, чтобы становиться картинкой BLA, средством сращивания битового потока после сращивания битового потока в местоположении картинки CRA. В некоторых случаях, единица доступа, которая содержит картинку RAP, может именоваться в данном документе единицей доступа RAP. Единица доступа BLA является единицей доступа, которая содержит картинку BLA. В HEVC WD, картинка BLA может быть внутренней картинкой произвольного доступа, для которой каждая единица NAL VCL имеет `nal_unit_type` равный `BLA_W_LP`, `BLA_W_RADL`, или `BLA_N_LP`.

[0067] В целом, картинка IRAP содержит только I слайсы, и может быть картинкой BLA, картинкой CRA, или картинкой IDR. Например, HEVC WD указывает, что картинка IRAP может быть кодированной картинкой для которой единица NAL VCL имеет `nal_unit_type` в диапазоне от `BLA_W_LP` до `RSV_IRAP_VCL23`, включительно. Кроме того, HEVC WD указывает, что первая картинка в битовом потоке в очередности декодирования должна быть картинкой IRAP. Таблица 7-1 в HAVC WD показывает коды типа единицы NAL и классы типа единицы NAL. Таблица 7-1 HEVC WD воспроизводится ниже.

Таблица 7-1 - Коды типа единицы NAL и классы типа единицы NAL

<code>nal_unit_type</code>	Имя <code>nal_unit_type</code>	Содержимое единицы NAL и структура синтаксиса Rbsp	Класс типа единицы NAL
0 1	TRAIL_N TRAIL_R	Кодированный сегмент слайса у замыкающей картинки не-TSA, не-STSA <code>slice_segment_layer_rbsp()</code>	VCL
2 3	TSA_N TSA_R	Кодированный сегмент слайса у картинки TSA <code>slice_segment_layer_rbsp()</code>	VCL
4 5	STSA_N STSA_R	Кодированный сегмент слайса у картинки STSA <code>slice_segment_layer_rbsp()</code>	VCL
6 7	RADL_N RADL_R	Кодированный сегмент слайса у картинки RADL <code>slice_segment_layer_rbsp()</code>	VCL
8 9	RASL_N RASL_R	Кодированный сегмент слайса у картинки RASL <code>slice_segment_layer_rbsp()</code>	VCL
10 12 14	RSV_VCL_N10 RSV_VCL_N12 RSV_VCL_N14	Зарезервированные не-IRAP суб-слоя не-опорные типы единицы NAL VCL	VCL
11 13 15	RSV_VCL_N11 RSV_VCL_N13 RSV_VCL_N15	Зарезервированные не-IRAP суб-слоя не-опорные типы единицы NAL VCL	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_N_LP	Кодированный сегмент слайса картинки BLA <code>slice_segment_layer_rbsp()</code>	VCL
19	IDR_W_RADL	Кодированный сегмент слайса	VCL

20	IDR_N_LP	картинки IDR slice_segment_layer_rbsp()	
21	CRA_NUT	Кодированный сегмент слайса картинки CRA slice_segment_layer_rbsp()	VCL
22	RSV_IRAP_VCL22	Зарезервированные типы	VCL
23	RSV_IRAP_VCL23	единицы NAL VCL IRAP	
24...31	RSR_VCL24... RSR_VCL31	Зарезервированные типы единицы NAL VCL не-IRAP	VCL
32	VPS_NUT	Набор параметров видео video_parameter_set_rbsp()	не-VCL
33	SPS_NUT	Набор параметров последовательности seq_parameter_set_rbsp()	не-VCL
34	PPS_NUT	Набор параметров последовательности pic_parameter_set_rbsp()	не-VCL
35	AUD_NUT	Разделитель единицы доступа access_unit_delimiter_rbsp()	не-VCL
36	EOS_NUT	Конец последовательности end_of_seq_rbsp()	не-VCL
37	EOB_NUT	Конец битового потока end_of_bitstream_rbsp()	не-VCL
38	FD_NUT	Данные заполнителя filler_data_rbsp()	не-VCL
39	PREFIX_SEI_NUT	Добавочная информация	не-VCL
40	SUFFIX_SEI_NUT	улучшения sei_rbsp()	
41..47	RSV_NVCL41.. RSV_NVCL47	Зарезервировано	не-VCL
48..63	UNSPEC48.. UNSPEC63	Не оговорено	не-VCL

[0068] Одно отличие картинок BLA от картинок CRA состоит в следующем. Применительно к картинке CRA, ассоциированные ведущие картинки являются корректно декодируемыми, если декодирование начинается с картинки RAP до картинки CRA в очередности декодирования. Тем не менее, ведущие картинки, ассоциированные с картинкой CRA, могут не быть корректно декодируемыми, когда происходит произвольный доступ с картинки CRA (т.е., когда декодирование начинается с картинки CRA, или другими словами,

когда картинка CRA является первой картинкой в битовом потоке). В противоположность, может присутствовать сценарий, где ведущие картинки, ассоциированные с картинкой BLA являются декодируемыми, даже когда декодирование начинается с картинки RAP до картинки BLA в очередности декодирования.

[0069] Некоторые из ведущих картинок, ассоциированных с конкретной картинкой CRA или конкретной картинкой BLA, могут быть корректно декодируемыми даже когда конкретная картинка CRA или конкретная картинка BLA является первой картинкой в битовом потоке. Эти ведущие картинки могут именоваться декодируемыми ведущими картинками (DLP) или Декодируемыми Ведущими Произвольного Доступа (RADL) картинками. В HEVC WD, картинка RADL может быть кодированной картинкой для которой единица NAL VCL имеет `nal_unit_type` равный `RADL_R` или `RADL_N`. Кроме того, HEVC WD указывает, что все картинки RADL являются ведущими картинками и что картинки RADL не используются в качестве опорных картинок для процесса декодирования замыкающих картинок той же самой ассоциированной картинки IRAP. Когда присутствуют, все картинки RADL предшествуют, в очередности декодирования, всем замыкающим картинкам той же самой ассоциированной картинки IRAP. HEVC WD указывает, что единица доступа RADL может быть единицей доступа, в которой кодированная картинка является картинкой RADL. Замыкающая картинка может быть картинкой, которая следует за ассоциированной картинкой IRAP (т.е., предшествующая картинка IRAP в очередности декодирования) в очередности вывода.

[0070] Другие ведущие картинки могут именоваться не-декодируемыми ведущими картинками (NLP) или Пропускаемыми Ведущими Произвольного Доступа (RASL) картинками. В HEVC WD, картинка RASL может быть кодированной картинкой, для которой каждая единица NAL VCL имеет `nal_unit_type` равный `RASL_R` или `RASL_N`. Все картинки RASL являются ведущими картинками ассоциированной картинки BLA или CRA.

[0071] При условии, что необходимые наборы параметров являются доступными, когда они должны быть активированы, картинка IRAP и все последующие картинки не-RASL в очередности

декодирования могут быть корректно декодированы без выполнения процесса декодирования любых картинок, которые предшествуют картинке IRAP в очередности декодирования. В битовом потоке могут присутствовать картинки, которые содержат только I слайсы, которые не являются картинками IRAP.

[0072] В многовидовом кодировании, может присутствовать несколько видов одной и той же сцены с разных точек зрения. Понятие «единица доступа» может быть использовано, чтобы ссылаться на набор картинок, которые соответствуют одному и тому же экземпляру времени. Следовательно, видеоданные могут быть осмыслены как ряд единиц доступа, происходящих с течением времени. «Компонент вида» может быть кодированным представлением вида в одной единице доступа. В данном раскрытии «вид» может относиться к последовательности или набору компонентов вида, ассоциированных с одним и тем же идентификатором вида. Компонент вида может содержать компонент вида по текстуре и компонент вида по глубине. В данном раскрытии «вид» может относиться к набору или последовательности из одного или более компонентов вида, ассоциированных с одним и тем же идентификатором вида.

[0073] Компонент вида по текстуре (т.е., картинка текстуры) может быть кодированным представлением текстуры вида в одной единице доступа. Вид текстуры может быть последовательностью компонентов вида по текстуре, ассоциированных с идентичным значением индекса очередности вида. Индекс очередности вида у вида может указывать позицию камеры у вида по отношению к другим видам. Компонент вида по глубине (т.е., картинка глубины) может быть кодированным представлением глубины вида в одной единице доступа. Вид глубины может быть набором или последовательностью из одного или более компонентов вида по глубине, ассоциированных с идентичным значением индекса очередности вида.

[0074] В MV-HEVC, 3D-HEVC, и SHVC, кодер видео может генерировать битовый поток, который содержит ряд единиц NAL. Разные единицы NAL битового потока могут быть ассоциированы с разными слоями битового потока. Слой может быть определен как набор единиц NAL VCL и ассоциированных единиц NAL не-VCL, которые имеют один и тот же идентификатор слоя. Слой может быть

эквивалентен виду в многовидовом кодировании видео. В многовидовом кодировании видео, слой может содержать все компоненты вида одного и того же слоя с разными экземплярами времени. Каждый компонент вида может быть кодированной картинкой сцены видео, принадлежащей конкретному виду в конкретный экземпляр времени. В некоторых примерах 3D кодирования видео, слой может содержать либо все кодированные картинки глубины конкретного вида, либо кодированные картинки текстуры конкретного вида. В других примерах 3D кодирования видео, слой может содержать как компоненты вида по текстуре, так и компоненты вида по глубине конкретного вида. Сходным образом, в контексте масштабируемого кодирования видео, слой, как правило, соответствует кодированным картинкам с характеристиками видео отличными от кодированных картинок в других слоях. Такие характеристики видео, как правило, включают в себя пространственное разрешение и уровень качества (например, Отношение Сигнала-к-Шуму). В HEVC и его расширениях, временная масштабируемость может быть достигнута в рамках одного слоя посредством определения группы картинок с конкретным временным уровнем в качестве суб-слоя.

[0075] Применительно к каждому соответствующему слою битового потока, данные в более низком слое могут быть декодированы без ссылки на данные в любом более высоком слое. В масштабируемом кодировании видео, например, данные в базовом слое могут быть декодированы без ссылки на данные в слое улучшения. В целом, единицы NAL могут инкапсулировать только данные одного слоя. Следовательно, единицы NAL, инкапсулирующие данные наивысшего оставшегося слоя битового потока, могут быть удалены из битового потока не оказывая влияния на возможность декодирования данных в оставшихся слоях битового потока. В многовидовом кодировании и 3D-HEVC, более высокие слои могут включать в себя дополнительные компоненты вида. В SHVC, более высокие слои могут включать в себя данные улучшения отношения сигнала к шуму (SNR), пространственные данные улучшения, и/или временные данные улучшения. В MV-HEVC, 3D-HEVC и SHVC, слой может именоваться «базовым слоем», если декодер видео может

декодировать картинки в слое без ссылки на данные любого другого слоя. Базовый слой может соответствовать базовой спецификации HEVC (например, HEVC WD).

[0076] В SVC, слои отличные от базового слоя могут именоваться «слоями улучшения» и могут предоставлять информацию, которая улучшает визуальное качество видеоданных, декодированных из битового потока. SVC может улучшать пространственное разрешение, отношение сигнала-к-шуму (т.е., качество) или временную частоту. В масштабируемом кодировании видео (например, SHVC), «представление слоя» может быть закодированным представлением пространственного слоя в одной единице доступа. Для простоты объяснения, данное раскрытие относится к компонентам вида и/или представлениям слоя как «компонентам вида/представлениям слоя» или просто «картинкам».

[0077] Чтобы реализовать слои в HEVC, заголовки единиц NAL включают в себя элемент синтаксиса `nuh_layer_id`, который ранее упоминался как элемент синтаксиса `nuh_reserved_zero_6bits` в различных рабочих проектах, которые предшествовали итоговому стандарту HEVC. В базовом стандарте HEVC, элемент синтаксиса `nuh_layer_id` ограничивается значением 0. Тем не менее, в MV-HEVC, 3D-HEVC, и SVC, элемент синтаксиса `nuh_layer_id` может быть больше 0, чтобы указывать идентификатор слоя. Единицы NAL битового потока, которые имеют элементы синтаксиса `nuh_layer_id`, которые указывают разные значения, принадлежат к разным слоям битового потока.

[0078] В некоторых примерах, элемент синтаксиса `nuh_layer_id` единицы NAL равен 0, если единица NAL относится к базовому слою в многовидовом кодировании (например, MV-HEVC), 3DV кодировании (например, 3D-HEVC), или масштабируемом кодировании видео (например, SHVC). Данные в базовом слое битового потока могут быть декодированы без ссылки на данные в любом другом слое битового потока. Если единица NAL не относится к базовому слою в многовидовом кодировании, 3DV, или масштабируемом кодировании видео, элемент синтаксиса `nuh_layer_id` единицы NAL может иметь ненулевое значение.

[0079] Кроме того, некоторые компоненты вида/представления

слоя внутри слоя могут быть декодированы без ссылки на другие компоненты вида/представления слоя внутри того же самого слоя. Следовательно, единицы NAL, инкапсулирующие данные определенных компонентов вида/представлений слоя у слоя, могут быть удалены из битового потока не оказывая влияния на возможность декодирования других компонентов вида/представлений слоя в слое. Удаление единиц NAL, инкапсулирующих данные таких компонентов вида/представлений слоя, может сократить частоту кадров битового потока. Подмножество компонентов вида/представлений слоя внутри слоя, которое может быть декодировано без ссылки на другие компоненты вида/представления слоя внутри слоя, может упоминаться в данном документе как «суб-слой» или «временной суб-слой».

[0080] Единицы NAL включают в себя элементы синтаксиса `temporal_id`, которые указывают временные идентификаторы (т.е., `TemporalId`) единиц NAL. Временной идентификатор единицы NAL идентифицирует суб-слой, к которому принадлежит единица NAL. Следовательно, каждый суб-слой битового потока может иметь разный временной идентификатор. В целом, если временной идентификатор первой единицы NAL слоя меньше временного идентификатора второй единицы NAL того же самого слоя, данные, инкапсулируемые первой единицей NAL, могут быть декодированы без ссылки на данные, инкапсулируемые второй единицей NAL.

[0081] Битовый поток может быть ассоциирован со множеством рабочих точек. Каждая рабочая точка битового потока ассоциирована с набором идентификаторов слоя (например, набором значений `nuh_layer_id`) и временным идентификатором. Набор идентификаторов слоя может быть обозначен как `OpLayerIdSet`, а временной идентификатор может быть обозначен как `TemporalID`. Если идентификатор слоя единицы NAL находится в наборе рабочей точки идентификаторов слоя и временной идентификатор единицы NAL меньше или равен временному идентификатору рабочей точки, единица NAL ассоциирована с рабочей точкой. Следовательно, рабочая точка может соответствовать подмножеству единиц NAL в битовом потоке. HEVC определяет рабочую точку как битовый поток, созданный из другого битового потока посредством операции

процесса извлечения битового суб-потoka с другим битовым потоком, целевым наивысшим TemporalId, и целевым списком идентификаторов слоя в качестве вводов.

[0082] Как введено выше, данное раскрытие относится к сохранению видеоконтента в файле на основе базового формата мультимедийного файла ISO (ISOBMFF). В частности, данное раскрытие описывает разнообразные методики для сохранения видеопотоков, содержащих несколько кодированных слоев, при этом каждый слой может быть масштабируемым слоем, видом текстуры, видом глубины, или другими типами слоев или видов. Методики данного раскрытия могут быть применены к, например, сохранению видеоданных MV-HEVC, видеоданных SHVC, видеоданных 3D-HEVC, и/или других типов видеоданных.

[0083] Теперь кратко будут обсуждаться форматы файла и стандарты формата файла. Стандарты формата файла включают в себя базовый формат мультимедийного файла ISO (ISOBMFF, ISO/IEC 14496-12, далее, «ISO/IEC 14996-12») и другие стандарты формата файла, полученные из ISOBMFF, включая формат файла MPEG-4 (ISO/IEC 14496-14), формат файла 3GPP (3GPP TS 26.244) и формат файла AVC (ISO/IEC 14496-15, далее «ISO/IEC 14996-15»). Следовательно, ISO/IEC 14496-12 описывает базовый формат мультимедийного файла ISO. Другие документы расширяют базовый формат мультимедийного файла ISO для конкретных приложений. Например, ISO/IEC 14496-15 описывает перенос видео структурированного в единицах NAL в базовом формате мультимедийного файла ISO. H.264/AVC и HEVC, как впрочем и их расширения, являются примерами видео структурированного в единицах NAL. ISO/IEC 14496-15 включает в себя разделы, описывающие перенос единиц NAL H.264/AVC. Дополнительно, раздел 8 ISO/IEC 14496-15 описывает перенос единиц NAL HEVC.

[0084] ISOBMFF используется в качестве базы для многих форматов инкапсуляции кодека, таких как Формат Файла AVC, как впрочем и для многих форматов мультимедийного контейнера, таких как Формат Файла MPEG-4, Формат Файла 3GPP (3GP), и Формат Файла DVB. В дополнение к непрерывному мультимедиа, такому как аудио и видео, статическое мультимедиа, такое как изображения, как

впрочем и метаданные, может быть сохранено в файле, который согласуется с ISOBMFF. Файлы, структурированные в соответствии с ISOBMFF, могут быть использованы для многих целей, включая локальное воспроизведение мультимедийного файла, прогрессивная загрузка удаленного файла, сегменты для Динамической Адаптивной Поточковой Передачи через HTTP (DASH), контейнеры для контента, который должен быть передан потоковым образом, и его инструкции формирования пакетов, и запись принимаемых в режиме реального времени мультимедийных потоков. Следовательно, несмотря на то, что изначально разработан для сохранения, ISOBMFF доказал свою ценность для потоковой передачи, например, для прогрессивной загрузки или DASH. Применительно к целям потоковой передачи, могут быть использованы фрагменты фильма, определенные в ISOBMFF.

[0085] Файл, согласующийся с форматом файла HEVC, может содержать ряд объектов, именуемых контейнерами. Контейнер может быть объектно-ориентированным строительным блоком, который определяется уникальным идентификатором типа и длиной. Например, контейнер может быть элементарной структурой синтаксиса в ISOBMFF, включающей в себя четырехзначный кодированный тип контейнера, счетчик байтов контейнера, и полезную нагрузку. Другими словами, контейнер может быть структурой синтаксиса, содержащей кодированный тип контейнера, счетчик байтов контейнера, и полезную нагрузку. В некоторых случаях, все данные в файле, согласующемся с форматом файла HEVC, могут содержаться внутри контейнеров и могут отсутствовать данные в файле, которые не находятся в контейнере. Следовательно, файл ISOBMFF может состоять из последовательности контейнеров, и контейнеры могут содержать другие контейнеры. Например, полезная нагрузка контейнера может включать в себя один или более дополнительных контейнеров. Фиг. 5А, Фиг. 5В, и Фиг. 6, описываемые подробно в другом месте в данном раскрытии, показывают примеры контейнеров внутри файла, в соответствии с одной или более методик данного раскрытия.

[0086] Файл, согласующийся с ISOBMFF, может включать в себя разнообразные типы контейнеров. Например, файл, согласующийся с

ISOBMFF, может включать в себя контейнер типа файла, контейнер мультимедийных данных, контейнер фильма, контейнер фрагмента фильма, и т.п. В данном примере, контейнер типа файла включает в себя тип файла и информацию совместимости. Контейнер мультимедийных данных может содержать выборки (например, кодированные картинки). Контейнер фильма («moov») содержит метаданные для непрерывных мультимедийных потоков, присутствующих в файле. Каждый из непрерывных мультимедийных потоков может быть представлен в файле в качестве дорожки. Например, контейнер фильма может содержать метаданные касательно фильма (например, логические и хронометражные зависимости между выборками, а также указатели на местоположения выборок). Контейнеры фильма могут включать в себя несколько типов суб-контейнеров. Суб-контейнеры в контейнере фильма могут включать в себя один или более контейнеров дорожки. Контейнер дорожки может включать в себя информацию касательно индивидуальной дорожки фильма. Контейнер дорожки может включать в себя контейнер заголовка дорожки, который указывает общую информацию одной дорожки. В дополнение, контейнер дорожки может включать в себя контейнер мультимедиа, который содержит контейнер информации мультимедиа. Контейнер информации мультимедиа может включать в себя контейнер таблицы выборок, который содержит данные, индексирующие выборки мультимедиа в дорожке. Информация в контейнере таблицы выборок может быть использована, чтобы располагать выборки по времени и, применительно к каждой из выборок дорожки, тип, размер, контейнер, и смещение в том контейнере у выборки. Следовательно, метаданные для дорожки заключены в контейнере Дорожки («trak»), тогда как контент мультимедиа дорожки либо заключен в контейнере Мультимедийных Данных («mdat»), либо непосредственно в отдельном файле. Контент мультимедиа для дорожек содержит (например, состоит из) последовательности выборок, таких как единицы доступа аудио или видео.

[0087] ISOBMFF указывает следующие типы дорожек: дорожка мультимедиа, которая содержит элементарный поток мультимедиа, дорожка подсказки, которая либо включает в себя инструкции

передачи мультимедиа, либо представляет собой принятый поток пакета, и дорожка синхронизированных метаданных, которая содержит синхронизированные по времени метаданные. Метаданные для каждой дорожки включают в себя список записей описания выборки, каждая предоставляющая формат кодирования или инкапсуляции, использованный в дорожке, и данные инициализации, необходимые для обработки того формата. Каждая выборка ассоциирована с одной из записей описания выборки дорожки.

[0088] ISOBMFF обеспечивает указание характерных для выборки метаданных с помощью разнообразных механизмов. Конкретные контейнеры внутри контейнера Таблицы Выборок («stbl») были стандартизованы, чтобы реагировать на общие потребности. Например, контейнер Выборки Синхронизации («stss») является контейнером внутри контейнера Таблицы Выборок. Контейнер Выборки Синхронизации используется, чтобы перечислять выборки произвольного доступа у дорожки. Данное раскрытие может относиться к выборке, перечисленной в контейнере Выборки Синхронизации, как к выборке синхронизации. В другом примере, механизм группирования выборок обеспечивает соотнесение выборок в соответствии с четырехзначным типом группирования с группами выборок, которые совместно используют одно и то же свойство, указанное в качестве записи описания группы выборок в файле. В ISOBMFF указано несколько типов группирования.

[0089] Контейнер таблицы выборок может включать в себя один или более контейнеров SampleToGroup и один или более контейнеров описания группы выборок (т.е., контейнеры SampleGroupDescription). Контейнер SampleToGroup может быть использован, чтобы определять группу выборок, к которой принадлежит выборка, наряду с ассоциированным описанием группы выборок. Другими словами, контейнер SampleToGroup может указывать группу, к которой принадлежит выборка. Контейнер SampleToGroup может иметь тип контейнера «sbgr». Контейнер SampleToGroup может включать в себя элемент типа группирования (например, grouping_type). Элемент типа группирования может быть целым числом, которое идентифицирует тип (т.е., критерий, используемый, чтобы формировать группы выборок) группирования

выборки. Кроме того, контейнер `SampleToGroup` может включать в себя одну или более записей. Каждая запись в контейнере `SampleToGroup` может быть ассоциирована с разными, не перекрывающимися рядами последовательных выборок в дорожке. Каждая запись может указывать элемент счетчика выборки (например, `sample_count`) и элемент индекса описания группы (например, `group_description_index`). Элемент счетчика выборки записи может указывать количество выборок, ассоциированных с записью. Другими словами, элемент счетчика выборки записи может быть целым числом, которое задает количество последовательных выборок с одним и тем же дескриптором группы выборок. Элемент индекса описания группы может идентифицировать контейнер `SampleGroupDescription`, который содержит описание выборок, ассоциированных с записью. Элементы индекса описания группы нескольких записей могут идентифицировать один и тот же контейнер `SampleGroupDescription`.

[0090] На данный момент конфигурации формата файла могут иметь одну или более проблем. Чтобы хранить видеоконтент конкретного кодека видео на основании ISOBMFF, может потребоваться спецификация формата файла для этого кодека видео. Для сохранения видеопотоков, содержащих несколько слоев, таких как MV-HEVC и SHVC, существует возможность повторного использования некоторых концепций из формата файла SVC и MVC. Тем не менее, многие части не могут быть непосредственно использованы для видеопотоков SHVC и MV-HEVC. Непосредственное применение формата файла HEVC обладает, по меньшей мере, следующими недостатками: битовые потоки SHVC и MV-HEVC могут начинаться с единицы доступа, которая содержит картинку IRAP в базовом слое, но также может содержать другие картинки не-IRAP в других слоях и наоборот. В настоящее время выборка синхронизации не обеспечивает указания такой точки для произвольного доступа.

[0091] Данное раскрытие описывает потенциальные решения вышеприведенных проблем, как впрочем и обеспечивает другие потенциальные улучшения, чтобы обеспечивать эффективное и гибкое сохранение видеопотоков, содержащих несколько слоев. Методики, описанные в данном раскрытии, потенциально применяются к любому

формату файла для сохранения такого видеоконтента, кодированного любым кодеком видео, несмотря на то, что описание является конкретным для сохранения видеопотоков SHVC и MV-HEVC на основании формата файла HEVC, который указывается в Статье 8 документа ISO/IEC 14496-15.

[0092] Примерная реализация некоторых методик данного раскрытия описывается ниже. Примерная реализация, описываемая ниже, основана на последней интегрированной спецификации 14496-15 в выходном документе MPEG W13478. Изменения Приложении А (показанные с помощью подчеркивания) и добавленные разделы (Раздел 9 для SHVC и Раздел 10 для MV-HEVC) включены ниже. Другими словами, конкретные примеры данного раскрытия могут модифицировать Приложение А документа ISO/IEC 14496-15 и могут добавлять разделы 9 и/или 10 в документ ISO/IEC 14496-15. Текст, показанный с помощью подчеркивания и двойного подчеркивания может быть особой значимости для примеров данного раскрытия. Несмотря на то, что понятие SHVC используется в разных местах в описываемых в данном документе примерах, конфигурация согласно данному раскрытию предназначена фактически не только для поддержки кодека SHVC, но вместо этого могут поддерживаться все многослойные кодеки, включая MV-HEVC, 3D-HEVC, при условии, что явно не упоминается обратное.

[0093] Спецификация ISOBMFF указывает шесть типов Точек Доступа к Потoku (SAP) для использования с DASH. Первые два типа SAP (типы 1 и 2) соответствуют картинкам IDR в H.264/AVC и HEVC. Третий тип SAP (тип 3) соответствует точкам произвольного доступа открытой-GOP, следовательно картинкам BLA или CRA в HEVC. Четвертый тип SAP (тип 4) соответствует точкам произвольного доступа GDR.

[0094] В текущем формате файла L-HEVC, некоторая информация высокого уровня (например, информация слоев в битовом потоке, скорость передачи битов, частота кадров, временные суб-слои, параллелизм, рабочие точки, и т.д.) сигнализируется в контейнерах LHEVCSampleEntry, HEVCLHVCSampleEntry, LHVCDecoderConfigurationRecord, информации о контенте дорожки

('tcon') и OperationPointsInformationBox ('oinf'). В одном примере, структура синтаксиса вышеупомянутых контейнеров является следующей:

[0095] На основании текущей структуры вышеупомянутых контейнеров, и информации, которая содержится в них, для того чтобы воспроизводить контент в файле, проигрыватель может быть выполнен с возможностью сначала находить контейнер 'oinf' (только один в файле), чтобы знать какие рабочие точки включены, и затем выбирать одну из рабочих точек, для воспроизведения. Проигрыватель видео затем может проверять контейнеры 'tcon' (один в каждой дорожке, содержащей видео L-HEVC), чтобы знать, какие дорожки содержат слои выбранных рабочих точек.

```
//запись выборки LHVC и HEVCLHVC
класс LHEVCConfigurationBox расширяет Box('lhvc') {
    LHEVCDecoderConfigurationRecord() LHEVCConfig;
}
класс HEVCLHVCSampleEntry() расширяет HEVCSampleEntry() {
    LHEVCConfigurationBox lhvcconfig;
    MPEG4BitRateBox (); // опциональный
    MPEG4ExtensionDescriptorsBox (); // опциональный
    extra_boxes контейнеры; // опциональный
}
// Использовать данную дорожку при несовместимости с HEVC
класс LHEVCSampleEntryO расширяет VisualSampleEntry ('lhv1',
или 'lhe1') {
    LHVCConfigurationBox lhvcconfig;
    MPEG4BitRateBox (); // опциональный
    MPEG4ExtensionDescriptorsBox (); // опциональный
    Контейнер extra_boxes[];
}
aligned(8) класс LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion=1;
    unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
```

```

unsigned int(32) general_profile_compatibility_flags;
unsigned int(48) general_constraint_indicator_flags;
unsigned int(8) general_level_idc;
bit(1) complete_representation;
bit(3) зарезервировано='111'b;
unsigned int(12) min_spatial_segmentation_idc;
bit(6) зарезервировано='111111'b;
unsigned int(2) parallelismType;
bit(6) зарезервировано='111111'b;
unsigned int(2) chromaFormat;
bit(5) зарезервировано='11111'b;
unsigned int(3) bitDepthLumaMinus8;
bit(5) зарезервировано='11111'b;
unsigned int(3) bitDepthChromaMinus8;
bit(16) avgFrameRate;
bit(2) constantFrameRate;
bit(3) numTemporalLayers;
bit(1) temporalIdNested;
unsigned int(2) lengthSizeMinusOne;
unsigned int(8) numOfArrays;
for (j=0; j<numOfArrays; j++) {
bit(1) array_completeness;
unsigned int(1) зарезервировано=0;
unsigned int(6) NAL_unit_type;
unsigned int(16) numNalus;
for (i=0; i<numNalus; i++) {
unsigned int(16) nalUnitLength;
bit(8*nalUnitLength) nalUnit;
}
}
unsigned int(16) operationPointIdx;
}

класс      TrackContentsInfoBox      расширяет      FullBox('tcon',
версия=0, 0)){
    unsigned int (2) зарезервировано

```

```

unsigned int (6) num_layers_in_track
for (i=0; i<num_layers_in_track; i++){
unsigned int (4) зарезервировано
unsigned int (6) layer_id
unsigned int (3) min_sub_layer_id
unsigned int (3) max_sub_layer_id
}
}

класс OperationPointsInformation расширяет FullBox('oinf,
версия=0, 0){
    unsigned int(16) scalability_mask
    unsigned int(2) зарезервировано
    unsigned int(6) num_profile_tier_level
    for (i=1; i<=num_profile_tier_level; i++) {
    unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc;
    }
    unsigned int(16) num_operation_points
    for (i=0; i<num_operation_points) {
    unsigned int(16) operation_point_id
    unsigned int(8) max_temporal_id;
    unsigned int(8) layer_count
    for (i=0; i<layer_count; i++) {
    unsigned int(8) ptl_idx
    unsigned int(6) layer_id;
    unsigned int(1) is_outputlayer;
    unsigned int(1) is_alterate_outputlayer;
    }
    }
    unsigned int(8) max_layer_count
    for (i=0; i<max_layer_count; i++) {

```

```

unsigned int(8) dependent_layerID
unsigned int(8) num_layers_dependent_on
for (j=0; j<num_layers_dependent_on; j++) {
    unsigned int(8) dependent_on_layerID
}
for (j=0; j<16; j++) {
    if (scalability_mask & (1<<j))
        unsigned int(8) dimension_identifier
}
}
}

```

[0096] На основании текущей структуры вышеприведенных контейнеров, и информации, которая содержится в них, для того чтобы воспроизводить контент в файле, проигрыватель может быть выполнен с возможностью сначала нахождения контейнера 'oinf' (только один в файле), чтобы знать какие рабочие точки включены, и затем выбирать одну из рабочих точек, для воспроизведения. Проигрыватель видео затем может проверять контейнеры 'tcon' (один в каждой дорожке, содержащей видео L-HEVC), чтобы знать, какие дорожки содержат слои выбранных рабочих точек.

[0097] С учетом вышеприведенного основного использования текущей конфигурации, данное раскрытие предлагает, чтобы еще информация, такая как формат представления (который включает в себя пространственное разрешение, битовую глубину, и формат цвета), скорость передачи битов, и частота кадров, была включена в контейнер 'oinf' для обеспечения выбора рабочих точек. Запись выборки в каждой дорожке не включает в себя один набор такой информации, а только для конкретной рабочей точки. Когда в одной дорожке содержится несколько рабочих точек, информация для других рабочих точек отсутствует.

[0098] Другая проблема связана с тем, что семантика многих полей в контейнере LHEVCDecoderConfigurationRecord не является четкой, и некоторые из них являются запутывающими. Например, профиль, ярус и уровень (PTL), chromaFormat, bitDepthLumaMinus8, и bitDepthChromaMinus8 являются характерными для слоя

свойствами, однако в настоящее время упоминаются как применяемые к рабочей точке, указываемой `operationPointIdx`. Когда рабочая точка содержит более одного слоя, семантика является просто неточной.

[0099] Фактически, на основании этапов обычного базового использования конфигурации, некоторая информация в записи выборки не является действительно полезной, в частности когда присутствует достаточно информации в контейнере 'oinf' для выбора рабочей точки.

[0100] Еще одна проблема связана с тем, что в SHVC и MV-HEVC, PTL сигнализируется только для каждого необходимого слоя (т.е., слоя, который является либо слоем вывода, либо слоем, на который напрямую или косвенно ссылается слой вывода внутри рабочей точки или как того, так и другого), а не для любого не являющегося необходимым слоя (слоя, который не является необходимым). Вследствие этого, в конфигурации формата файла может быть ненужным сигнализировать PTL для не являющихся необходимыми слоев.

[0101] Сущность способов и методик, которые описываются в данном раскрытии, перечисляется ниже. Примерные подробные реализации предоставляются в последующих разделах. Способы и методики данного раскрытия могут применяться независимо или могут применяться в сочетании.

[0102] Первая методика данного раскрытия включает в себя удаление сигнализации `MPEG4BitRateBox()` после `LHEVCConfigurationBox` внутри записи выборки LHEVC и записи выборки HEVCLHVC. Вместо этого обеспечивая сигнализацию информации скорости передачи битов для каждой из рабочих точек в контейнере 'oinf'.

[0103] Вторая методика данного раскрытия включает в себя сигнализацию информации формата представления (который включает в себя пространственное разрешение, битовую глубину, и формат цвета) для каждой рабочей точки точек в контейнере 'oinf'.

[0104] Третья методика данного раскрытия включает в себя удаление из `LHEVCDecoderConfigurationRecord` информации PTL,

информации формата представления, информации частоты кадров, которая либо уже предоставлена в контейнере 'oinf', либо предлагается добавлять в контейнер 'oinf'. Оставшаяся информация в LHEVCDecoderConfigurationRecord применяется к всем слоям, которые содержатся в дорожке. В другом примере третьей методики, структура LHEVCDecoderConfigurationRecord реструктуризируется таким образом, что информация формата представления и информация частоты кадров, и возможно дополнительные параметры/информация (например, информация параллелизма), сигнализируется для каждого слоя. Элемент синтаксиса unsigned int(2) parallelismType в LHEVCDecoderConfigurationRecord может указывать, какой тип свойств(а) параллельного декодирования может быть использован, чтобы декодировать картинку в слое. Тайл, фронт волны, или слайсы являются примерами механизмов сегментации картинки, которые могут быть использованы, чтобы способствовать параллельной обработке.

[0105] Четвертая методика данного раскрытия включает в себя удаление operationPointIdx из LHEVCDecoderConfigurationRecord. В другом примере четвертой методики, обеспечивается сигнализация списка индексов рабочей точки, которые ассоциированы с дорожкой в LHEVCDecoderConfigurationRecord.

[0106] Пятая методика данного раскрытия включает в себя изменение семантики поля layer_count в контейнере 'oinf', чтобы подсчитывать только необходимые слои рабочей точки.

[0107] Примерные реализации способов и методик раскрытия описываются ниже. В примерах ниже, показаны изменения текста по отношению к формату файла HEVC и LHEVC. Добавленный текст показан между идентификаторами [НАЧАЛО ВСТАВКИ] и [КОНЕЦ ВСТАВКИ]. Удаленный текст показан между идентификаторами [НАЧАЛО УДАЛЕНИЯ] и [КОНЕЦ УДАЛЕНИЯ].

[0108] Ниже описывается первая реализация.

Данный раздел описывает подробные модификации для сигнализации LHEVCSampleEntry, HEVCLHVCSampleEntry, LHVCDerivationRecord и OperationPointsInformationBox ('oinf') для методик 1, 2, 3 (не включая ее пример a.), 4 (не

включая ее пример а.) и 5 раскрытия.

```

класс LHEVCConfigurationBox расширяет Box('lhvC') {
    LHEVCDecoderConfigurationRecord() LHEVCConfig;
}

класс HEVCLHVCSampleEntry() расширяет HEVCSampleEntry() {
    LHEVCConfigurationBox lhvcconfig;
    [НАЧАЛО УДАЛЕНИЯ] MPEG4BitRateBox (); // опциональный
[КОНЕЦ УДАЛЕНИЯ]
    MPEG4ExtensionDescriptorsBox (); // опциональный
    extra_boxes контейнеры; // опциональный
}

// Использовать данную дорожку при несовместимости с HEVC
класс LHEVCSampleEntry() расширяет VisualSampleEntry ('lhv1',
или 'lhe1') {
    LHVCCConfigurationBox lhvcconfig;
    [НАЧАЛО УДАЛЕНИЯ] MPEG4BitRateBox (); // опциональный
[КОНЕЦ УДАЛЕНИЯ]
    MPEG4ExtensionDescriptorsBox (); // опциональный
    Контейнер extra_boxes[];
}

aligned(8) класс LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion=1;
    [НАЧАЛО УДАЛЕНИЯ] unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc; [КОНЕЦ УДАЛЕНИЯ]
    bit(1) complete_representation;
    bit(3) зарезервировано='111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) зарезервировано='111111'b;
    unsigned int(2) parallelismType;
    [НАЧАЛО УДАЛЕНИЯ] bit(6) зарезервировано='111111'b;
    unsigned int(2) chromaFormat;

```

```

    bit(5) зарезервировано='11111'b;
    unsigned int(3) bitDepthLumaMinus8;
    bit(5) зарезервировано='11111'b;
    unsigned int(3) bitDepthChromaMinus8;
    bit(16) avgFrameRate;
    bit(2) constantFrameRate; [КОНЕЦ УДАЛЕНИЯ]
    [НАЧАЛО ВСТАВКИ]    bit(2)    зарезервировано='11'b;    [КОНЕЦ
ВСТАВКИ]
    bit(3) numTemporalLayers;
    bit(1) temporalIdNested;
    unsigned int(2) lengthSizeMinusOne;
    unsigned int(8) numOfArrays;
    for (j=0; j<numOfArrays; j++) {
    bit(1) array_completeness;
    unsigned int(1) зарезервировано=0;
    unsigned int(6) NAL_unit_type;
    unsigned int(16) numNalus;
    for (i=0; i<numNalus; i++) {
    unsigned int(16) nalUnitLength;
    bit(8*nalUnitLength) nalUnit;
    }
    }
    [НАЧАЛО УДАЛЕНИЯ]    unsigned    int(16)    operationPointIdx;
[КОНЕЦ УДАЛЕНИЯ]
    }

    класс OperationPointsInformation расширяет FullBox('oinf',
версия=0, 0){
        unsigned int(16) scalability_mask
        unsigned int(2) зарезервировано
        unsigned int(6) num_profile_tier_level
        for (i=1; i<=num_profile_tier_level; i++) {
        unsigned int(2) general_profile_space;
        unsigned int(1) general_tier_flag;
        unsigned int(5) general_profile_idc;
        unsigned int(32) general_profile_compatibility_flags;

```

```

unsigned int(48) general_constraint_indicator_flags;
unsigned int(8) general_level_idc;
}
unsigned int(16) num_operation_points
for (i=0; i<num_operation_points) {
unsigned int(16) operation_point_id
unsigned int(8) max_temporal_id;
unsigned int(8) layer_count;
for (i=0; i<layer_count; i++) {
unsigned int(8) ptl_idx
unsigned int(6) layer_id;
unsigned int(1) is_outputlayer;
unsigned int(1) is_alternate_outputlayer;
}
[НАЧАЛО ВСТАВКИ]
unsigned int(16) minPicWidth;
unsigned int(16) minPicHeight;
unsigned int(16) maxPicWidth;
unsigned int(16) maxPicHeight;
unsigned int(2) maxChromaFormat;
unsigned int(3) maxBitDepthMinus8;
unsigned int(1) зарезервировано
unsigned int(1) frame_rate_info_flag
unsigned int(1) bit_rate_info_flag
if (frame_rate_info_flag) {
bit(16) avgFrameRate;
unsigned int(6) зарезервировано
bit(2) constantFrameRate;
}
if (bit_rate_info_flag) {
unsigned int(32) maxBitRate;
unsigned int(32) avgBitRate;
}[КОНЕЦ ВСТАВКИ]
}
unsigned int(8) max_layer_count
for (i=0; i<max_layer_count; i++) {

```

```

unsigned int(8) dependent_layerID
unsigned int(8) num_layers_dependent_on
for (j=0; j<num_layers_dependent_on; j++) {
unsigned int(8) dependent_on_layerID
}
for (j=0; j<16; j++) {
if (scalability mask & (1<<j))
unsigned int(8) dimension_identifier
}
}
}

```

...

layer_count: Данное поле указывает количество [НАЧАЛО ВСТАВКИ] необходимых [КОНЕЦ ВСТАВКИ] слоев, которые являются частью [НАЧАЛО ВСТАВКИ] [КОНЕЦ ВСТАВКИ] [НАЧАЛО УДАЛЕНИЯ] [КОНЕЦ УДАЛЕНИЯ] рабочей точки.

...

[НАЧАЛО ВСТАВКИ]

minPicWidth указывает наименьшее значение индикаторов ширины яркости как определено параметром pic_width_in_luma_samples в ISO/IEC 23008-2 для потока рабочей точки.

minPicHeight указывает наименьшее значение индикаторов высоты яркости как определено параметром pic_height_in_luma_samples в ISO/IEC 23008-2 для потока рабочей точки.

maxPicWidth указывает наибольшее значение индикаторов ширины яркости как определено параметром pic_width_in_luma_samples в ISO/IEC 23008-2 для потока рабочей точки.

maxPicHeight указывает наибольшее значение индикаторов высоты яркости как определено параметром pic_height_in_luma_samples в ISO/IEC 23008-2 для потока рабочей точки.

maxChromaFormat указывает наибольшее значение индикатора

`chroma_format` как определено параметром `chroma_format_idc` в ISO/IEC 23008-2 для потока рабочей точки.

`maxBitDepthMinus8` указывает наибольшее значение индикаторов битовой глубины яркости и цветности как определено параметрами `bit_depth_luma_minus8` и `bit_depth_chroma_minus8`, соответственно, в ISO/IEC 23008-2 для потока рабочей точки.

`frame_rate_info_flag` равный 0 указывает на то, что информация частоты кадров не присутствует для рабочей точки. Значение 1 указывает на то, что информация частоты кадров присутствует для рабочей точки.

`bit_rate_info_flag` равный 0 указывает на то, что информация скорости передачи битов не присутствует для рабочей точки. Значение 1 указывает на то, что информация скорости передачи битов присутствует для рабочей точки.

`avgFrameRate` задает среднюю частоту кадров в единицах кадров/(265 секунд) для рабочей точки. Значение 0 указывает неуказанную среднюю частоту кадров.

`constantFrameRate` равный 1 указывает на то, что поток рабочей точки имеет постоянную частоту кадров. Значение 2 указывает на то, что представление каждого временного слоя в потоке рабочей точки имеет постоянную частоту кадров. Значение 0 указывает на то, что поток рабочей точки может или может не иметь постоянную частоту кадров.

`maxBitRate` задает максимальную скорость передачи битов в битах/секунду у потока рабочей точки, через любое окно в одну секунду.

`avgBitRate` задает среднюю скорость передачи битов в битах/секунду у потока рабочей точки.

...

[КОНЕЦ ВСТАВКИ]

[0109] Ниже описывается вторая реализация.

Данный раздел описывает подробные модификации сигнализации `LHVCDerDecoderConfigurationRecord` для примера 3(a) раскрытия.

```
aligned(8) класс LHEVCDerDecoderConfigurationRecord {
    unsigned int(8) configurationVersion=1;
```

```

[НАЧАЛО УДАЛЕНИЯ] unsigned int(2) general_profile_space;
unsigned int(1) general_tier_flag;
unsigned int(5) general_profile_idc;
unsigned int(32) general_profile_compatibility_flags;
unsigned int(48) general_constraint_indicator_flags;
unsigned int(8) general_level_idc;
bit(1) complete_representation;
bit(3) зарезервировано='111'b; [КОНЕЦ УДАЛЕНИЯ]

[НАЧАЛО ВСТАВКИ] bit(2) зарезервировано='11'b; [КОНЕЦ
ВСТАВКИ]

[НАЧАЛО ВСТАВКИ] unsigned int(6) num_layers; [КОНЕЦ
ВСТАВКИ]

for (j=0; j<num_layers; j++) {
[НАЧАЛО ВСТАВКИ] unsigned int(8) layer_id; [КОНЕЦ ВСТАВКИ]
unsigned int(12) min_spatial_segmentation_idc;
bit(6) зарезервировано='111111'b;
unsigned int(2) parallelismType;
bit(6) зарезервировано='111111'b;
unsigned int(2) chromaFormat;
[НАЧАЛО ВСТАВКИ] bit(6) зарезервировано='111111'b; [КОНЕЦ
ВСТАВКИ]

[НАЧАЛО УДАЛЕНИЯ] bit(5) зарезервировано='11111'b; [КОНЕЦ
УДАЛЕНИЯ]

unsigned int(3) bitDepthLumaMinus8;
bit(5) зарезервировано='11111'b;
unsigned int(3) bitDepthChromaMinus8;
[НАЧАЛО ВСТАВКИ] bit(5) зарезервировано='11111'b; [КОНЕЦ
ВСТАВКИ]

[НАЧАЛО УДАЛЕНИЯ] bit(16) avgFrameRate;
bit(2) constantFrameRate; [КОНЕЦ УДАЛЕНИЯ]
bit(3) numTemporalLayers;
bit(1) temporalIdNested;

[НАЧАЛО ВСТАВКИ] bit(4) зарезервировано='1111'b; [КОНЕЦ
ВСТАВКИ]

```

```

    }
    [НАЧАЛО ВСТАВКИ]    bit(1)    complete_representation;    [КОНЕЦ
ВСТАВКИ]
    unsigned int(2) lengthSizeMinusOne;
    [НАЧАЛО ВСТАВКИ]    bit(5)    зарезервировано='11111'b;    [КОНЕЦ
ВСТАВКИ]
    unsigned int(8) numOfArrays;
    for (j=0; j<numOfArrays; j++) {
    bit(1) array_completeness;
    unsigned int(1) арезервировано=0;
    unsigned int(6) NAL_unit_type;
    unsigned int(16) numNalus;
    for (i=0; i<numNalus; i++) {
    unsigned int(16) nalUnitLength;
    bit(8*nalUnitLength) nalUnit;
    }
    }
    [НАЧАЛО УДАЛЕНИЯ]    unsigned    int(16)    operationPointIdx;
[КОНЕЦ УДАЛЕНИЯ]
    }
    [НАЧАЛО ВСТАВКИ]
    num_layers указывает количество слоев в дорожке.
    layer_id указывает значение ID слоя для которого
предоставляется информация в данном цикле.
    [КОНЕЦ ВСТАВКИ]

```

[0110] Ниже описывается третья реализация.

Данный раздел описывает подробные модификации сигнализации LHVCDecoderConfigurationRecord для примера 4(a) раскрытия.

```

aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion=1;
    unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;

```

```

    unsigned int(8) general_level_idc;
    bit(1)          complete_representation;          bit(3)
зарезервировано='111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) зарезервировано='111111'b;
    unsigned int(2) parallelismType;
    bit(6) зарезервировано='111111'b;
    unsigned int(2) chromaFormat;
    bit(5) зарезервировано='11111'b;
    unsigned int(3) bitDepthLumaMinus8;
    bit(5) зарезервировано='11111'b;
    unsigned int(3) bitDepthChromaMinus8;
    bit(16) avgFrameRate;
    bit(2) constantFrameRate;
    bit(3) numTemporalLayers;
    bit(1) temporalIdNested;
    unsigned int(2) lengthSizeMinusOne;
    unsigned int(8) numOfArrays;
    for (j=0; j<numOfArrays; j++) {
    bit(1) array_completeness;
    unsigned int(1) зарезервировано=0;
    unsigned int(6) NAL_unit_type;
    unsigned int(16) numNalus;
    for (i=0; i<numNalus; i++) {
    unsigned int(16) nalUnitLength;
    bit(8*nalUnitLength) nalUnit;
    }
    }
    [НАЧАЛО УДАЛЕНИЯ]    unsigned    int(16)    operationPointIdx;
[КОНЕЦ УДАЛЕНИЯ]
    [НАЧАЛО ВСТАВКИ]
    unsigned int(16) numOfOperationPoints;
    for (j=0; j<numOfOperationPoints; j++) {
    unsigned int(16) operationPointIdx;
    } [КОНЕЦ ВСТАВКИ]

```

}

[НАЧАЛО ВСТАВКИ] numOperationPoints: Данное поле сигнализирует количество рабочих точек, которые доступны для дорожки. [КОНЕЦ ВСТАВКИ]

operationPointIdx: Данное поле сигнализирует индекс рабочей точки, которая фиксируется в контейнере информации рабочей точки. [НАЧАЛО УДАЛЕНИЯ] Значения general_profile_space, general_tier_flag, general_profile_idc, general_profile_compatibility_flags, general_constraint_indicator_flag и general_level_idc в LHEVCDecoderConfigurationRecord должны быть точно такими же, как соответствующие значения у operationPointIdx-ой рабочей точки в контейнере информации рабочей точки. [КОНЕЦ УДАЛЕНИЯ] [НАЧАЛО ВСТАВКИ] Значение max_temporal_id в operationPointIdx-ой рабочей точке в контейнере информации рабочей точки должно быть меньше или равно значению numTemporalLayers.[КОНЕЦ ВСТАВКИ]

ПРИМЕЧАНИЕ Дорожка может быть ассоциирована с одним или [НАЧАЛО УДАЛЕНИЯ] представлять собой [КОНЕЦ УДАЛЕНИЯ] более чем один набор слоев вывода [НАЧАЛО УДАЛЕНИЯ] и, следовательно, более чем одним профилем [КОНЕЦ УДАЛЕНИЯ]. Проигрыватель может выяснять, какие слои должны быть декодированы и какие слои должны быть выведены в соответствии с информацией профиля в LHEVCDecoderConfigurationRecord [НАЧАЛО ВСТАВКИ] для выбранной рабочей точки с индексом operationPointIdx [КОНЕЦ ВСТАВКИ] посредством исследования информации, предоставленной для operationPointIdx-ой рабочей точки в контейнере информации рабочей точки.

ПРИМЕЧАНИЕ Для каждого вспомогательного слоя картинки, включенного в дорожку, рекомендуется включать, внутри nalUnit, единицу NAL SEI, содержащую декларативное сообщение SEI, такое как сообщение SEI информации представления глубины для вспомогательных слоев кортике по глубине, указывающее характеристики вспомогательного слоя картинки.

[0111] Фиг. 2 является структурной схемой, иллюстрирующей примерный кодер 20 видео, который может реализовывать методики, описываемые в данном раскрытии. Кодер 20 видео может быть

выполнен с возможностью вывода с одним видом, многовидовых, масштабируемых, 3D, и других типов видеоданных. Кодер 20 видео может быть выполнен с возможностью вывода видео на объект 27 пост-обработки. Объект 27 пост-обработки предназначен для представления примера объекта видео, такого как MANE или устройство монтажа/редактирования, которое может обрабатывать закодированные видеоданные от кодера 20 видео. В некоторых случаях, объект обработки пост-обработки может быть примером сетевого объекта. В некоторых системах кодирования видео, объект 27 пост-обработки и кодер 20 видео могут быть частями отдельных устройств, тогда как в других случаях, функциональность, которая описывается в отношении объекта 27 пост-обработки, может быть выполнена точно тем же устройством, которое содержит кодер 20 видео. Объект 27 пост-обработки может быть видеоустройством. В некоторых примерах, объект 27 пост-обработки может быть точно таким же как устройство 34 генерирования файла на Фиг. 1.

[0112] Кодер 20 видео может выполнять внутреннее- и внешнее-кодирование блоков видео внутри слайсов видео. Внутреннее-кодирование основано на пространственном предсказании, чтобы сократить или удалить пространственную избыточность в видео внутри заданного кадра или картинки видео. Внешнее-кодирование основано на временном предсказании, чтобы сократить или удалить временную избыточность в видео внутри смежных кадров или картинок видеопоследовательности. Внутренний-режим (I режим) может относиться к любому из нескольких основанных на пространстве режимов сжатия. Внешние-режимы, такой как однонаправленное предсказание (P режим) или двунаправленное предсказание (B режим), могут относиться к любому из нескольких основанных на времени режимов сжатия.

[0113] В примере Фиг. 2, кодер 20 видео включает в себя модуль 37 разбиения, модуль 41 обработки предсказания, модуль 63 фильтра, память 64 опорной картинки, сумматор 50, модуль 52 обработки преобразования, модуль 54 квантования, и модуль 56 энтропийного кодирования. Модуль 41 обработки предсказания включает в себя модуль 42 оценки движения, модуль 44 компенсации движения, и модуль 46 обработки внутреннего предсказания.

Применительно к восстановлению блока видео, кодер 20 видео также включает в себя модуль 58 обратного квантования, модуль 60 обработки обратного преобразования, и сумматор 62. Модуль 63 фильтра предназначен для представления одного или более контурных фильтров, таких как фильтр устранения блочности, адаптивный контурный фильтр (ALF), и фильтр адаптивного к выборке смещения (SAO). Несмотря на то, что модуль 63 фильтра показан на Фиг. 2 как заключенный в контурном фильтре, в других конфигурациях, модуль 63 фильтра может быть реализован в качестве пост-контурного фильтра.

[0114] Память 35 видеоданных кодера 20 видео может хранить видеоданные, которые должны быть закодированы компонентами кодера 20 видео. Видеоданные, хранящиеся в памяти 35 видеоданных, могут быть получены, например, от источника 18 видео. Память 64 опорной картинки может быть памятью опорной картинки, которая хранит опорные видеоданные для использования при кодировании видеоданных кодером 20 видео, например, при режимах внутреннего- или внешнего-кодирования. Память 35 видеоданных и память 64 опорной картинки могут быть сформированы посредством любого из многообразия устройств памяти, таких как динамическая память с произвольным доступом (DRAM), включая синхронную DRAM (SDRAM), магниторезистивную RAM (MRAM), резистивную RAM (RRAM), или другие типы устройств памяти. Память 35 видеоданных и память 64 опорной картинки могут быть предоставлены одним и тем же устройством памяти или отдельными устройствами памяти. В различных примерах, память 35 видеоданных может быть на кристалле с другими компонентами кодера 20 видео, или вне кристалла по отношению к этим компонентам.

[0115] Как показано на Фиг. 2, кодер 20 видео принимает видеоданные, и модуль 37 разбиения разбивает данные на блоки видео. Данное разбиение также может включать в себя разбиение на слайсы, тайлы, или другие большие единицы, как впрочем и разбиение блока видео, например, в соответствии со структурой квадродерева из LCU и CU. Кодер 20 видео в целом иллюстрирует компоненты, которые кодируют блоки видео внутри слайса видео, который должен быть закодирован. Слайс может быть разделен на

несколько блоков видео (и возможно на наборы блоков видео, упоминаемых как тайлы). Модуль 41 обработки предсказания может выбирать один из множества возможных режимов кодирования, как например один из множества режимов внутреннего кодирования или один из множества режимов внешнего кодирования, для текущего блока видео на основании результатов ошибки (например, скорости кодирования и уровня искажения). Модуль 41 обработки предсказания может предоставлять результирующий внутренне- или внешне-кодированный блок сумматору 50, чтобы генерировать остаточные данные блока, и сумматору 62, чтобы восстанавливать закодированный блок для использования в качестве опорной картинки.

[0116] Модуль 46 обработки внутреннего предсказания внутри модуля 41 обработки предсказания может выполнять кодирование с внутренним предсказанием текущего блока видео по отношению к одному или более соседним блокам в том же самом кадре или слайсе, что и текущий блок, который должен быть закодирован, чтобы обеспечить пространственное сжатие. Модуль 42 оценки движения и модуль 44 компенсации движения внутри модуля 41 обработки предсказания выполняют кодирование с внешним предсказанием текущего блока видео по отношению к одному или более блокам предсказания в одной или более опорных картинках, чтобы обеспечить временное сжатие.

[0117] Модуль 42 оценки движения может быть выполнен с возможностью определения режима внешнего-предсказания для слайса видео в соответствии с предварительно определенным шаблоном для видеопоследовательности. Предварительно определенный шаблон может объявлять слайсы видео в последовательности в качестве P слайсов, B слайсов или GPB слайсов. Модуль 42 оценки движения и модуль 44 компенсации движения могут быть высоко интегрированными, но иллюстрируются отдельно в концептуальных целях. Оценка движения, выполняемая модулем 42 оценки движения, является процессом генерирования векторов движения, которые оценивают движение блоков видео. Вектор движения, например, может указывать перемещение PU блока видео внутри текущего кадра или картинки видео по отношению к блоку предсказания внутри

опорной картинки.

[0118] Блок предсказания является блоком который обнаружен как точно совпадающий с PU блока видео, который должен быть закодирован, в выражении пиксельной разности, которая может быть определена посредством суммы абсолютных разностей (SAD), суммы квадратичных разностей (SSD), или других метрик разности. В некоторых примерах, кодер 20 видео может вычислять значения суб-целочисленных позиций пикселя у опорных картинок, хранящихся в памяти 64 опорной картинки. Например, кодер 20 видео может интерполировать значения одной четвертой позиций пикселя, одной восьмой позиций пикселя, или других дробных позиций пикселя опорной картинки. Вследствие этого, модуль 42 оценки движения может выполнять поиск движения по отношению к полным позициям пикселя и дробным позициям пикселя и выводить вектор движения с дробной точностью пикселя.

[0119] Модуль 42 оценки движения вычисляет вектор движения для PU блока видео во внешне-кодированном слайсе посредством сравнения позиции PU с позицией блока предсказания опорной картинки. Опорная картинка может быть выбрана из первого списка опорных картинок (Список 0) или второго списка опорных картинок (Список 1), каждый из которых идентифицирует одну или более опорных картинок, хранящихся в памяти 64 опорной картинки. Модуль 42 оценки движения отправляет вычисленный вектор движения модулю 56 энтропийного кодирования и модулю 44 компенсации движения.

[0120] Компенсация движения, выполняемая модулем 44 компенсации движения, может задействовать привлечение или генерирование блока предсказания на основании вектора движения, определенного посредством оценки движения, возможно выполняя интерполяции до суб-пиксельной точности. По приему вектора движения для PU текущего блока видео, модуль 44 компенсации движения может определять местоположение блока предсказания, на который указывает вектор движения, в одном из списков опорных картинок. Кодер 20 видео может формировать блок остатка предсказания видео посредством вычитания значений пикселя блока предсказания из значений пикселя текущего блока видео, который

кодируется, формируя значения пиксельной разности. Значения пиксельной разности формируют остаточные данные для блока, и могут включать в себя компоненты разности как яркости, так и цветности. Сумматор 50 представляет собой компонент или компоненты, которые выполняют данную операцию вычитания. Модуль 44 компенсации движения также может генерировать элементы синтаксиса, ассоциированные с блоками видео и слайсом видео для использования декодером 30 видео при декодировании блоков видео у слайса видео.

[0121] Модуль 46 обработки внутреннего предсказания может внутренне-предсказывать текущий блок, в качестве альтернативы внешнему предсказанию, выполняемому модулем 42 оценки движения и модулем 44 компенсации движений, как описано выше. В частности, модуль 46 обработки внутреннего предсказания может определять режим внутреннего предсказания для использования при кодировании текущего блока. В некоторых примерах, модуль 46 обработки внутреннего предсказания может кодировать текущий блок, используя разнообразные режимы внутреннего предсказания, например, во время отдельных проходов кодирования, и модуль 46 обработки внутреннего предсказания (или модуль 40 выбора режима, в некоторых примерах) может выбирать пригодный режим внутреннего предсказания для использования из протестированных режимов. Например, модуль 46 обработки внутреннего предсказания может вычислять значения скорости-к-искажению, используя анализ скорости-к-искажению для разнообразных протестированных режимов внутреннего предсказания, и выбирать режим внутреннего предсказания с наилучшими характеристиками скорости-к-искажению из числа протестированных режимов. Анализ скорости-к-искажению главным образом определяет величину искажения (или ошибки) между закодированным блоком и исходным, незакодированным блоком, который был закодирован, чтобы создать закодированный блок, как впрочем и скорость передачи битов (т.е., количество битов), использованную для создания закодированного блока. Модуль 46 обработки внутреннего предсказания может вычислять отношения из искажений и скоростей для разных закодированных блоков, чтобы определять, какой режим внутреннего предсказания показывает

наилучшее значение скорости-к-искажению для блока.

[0122] В любом случае, после выбора режима внутреннего предсказания для блока, модуль 46 обработки внутреннего предсказания может предоставлять информацию, указывающую выбранный режим внутреннего предсказания для блока, модулю 56 энтропийного кодирования. Модуль 56 энтропийного кодирования может кодировать информацию, указывающую выбранный режим внутреннего предсказания в соответствии с методиками данного раскрытия. Кодер 20 видео может включать в передаваемый битовый поток данные конфигурации, которые могут включать в себя множество из таблиц индекса режима внутреннего предсказания и множество из модифицированных таблиц индекса режима внутреннего предсказания (также именуемых как таблицы соотнесения кодового слова), определения контекстов кодирования для разнообразных блоков, и указания наиболее вероятного режима внутреннего предсказания, таблицу индекса режима внутреннего предсказания, и модифицированную таблицу индекса режима внутреннего предсказания для использования для каждого из контекстов.

[0123] После того, как модуль 41 обработки предсказания генерирует блок предсказания для текущего блока видео через либо внешнее предсказание, либо внутренне предсказание, кодер 20 видео может формировать блок остатка предсказания видео посредством вычитания блока предсказания из текущего блока видео. Остаточные видеоданные в блоке остатка предсказания могут быть включены в одну или более TU и применены к модулю 52 обработки преобразования. Модуль 52 обработки преобразования преобразует остаточные видеоданные в остаточные коэффициенты преобразования, используя преобразование, такое как дискретное косинусное преобразование (DCT) или концептуально сходное преобразование. Блок 52 обработки преобразования может преобразовывать остаточные видеоданные из области пикселя в область преобразования, такую как частотная область.

[0124] Модуль 52 обработки преобразования может отправлять результирующие коэффициенты преобразования модулю 54 квантования. Модуль 54 квантования квантует коэффициенты преобразования, чтобы дополнительно сократить скорость передачи

битов. Процесс квантования может уменьшать битовую глубину, ассоциированную с некоторыми или всеми из коэффициентов. Степень квантования может модифицироваться посредством регулирования параметра квантования. В некоторых примерах, модуль 54 квантования затем может выполнять сканирование матрицы, включающей в себя квантованные коэффициенты преобразования. В качестве альтернативы, сканирование может выполнять модуль 56 энтропийного кодирования.

[0125] Вслед за квантованием, модуль 56 энтропийного кодирования может энтропийно кодировать элементы синтаксиса, представляющие квантованные коэффициенты преобразования. Например, модуль 56 энтропийного кодирования может выполнять контекстно-зависимое адаптивное кодирование с переменной длиной кодового слова (CAVLC), контекстно-зависимое адаптивное бинарное арифметическое кодирование (CABAC), основанное на синтаксисе контекстно-зависимое адаптивное бинарное арифметическое кодирование (SBAC), энтропийное кодирование с разбиением на интервал вероятности (PIPE) или другую методологию или методику энтропийного кодирования. Вслед за энтропийным кодированием посредством модуля 56 энтропийного кодирования, закодированный битовый поток может быть передан декодеру 30 видео, или заархивирован для дальнейшей передачи или извлечения декодером 30 видео. Модуль 56 энтропийного кодирования также может энтропийно кодировать векторы движения и другие элементы синтаксиса для текущего кодируемого слайса видео.

[0126] Модуль 58 обратного квантования и модуль 60 обработки обратного преобразования применяют обратное квантование и обратное преобразование, соответственно, чтобы восстанавливать блок остатка предсказания в области пикселя для дальнейшего использования в качестве опорного блока опорной картинки. Модуль 44 компенсации движения может вычислять опорный блок, посредством сложения блока остатка предсказания с блоком предсказания одной из опорных картинок в одном из списков опорных картинок. Модуль 44 компенсации движения также может применять один или более фильтров интерполяции к восстановленному блоку остатка предсказания, чтобы вычислять

суб-целочисленные значения пикселя для использования при оценке движения. Сумматор 62 складывает восстановленный блок остатка предсказания с блоком предсказания с компенсацией движения, созданным модулем 44 компенсации движения, чтобы создать опорный блок для сохранения в памяти 64 опорной картинки. Опорный блок может быть использован модулем 42 оценки движения и модулем 44 компенсации движения в качестве опорного блока для внешнего предсказания блока в последующем кадре или картинке видео.

[0127] Кодер 20 видео представляет собой пример кодера видео, выполненного с возможностью генерирования видеоданных, которые могут быть сохранены, используя методики формата файла, описываемые в данном раскрытии.

[0128] Фиг. 3 является структурной схемой, иллюстрирующей примерный декодер 30 видео, который может реализовывать методики, описываемые в данном раскрытии. Декодер 30 видео может быть выполнен с возможностью декодирования с одним видом, многовидовых, масштабируемых, 3D, или других типов видеоданных. В примере Фиг. 3, декодер 30 видео включает в себя модуль 80 энтропийного декодирования, модуль 81 обработки предсказания, модуль 86 обратного квантования, модуль 88 обработки обратного преобразования, сумматор 90, модуль 91 фильтра, и память 92 опорной картинки. Модуль 81 обработки предсказания включает в себя модуль 82 компенсации движения и модуль 84 обработки внутреннего предсказания. Декодер 30 видео может, в некоторых примерах, выполнять проход декодирования в целом обратно проходу кодирования, описанному в отношении кодера 20 видео с Фиг. 2.

[0129] Буфер 79 кодированной картинки (CPB) может принимать и сохранять закодированные видеоданные (например, единицы NAL) битового потока. Видеоданные, сохраненные в CPB 79, могут быть получены, например, из линии 16 связи, например, от локального источника видео, такого как камера, через связь проводной и беспроводной сети видеоданных, или посредством осуществления доступа к физическим носителям информации для хранения данных. CPB 79 может формировать память видеоданных, которая хранит закодированные видеоданные из закодированного битового потока видео. CPB 79 может быть памятью опорной картинки, которая

хранит опорные видеоданные для использования при декодировании видеоданных посредством декодера 30 видео, например, в режимах внутреннего- или внешнего-кодирования. СРВ 79 и память 92 опорной картинки могут быть сформированы посредством любого из многообразия устройств памяти, таких как динамическая память с произвольным доступом (DRAM), включая синхронную DRAM (SDRAM), магниторезистивную RAM (MRAM), резистивную RAM (RRAM), или другие типы устройств памяти. СРВ 79 и память 92 опорной картинки могут быть предоставлены одним и тем же устройством памяти или отдельными устройствами памяти. В разнообразных примерах, СРВ 79 может быть на кристалле с другими компонентами декодера 30 видео, или вне кристалла по отношению к этим компонентам.

[0130] Во время процесса декодирования, декодер 30 видео принимает закодированный битовый поток видео, который представляет собой блоки видео закодированного слайса видео и ассоциированные элементы синтаксиса, от кодера 20 видео. Декодер 30 видео может принимать закодированный битовый поток видео от сетевого объекта 29. Сетевой объект 29 может, например, быть сервером, MANE, средством редактирования/монтажа видео, или другим таким устройством, выполненным с возможностью реализации одной или более методик, описанных выше. Сетевой объект 29 может или может не включать в себя кодер видео, такой как кодер 20 видео. Некоторые из методик, описываемые в данном раскрытии, могут быть реализованы посредством сетевого объекта 29 до того, как сетевой объект 29 передает закодированный битовый поток видео декодеру 30 видео. В некоторых системах декодирования видео, сетевой объект 29 и декодер 30 видео могут быть частями отдельных устройств, тогда как в других случаях, функциональность, описываемая в отношении сетевого объекта 29, может быть выполнена посредством того же самого устройства, которое содержит декодер 30 видео. Сетевой объект 29 может рассматриваться в качестве видеоустройства. Кроме того, в некоторых примерах, сетевой объект 29 является устройством 34 генерирования файла на Фиг. 1.

[0131] Модуль 80 энтропийного декодирования декодера 30

видео энтропийно декодирует конкретные элементы синтаксиса битового потока, чтобы генерировать квантованные коэффициенты, векторы движения, и другие элементы синтаксиса. Модуль 80 энтропийного декодирования переадресовывает векторы движения и другие элементы синтаксиса модулю 81 обработки предсказания. Декодер 30 видео может принимать элементы синтаксиса на уровне слайса видео и/или на уровне блока видео.

[0132] Когда слайс видео является кодированным в качестве внутренне-кодированного (I) слайса, модуль 84 обработки внутреннего предсказания модуля 81 обработки предсказания может генерировать данные предсказания для блока видео текущего слайса видео на основании просигнализированного режима внутреннего предсказания и данных из ранее декодированных блоков текущего кадра или картинки. Когда кадр видео является кодированным в качестве внешне-кодированного (т.е., В, Р или GРВ) слайса, модуль 82 компенсации движения модуля 81 обработки предсказания создает блоки предсказания для блока видео текущего слайса видео на основании векторов движения и других элементов синтаксиса, принимаемых от модуля 80 энтропийного декодирования. Блоки предсказания могут быть созданы из одной из опорных картинок в одном из списков опорных картинок. Декодер 30 видео может создавать списки опорного кадра, Список 0 и Список 1, используя методики создания по умолчанию на основании опорных картинок, хранящихся в памяти 92 опорной картинки.

[0133] Модуль 82 компенсации движения определяет информацию предсказания для блока видео текущего слайса видео посредством синтаксического анализа векторов движения и других элементов синтаксиса, и использует информацию предсказания, чтобы создавать блоки предсказания для текущего декодируемого блока видео. Например, модуль 82 компенсации движения использует некоторые из принятых элементов синтаксиса, чтобы определять режим предсказания (например, внутреннее- или внешнее-предсказание), использованное, чтобы кодировать блоки видео слайса видео, тип слайса внешнего-предсказания (например, В слайс, Р слайс, или GРВ слайс), информацию создания для одного или более списков опорной картинки для слайса, векторы движения

для каждого внешне-закодированного блока видео у слайса, статус внешнего-предсказания для каждого внешне-кодированного блока видео у слайса, и другую информацию, чтобы декодировать блоки видео в текущем слайсе видео.

[0134] Модуль 82 компенсации движения также может выполнять интерполяцию на основании фильтров интерполяции. Модуль 82 компенсации движения может использовать фильтры интерполяции как используемые кодером 20 видео во время кодирования блоков видео, чтобы вычислять интерполированные значения для суб-целочисленных пикселей опорных блоков. В данном случае, модуль 82 компенсации движения может определять фильтры интерполяции, использованные кодером 20 видео, из принятых элементов синтаксиса, и может использовать фильтры интерполяции, чтобы создавать блоки предсказания.

[0135] Модуль 86 обратного квантования обратно квантует, т.е., де-квантует, квантованные коэффициенты преобразования, предоставленные в битовом потоке, и декодированные модулем 80 энтропийного декодирования. Процесс обратного квантования может включать в себя использование параметра квантования, вычисленного кодером 20 видео для каждого блока видео в слайсе видео, чтобы определять степень квантования и, подобным образом, степень обратного квантования, которая должна применяться. Модуль 88 обработки обратного преобразования применяет обратное преобразование, например, обратное DCT, обратное целочисленное преобразование, или концептуально сходный процесс обратного преобразования, к коэффициентам преобразования с тем, чтобы создать блоки остатка предсказания в области пикселя.

[0136] После того, как модуль 82 компенсации движения генерирует блок предсказания для текущего блока видео на основании векторов движения и других элементов синтаксиса, декодер 30 видео формирует декодированный блок видео посредством суммирования блоков остатка предсказания от модуля 88 обработки обратного преобразования с соответствующими блоками предсказания, сгенерированными модулем 82 компенсации движения. Сумматор 90 представляет собой компонент или компоненты, которые выполняют данную операцию суммирования. При желании, контурные

фильтры (либо в цикле кодирования, либо после цикла кодирования) также могут быть использованы, чтобы сглаживать переходы пикселя, или иным образом улучшать качество видео. Модуль 91 фильтра предназначен, чтобы представлять собой один или более контурных фильтров, таких как фильтр устранения блочности, адаптивный контурный фильтр (ALF), и фильтр адаптивного к выборке смещения (SAO). Несмотря на то, что модуль 91 фильтра показан на Фиг. 3 как заключенный в контурном фильтре, в других конфигурациях, модуль 91 фильтра может быть реализован в качестве пост-контурного фильтра. Декодированные блоки видео в заданном кадре или картинке затем сохраняются в памяти 92 опорной картинки, которая хранит опорные картинки, используемые для последующей компенсации движения. Память 92 опорной картинки также может хранить декодированное видео для последующего представления на устройстве отображения, таком как устройство 32 отображения на Фиг. 1.

[0137] Декодер 30 видео с Фиг. 3 представляет собой пример декодера видео, выполненного с возможностью декодирования видеоданных, которые могут быть сохранены, используя методики формата файла, описанные в данном раскрытии.

[0138] Фиг. 4 является структурной схемой, иллюстрирующей примерный набор устройств, которые формируют часть сети 100. В данном примере, сеть 100 включает в себя устройства 104А, 104В маршрутизации (устройства 104 маршрутизации) и устройство 106 перекодировки. Устройства 104 маршрутизации и устройство 106 перекодировки предназначены представлять собой небольшое количество устройств, которые могут формировать часть сети 100. Другие сетевые устройства, такие как коммутаторы, концентраторы, шлюзы, межсетевые экраны, мосты, и другие такие устройства также могут быть включены в сеть 100. Более того, дополнительные сетевые устройства могут быть предоставлены по пути сети между серверным устройством 102 и клиентским устройством 108. Серверное устройство 102 может соответствовать устройству-источнику 12 (Фиг. 1), тогда как клиентское устройство 108 может соответствовать устройству-получателю 14 (Фиг. 1), в некоторых примерах.

[0139] В целом, устройства 104 маршрутизации реализуют один или более протоколов маршрутизации, чтобы осуществлять обмен сетевыми данными через сеть 100. В некоторых примерах, устройства 104 маршрутизации могут быть выполнены с возможностью выполнения операций прокси-функции или кэширования. Вследствие этого, в некоторых примерах, устройства 104 маршрутизации могут упоминаться как прокси-устройства. В целом, устройства 104 маршрутизации исполняют протоколы маршрутизации, чтобы выявлять маршруты по сети 100. Посредством исполнения таких протоколов маршрутизации, устройство 104В маршрутизации может выявлять сетевой маршрут от себя самого до серверного устройства 102 через устройство 104А маршрутизации.

[0140] Методики данного раскрытия могут быть реализованы посредством сетевых устройств, таких как устройства 104 маршрутизации и устройство 106 перекодировки, но также могут быть реализованы посредством клиентского устройства 108. Таким образом, устройства 104 маршрутизации, устройство 106 перекодировки, и клиентское устройство 108 представляют собой примеры устройств, выполненных с возможностью выполнения методик данного раскрытия. Более того, устройства Фиг. 1, и кодер 20, иллюстрируемый на Фиг. 2, и декодер 30, иллюстрируемый на Фиг. 3, также являются примерами устройств, которые могут быть выполнены с возможностью выполнения одной или более методик данного раскрытия.

[0141] Фиг. 5А является концептуальной схемой, иллюстрирующей примерную структуру файла 300, в соответствии с одной или более методик данного раскрытия. В примере Фиг. 5А, файл 300 включает в себя контейнер 302 фильма и множество контейнеров 304 мультимедийных данных. Несмотря на то, что в примере Фиг. 5А они иллюстрируются, как находящиеся в одном и том же файле, в других примерах контейнер 302 фильма и контейнеры 304 мультимедийных данных могут находиться в отдельных файлах. Как указано выше, контейнер может быть объектно-ориентированным строительным блоком, который определяется уникальным идентификатором типа и длиной. Например, контейнер может быть элементарной структурой синтаксиса в

ISOBMFF, включающей в себя четырехзначный кодированный тип контейнера, счетчик байтов контейнера, и полезную нагрузку.

[0142] Контейнер 302 фильма может содержать метаданные дорожек файла 300. Каждая дорожка файла 300 может содержать непрерывный поток мультимедийных данных. Каждый из контейнеров 304 мультимедийных данных может включать в себя одну или более выборок 305. Каждая из выборок 305 может содержать единицу доступа аудио и видео. Как описано где-то в другом месте в данном раскрытии, каждая единица доступа может содержать несколько кодированных картинок при многовидовом кодировании (например, MV-HEVC и 3D-HEVC) и масштабируемом кодировании видео (например, SHVC). Например, единица доступа может включать в себя одну или более кодированных картинок для каждого слоя.

[0143] Кроме того, в примере Фиг. 5А, контейнер 302 фильма включает в себя контейнер 306 дорожки. Контейнер 306 дорожки может заключать метаданные для дорожки файла 300. В других примерах, контейнер 302 фильма может включать в себя несколько контейнеров дорожки для разных дорожек файла 300. Контейнер 306 дорожки включает в себя контейнер 307 мультимедиа. Контейнер 307 мультимедиа может содержать все объекты, которые объявляют информацию касательно мультимедийных данных внутри дорожки. Контейнер 307 мультимедиа может включать в себя контейнер 308 информации мультимедиа. Контейнер 308 информации мультимедиа может содержать все объекты, которые объявляют информацию характеристики у мультимедиа в дорожке. Контейнер 308 информации мультимедиа включает в себя контейнер 309 таблицы выборок. Контейнер 309 таблицы выборок может указывать характерные для выборки метаданные.

[0144] В примере Фиг. 5А, контейнер 309 таблицы выборок включает в себя контейнер 310 SampleToGroup и контейнер 312 SampleGroupDescription, и контейнер 312 SampleGroupDescription включает в себя контейнер 316 oinf. В других примерах, контейнер 309 таблицы выборок может включать в себя другие контейнеры в дополнение к контейнеру 310 SampleToGroup и контейнеру 312 SampleGroupDescription, и/или может включать в себя несколько контейнеров SampleToGroup и контейнеров SampleGroupDescription.

Контейнер 310 SampleToGroup может соотносить выборки (например, конкретные одни из выборок 305) с группой из выборок. Контейнер 312 SampleGroupDescription может указывать свойство, которое совместно используется выборками в группе из выборок (т.е., группе выборок). Кроме того, контейнер 309 таблицы выборки может включать в себя контейнеры 311 записи выборки. Каждый из контейнеров 311 записи выборки может соответствовать выборке в группе из выборок. В некоторых примерах, контейнеры 311 записи выборки являются экземплярами класса Доступной Произвольным образом Записи Выборки, который расширяет базовый класс описания группы выборок.

[0145] В соответствии с одной или более методик данного раскрытия, Контейнер 312 SampleGroupDescription может указывать то, что каждая выборка из группы выборок содержит, по меньшей мере, одну картинку IRAP. Таким образом, устройство 34 генерирования файла может генерировать файл, который содержит контейнер 306 дорожки, который содержит метаданные для дорожки в файле 300. Мультимедийные данные для дорожки содержат последовательность выборок 305. Каждая из выборок может быть единицей доступа видео многослойных видеоданных (например, видеоданных SHVC, MV-HEVC, или 3D-HEVC). Кроме того, как часть генерирования файла 300, устройство 34 генерирования файла может генерировать, в файле 300, дополнительный контейнер (например, контейнер 309 таблицы выборок), который документирует все выборки 305, содержащие, по меньшей мере, одну картинку IRAP. Другими словами, дополнительный контейнер идентифицирует все выборки 305, содержащие, по меньшей мере, одну картинку IRAP. В примере Фиг. 5А, дополнительный контейнер определяет группу выборок, которая документирует (например, идентифицирует) все выборки 305, содержащие, по меньшей мере, одну картинку IRAP. Другими словами, дополнительный контейнер указывает, что выборки 305, содержащие, по меньшей мере, одну картинку IRAP, принадлежат к группе выборок.

[0146] В соответствии с методиками данного раскрытия, Контейнер 312 SampleGroupDescription, может включать в себя контейнер 316 oinf. Контейнер oinf может хранить информацию

формата представления для каждой рабочей точки видеоданных. Информация формата представления может включать в себя одно или более из следующего: пространственное разрешение, битовую глубину, или формат цвета. Дополнительно, контейнер oinf может хранить счетчик слоя, который указывает количество необходимых слоев рабочей точки видеоданных. Контейнер oinf может дополнительно хранить информацию скорости передачи битов для каждой рабочей точки видеоданных. Таким образом, может не существовать потребности в сигнализации контейнера скорости передачи битов после контейнера конфигурации из-за сигнализации информации скорости передачи битов в контейнере oinf.

[0147] Дополнительно, может не существовать потребности в сохранении информации профиля, яруса, и уровня PTL, информации формата представления, и информации частоты кадров в записи конфигурации декодера формата файла. Вся прочая информация в записи конфигурации декодера может быть ассоциирована со всеми слоями видеоданных в дорожке. Запись конфигурации декодера для каждого слоя видеоданных может хранить информацию формата представления и информацию частоты кадров. Запись конфигурации декодера может хранить информацию параллелизма для каждого слоя видеоданных. Файлы, как правило, включают в себя только одну запись конфигурации декодера для дорожки, однако дорожка может содержать один или более слоев и одну или более рабочих точек. Информация PTL, информация формата представления, и информация частоты кадров может быть ассоциирована с либо каждым слоем, либо каждой ОР. Таким образом, в отличие от формата файла HEVC, который поддерживает только один слой, запись конфигурации декодера возможно не сможет должным образом обеспечивать данную ассоциацию для формата файла LHEVC, который поддерживает несколько слоев.

[0148] Запись конфигурации декодера может не хранить индекс рабочей точки в записи конфигурации декодера, где индекс рабочей точки ссылается на индекс рабочей точки, который документируется в контейнере информации о рабочей точке. Сохранение индекса рабочей точки в записи конфигурации декодера может предписывать устройству, воспроизводящему дорожку (т.е., ассоциированную с

записью конфигурации декодера), воспроизводить рабочую точку, на которую ссылается индекс рабочей точки. Тем не менее, может быть доступно больше рабочих точек. Удаление индекса рабочей точки может позволить устройству воспроизведения лучше идентифицировать все рабочие точки, поддерживаемые файлом. Запись конфигурации декодера может хранить список индексов рабочей точки, ассоциированных с дорожкой видеоданных. Запись конфигурации декодера может, например, быть извлечена из информации в контейнере 311 записи выборки на Фиг. 5А.

[0149] Запись конфигурации декодера хранит информацию, такую как размер поля длины, используемого в каждой выборке, чтобы указывать длину содержащихся в нем единиц NAL, как впрочем и наборов параметров, если храниться в записи выборки. Запись конфигурации декодера может, например, быть внешне оформлена (например, ее размер должен быть обеспечен структурой, которая ее содержит). Запись конфигурации декодера также может содержать поле версии, чтобы идентифицировать версию спецификации, которой следуют, с несовместимыми изменениями в записи, указываемыми посредством изменения номера версии. В противоположность, совместимые расширения данной записи могут не делать необходимым изменение кода версии конфигурации. Запись конфигурации декодера также может включать в себя значения для нескольких элементов синтаксиса HEVC, таких как `general_profile_space`, `general_tier_flag`, `general_profile_idc`, `general_profile_compatibility_flags`, `general_constraint_indicator_flags`, `general_level_idc`, `min_spatial_segmentation_idc`, `chroma_format_idc`, `bit_depth_luma_minus8` и `bit_depth_chroma_minus8`, которые определены в HEVC. Запись конфигурации декодера может содержать общую информацию, которая ассоциирует, с дорожкой, которая содержит запись конфигурации, количество временных суб-слоев, информацию сегментации, поддерживаемый тип параллелизма, и единицы NAL наборов параметров (например, VPS, SPS, PPS, SEI, и т.д.).

[0150] Кроме того, в соответствии с одной или более методик данного раскрытия, каждый из контейнеров 311 записи выборки

может включать в себя значение (например, `all_pics_are_IRAP`), указывающее, являются ли все кодированные картинки в соответствующей выборке картинками IRAP. В некоторых примерах, значение равное 1 указывает, что не все кодированные картинки в выборке являются картинками IRAP. Значение равное 0 указывает на то, что не требуется, чтобы каждая кодированная картинка в каждой выборке группы выборок являлась картинкой IRAP.

[0151] В некоторых примерах, когда не все кодированные картинки в конкретной выборке являются картинками IRAP, устройство 34 генерирования файла может включать, в один из контейнеров 311 записи выборки для конкретной выборки, значение (например, `num_IRAP_pics`), указывающее количество картинок IRAP в конкретной выборке. Дополнительно, устройство 34 генерирования файла может включать в себя, в записи выборки для конкретной выборки, значения, указывающие идентификаторы слоя у картинок IRAP в конкретной выборке. Устройство 34 генерирования файла также может включать, в запись выборки для конкретной выборки, значение, указывающее тип единицы NAL у единиц NAL VCL в картинках IRAP конкретной выборки.

[0152] Кроме того, в примере Фиг. 5А, контейнер 309 таблицы выборок включает в себя контейнер 314 информации суб-выборки. Несмотря на то, что пример Фиг. 5А показывает только один контейнер информации суб-выборки, контейнер 309 таблицы выборок может включать в себя несколько контейнеров информации суб-выборки. В целом, контейнер информации суб-выборки разработан, чтобы содержать информацию суб-выборки. Суб-выборка является непрерывным диапазоном байтов выборки. ISO/IEC 14496-12 указывает на то, что конкретное определение суб-выборки, должно обеспечиваться для заданной системы кодирования, такой как H.264/AVC или HEVC.

[0153] Раздел 8.4.8 в ISO/IEC 14496-15 указывает определение суб-выборки для HEVC. В частности, раздел 8.4.8 в ISO/IEC 14496-15 указывает, что для использования контейнера информации суб-выборки (8.7.7 в ISO/IEC 14496-12) в потоке HEVC, суб-выборка определяется на основе значения поля флагов контейнера информации суб-выборки. В соответствии с одной или

более методик данного раскрытия, если поле флагов в контейнере 314 информации суб-выборки равно 5, суб-выборка, соответствующая контейнеру 314 информации суб-выборки, содержит одну кодированную картинку и ассоциированные единицы NAL не-VCL. Ассоциированные единицы NAL не-VCL могут включать в себя единицы NAL, содержащие сообщения SEI, применимые к кодированной картинке, и единицы NAL, содержащие наборы параметров (например, VPS, SPS, PPS, и т.д.), применимые к кодированной картинке.

[0154] Таким образом, в одном примере, устройство 34 генерирования файла может генерировать файл (например, файл 300), который содержит контейнер дорожки (например, контейнер 306 дорожки), который содержит метаданные для дорожки в файле. В данном примере, мультимедийные данные для дорожки содержат последовательность выборок, каждая из выборок является единицей доступа видео у многослойных видеоданных (например, SHVC, MV-HEVC, или 3D-HEVC видеоданных). Кроме того, в данном примере, как часть устройства 34 генерирования файла, генерирующая файл, устройство 34 генерирования файла может генерировать, в файле, контейнер информации суб-выборки (например, контейнер 314 информации суб-выборки), который содержит флаги, которые указывают тип информации суб-выборки, заданный в контейнере информации суб-выборки. Когда флаги имеют конкретное значение, суб-выборка, соответствующая контейнеру информации суб-выборки, содержит в точности одну кодированную картинку и ноль или более единиц NAL не-VCL, ассоциированных с кодированной картинкой.

[0155] Кроме того, в соответствии с одной или более методик данного раскрытия, если поле флагов контейнера 314 информации суб-выборки равно 0, контейнер 314 информации суб-выборки дополнительно включает в себя значение DiscardableFlag, значение NoInterLayerPredFlag, значение LayerId, и значение TempId. Если поле флагов контейнера 314 информации суб-выборки равно 5, контейнер 314 информации суб-выборки может включать в себя значение DiscardableFlag, значение VclNalUnitType, значение LayerId, значение TempId, значение NoInterLayerPredFlag, значение SubLayerRefNalUnitFlag, и зарезервированное значение.

[0156] SubLayerRefNalUnitFlag равный 0 указывает на то, что

все единицы NAL в суб-выборке являются единицами NAL VCL у не-опорной картинки суб-слоя, как указано в ISO/IEC 23008-2 (т.е., HEVC). SubLayerRefNalUnitFlag равный 1 указывает на то, что все единицы NAL в суб-выборке являются единицами NAL VCL у опорной картинки суб-слоя, как указано в ISO/IEC 23008-2 (т.е., HEVC). Таким образом, когда устройство 34 генерирования файла генерирует контейнер 314 информации суб-выборки и флаги имеют конкретное значение (например, 5), устройство 34 генерирования файла включает, в контейнер 314 информации суб-выборки, дополнительный флаг, который указывает, являются ли все единицы NAL в суб-выборке единицами NAL VCL не-опорной картинки суб-слоя.

[0157] Значение DiscardableFlag указывает значение у значения discardable_flag единиц NAL VCL в суб-выборке. Как указано в разделе A.4 в ISO/IEC 14496-15, значение discardable_flag должно быть установлено в 1, если и только если все извлеченные или агрегированные единицы NAL имеют discardable_flag, установленный в 1, и устанавливается в 0 в противном случае. Единица NAL может иметь discardable_flag установленный в 1, если битовый поток, содержащий единицу NAL, может быть корректно декодирован без единицы NAL. Следовательно, единица NAL может быть «отбрасываемой», если битовый поток, содержащий единицу NAL, может быть корректно декодирован без единицы NAL. Все единицы NAL VCL в суб-выборке должны иметь одинаковое значение discardable_flag. Таким образом, когда устройство 34 генерирования файла генерирует контейнер 314 информации суб-выборки и флаги имеют конкретное значение (например, 5), устройство 34 генерирования файла включает, в контейнер 314 информации суб-выборки, дополнительный флаг (например, discardable_flag), который указывает, являются ли все из единиц NAL VCL суб-выборки отбрасываемыми.

[0158] Значение NoInterLayerPredFlag указывает значение у inter_layer_pred_enabled_flag у единиц NAL VCL в суб-выборке. inter_layer_pred_enabled_flag должен быть установлен в 1, если и только если все извлеченные или агрегированные единицы NAL VCL имеют inter_layer_pred_enabled_flag установленный в 1, и

устанавливается в 0 в противном случае. Все единицы NAL VCL в суб-выборке должны иметь одинаковое значение `inter_layer_pred_enabled_flag`. Следовательно, когда устройство 34 генерирования файла генерирует контейнер 314 информации суб-выборки и флаги имеют конкретное значение (например, 5), устройство 34 генерирования файла включает, в контейнер 314 информации суб-выборки, дополнительное значение (например, `inter_layer_pred_enabled_flag`), которое указывает является ли межслоевое предсказание разрешенным для всех единиц NAL VCL для суб-выборки.

[0159] `LayerId` указывает значение `nuh_layer_id` у единиц NAL в суб-выборке. Все единицы NAL в суб-выборке должны иметь одинаковое значение `nuh_layer_id`. Следовательно, когда устройство 34 генерирования файла генерирует контейнер 314 информации суб-выборки и флаги имеют конкретное значение (например, 5), устройство 34 генерирования файла включает, в контейнер 314 информации суб-выборки, дополнительное значение (например, `LayerId`), которое указывает идентификатор слоя каждой единицы NAL суб-выборки.

[0160] `TempId` указывает значение `TemporalId` у единиц NAL в суб-выборке. Все единицы NAL в суб-выборке должны иметь одинаковое значение `TemporalId`. Следовательно, когда устройство 34 генерирования файла генерирует контейнер 314 информации суб-выборки и флаги имеют конкретное значение (например, 5), устройство 34 генерирования файла включает, в контейнер 314 информации суб-выборки, дополнительное значение (например, `TempId`), которое указывает временной идентификатор каждой единицы NAL суб-выборки.

[0161] `VclNalUnitType` указывает значение элемента синтаксиса `nal_unit_type` у единиц NAL VCL в суб-выборке. Элемент синтаксиса `nal_unit_type` является элементом синтаксиса в заголовке единицы NAL у единицы NAL. Элемент синтаксиса `nal_unit_type` указывает тип RBSP, которая содержится в единице NAL. Все единицы NAL VCL с `nal_unit_type` в суб-выборке должны иметь одинаковое значение `nal_unit_type`. Следовательно, когда устройство 34 генерирования файла генерирует контейнер 314

информации суб-выборки и флаги имеют конкретное значение (например, 5), устройство 34 генерирования файла включает, в контейнер 314 информации суб-выборки, дополнительное значение (например, `VclNalUnitType`), которое указывает тип единицы NAL у единиц NAL VCL суб-выборки.

[0162] Фиг. 5B является концептуальной схемой, иллюстрирующей альтернативную примерную структуру файла 300, в соответствии с одной или более методик данного раскрытия. В примере Фиг. 5B, вместо включения контейнера 316 `oinf` в контейнер 312 описания группы выборок, как показано на Фиг. 5A, контейнер 316 `oinf` включается в контейнер 308 информации мультимедиа в качестве контейнера отдельного от контейнера 30 таблицы выборок. Содержимое и функция разнообразных контейнеров на Фиг. 5B могут во всем остальном быть точно такими же, как описанные в отношении Фиг. 5A.

[0163] Фиг. 6 является концептуальной схемой, иллюстрирующей примерную структуру файла 300, в соответствии с одной или более методик данного раскрытия. Как указано в разделе 8.4.9 в ISO/IEC 14496-15, HEVC допускает выборки формата файла, которые используются только для ссылки, а не вывода. Например, HEVC допускает неотображаемую опорную картинку в видео.

[0164] Кроме того, раздел 8.4.9 в ISO/IEC 14496-15 указывает, что когда любая такая не-выводимая выборка присутствует в дорожке, файл должен быть ограничен следующим образом.

1. Не-выводимой выборке должно быть задано время композиции, которое находится вне диапазона времени выборок, которые выводятся.

2. Список редактирования должен быть использован, чтобы исключать времена композиции не-выводимых выборок.

3. Когда дорожка включает в себя `CompositionOffsetBox` ('ctts'),

- a. должна быть использована версия 1 `CompositionOffsetBox`,

- b. значение `sample_offset` должно быть установлено равным -2^{31} для не-выводимой выборки,

c. CompositionToDecodeBox ('cslg') должен содержаться в SampleTableBox ('stbl') у дорожки, и

d. когда CompositionToDecodeBox присутствует для дорожки, значение поля leastDecodeToDisplayDelta в контейнере должно быть равно наименьшему смещению композиции в CompositionOffsetBox за исключением значений sample_offset для не-выводимых выборок.

ПРИМЕЧАНИЕ: Таким образом, leastDecodeToDisplayDelta больше -2^{31} .

[0165] Как указано в ISO/IEC 14496-12, CompositionOffsetBox обеспечивает смещение между временем декодирования и временем композиции. CompositionOffsetBox включает в себя набор значений sample_offset. Каждое из значений sample_offset является неотрицательным целым числом, которое задает смещение между временем композиции и временем декодирования. Время композиции относится к времени, в которое выборка должна быть выведена. Время декодирования относится к времени, в которое выборка должна быть декодирована.

[0166] Как указано выше, единица NAL кодированного слайса может включать в себя заголовок сегмента слайса. Заголовок сегмента слайса может быть частью сегмента кодированного слайса и может содержать элементы данных, которые относятся к первой или всем STU в сегменте слайса. В HEVC, заголовок сегмента слайса включает в себя элемент синтаксиса pic_output_flag. В целом, элемент синтаксиса pic_output_flag включается в первый заголовок сегмента слайса у слайса картинки. Следовательно, данное раскрытие может относиться к pic_output_flag первого заголовка сегмента слайса у слайса картинки как к pic_output_flag картинки.

[0167] Как указано в разделе 7.4.7.1 в HEVC WD, элемент синтаксиса pic_output_flag оказывает влияние на процессы вывода и удаления декодированной картинки как указывается в Приложении C у HEVC WD. В целом, если элемент синтаксиса pic_output_flag у заголовка сегмента слайса для сегмента слайса равен 1, выводится картинка, которая включает в себя слайс, соответствующий заголовку сегмента слайса. В противном случае, если элемент

синтаксиса `pic_output_flag` заголовка сегмента слайса для сегмента слайса равен 0, картинка, которая включает в себя слайс, соответствующий заголовку сегмента слайса, может быть декодирована для использования в качестве опорной картинки, но не выводиться.

[0168] В соответствии с одной или более методик данного раскрытия, ссылки в разделе 8.4.9 в ISO/IEC 14496-15 на HEVC могут быть заменены соответствующими ссылками на SHVC, MV-HEVC, или 3D-HEVC. Кроме того, в соответствии с одной или более методик данного раскрытия, когда единица доступа содержит некоторые кодированные картинки, которые имеют `pic_output_flag` равный 1 и некоторые другие картинки, которые имеют `pic_output_flag` равный 0, по меньшей мере две дорожки должны быть использованы для сохранения потока. Применительно к каждой соответствующей одной из дорожек, все кодированные картинки в каждой выборке соответствующей дорожки имеют одинаковое значение `pic_output_flag`. Следовательно, все кодированные картинки в первой одной из дорожек имеют `pic_output_flag` равный 0 и все кодированные картинки во второй одной из дорожек имеют `pic_output_flag` равный 1.

[0169] Соответственно, в примере Фиг. 6, устройство 34 генерирования файла может генерировать файл 400. Сходно с файлом 300 в примере на Фиг. 5А, файл 400 включает в себя контейнер 402 фильма и один или более контейнеров 404 мультимедийных данных. Каждый из контейнеров 404 мультимедийных данных может соответствовать разной дорожке файла 400. Контейнер 402 фильма может содержать метаданные для дорожек файла 400. Каждая дорожка файла 400 может содержать непрерывный поток мультимедийных данных. Каждый из контейнеров 404 мультимедийных данных может включать в себя одну или более выборок 405. Каждая из выборок 405 может содержать единицу доступа аудио или видео.

[0170] Как указано выше, в некоторых примерах, когда единица доступа содержит некоторые кодированные картинки, которые имеют `pic_output_flag` равный 1 и некоторые другие кодированные картинки, которые имеют `pic_output_flag` равный 0, по меньшей мере, две дорожки должны быть использованы, чтобы

хранить поток. Соответственно, в примере Фиг. 6, контейнер 402 фильма включает в себя контейнер 406 дорожки и контейнер 408 дорожки. Каждый из контейнеров 406 и 408 дорожки включает метаданные для отличной дорожки файла 400. Например, контейнер 406 дорожки может заключать метаданные для дорожки, обладающей кодированными картинками с `pic_output_flag` равным 0, и без картинок с `pic_output_flag` равным 1. Контейнер 408 дорожки может заключать метаданные для дорожки, обладающей кодированными картинками с `pic_output_flag` равным 1, и без картинок с `pic_output_flag` равным 0.

[0171] Таким образом, в одном примере, устройство 34 генерирования файла может генерировать файл (например, файл 400), который содержит контейнер мультимедийных данных (например, контейнер 404 мультимедийных данных), который заключает (например, содержит) контент мультимедиа. Контент мультимедиа содержит последовательность выборок (например, выборки 405). Каждая из выборок может быть единицей доступа многослойных видеоданных. В данном примере, когда устройство 34 генерирования файла генерирует файл, в ответ на определение того, что, по меньшей мере, одна единица доступа битового потока включает в себя кодированную картинку, которая имеет флаг вывода картинки равный 1 и кодированную картинку, которая имеет флаг вывода картинки равный 0, устройство 34 генерирования файла может использовать, по меньшей мере, две дорожки, чтобы хранить битовый поток в файле. Для каждой соответствующей дорожки из, по меньшей мере, двух дорожек, все кодированные картинки в каждой выборке соответствующей дорожки имеют одинаковое значение флага вывода картинки. Допускается вывод картинок с флагами вывода картинки равными 1, и допускается использование картинок с флагами вывода картинки равными 0 в качестве опорных картинок, но не допускается их вывод.

[0172] Фиг. 7 является блок-схемой, иллюстрирующей пример функционирования устройства 34 генерирования файла, в соответствии с одной или более методик данного раскрытия. Функционирование на Фиг. 7, наряду с функционированиями, которые иллюстрируются на других блок-схемах данного раскрытия, являются

примерами. Другие примерные функционирования в соответствии с методиками данного раскрытия могут включать в себя больше, меньше, или отличные действия.

[0173] В примере на Фиг. 7, устройство 34 генерирования файла генерирует файл. Как часть генерирования файла, устройство 34 генерирования файла получает (170) многослойные видеоданные и сохраняет (172) многослойные видеоданные в формате файла. Устройство 34 генерирования файла сохраняет (174) информацию формата представления для каждой рабочей точки многослойных видеоданных в контейнере oinf формата файла. Устройство 34 генерирования файла генерирует (176) файл видеоданных, отформатированный в соответствии с первым форматом. Информация формата представления может включать в себя одно или более из следующего: пространственное разрешение, битовую глубину, или формат цвета. Устройство 34 генерирования файла может дополнительно или в качестве альтернативы сохранять информацию скорости передачи битов для каждой рабочей точки многослойных видеоданных в контейнере oinf формата файла и/или может не сигнализировать контейнер скорости передачи битов после контейнера конфигурации формата файла. Устройство 34 генерирования файла может дополнительно или в качестве альтернативы не сохранять информацию профиля, яруса, и уровня (PTL), информацию формата представления, и информацию частоты кадров в записи конфигурации декодера формата файла и ассоциировать всю прочую информацию в записи конфигурации декодера со всеми слоями многослойных видеоданных в дорожке. Устройство 34 генерирования файла может дополнительно или в качестве альтернативы сохранять счетчик слоя в контейнере oinf формата файла, при этом счетчик слоя указывает количество необходимых слоев рабочей точки многослойных видеоданных.

[0174] Контейнер oinf может быть включен в контейнер информации мультимедиа, и контейнер oinf может быть включен в контейнер описания группы выборок. Контейнер описания группы выборок может быть включен в контейнер таблицы выборок, и контейнер таблицы выборок может быть включен в контейнер информации мультимедиа.

[0175] Устройство генерирования файла может сохранять информацию формата представления и информацию частоты кадров в записи конфигурации декодера для каждого слоя многослойных видеоданных. Устройство 34 генерирования файла может дополнительно или в качестве альтернативы сохранять информацию параллелизма в записи конфигурации декодера для каждого слоя многослойных видеоданных. Устройство 34 генерирования файла может не сохранять индекс рабочей точки в записи конфигурации декодера формата файла. Устройство 34 генерирования файла может дополнительно или в качестве альтернативы сохранять список индексов рабочей точки, ассоциированный с дорожкой многослойных видеоданных, в записи конфигурации декодера формата файла.

[0176] Фиг. 8 является блок-схемой, иллюстрирующей пример функционирования устройства чтения файла, такого как устройство-получатель 14, объект 27 пост-обработки, или сетевой объект 29. Функционирование на Фиг. 8, наряду с функционированиями, которые иллюстрируются на других блок-схемах данного раскрытия, являются примерами. Другие примеры функционирований в соответствии с методиками данного раскрытия могут включать в себя больше, меньше, или другие действия.

[0177] В примере на Фиг. 8, устройство чтения файла получает (180) файл многослойных видеоданных, отформатированный в соответствии с форматом файла. Устройство чтения файла, применительно к формату файла, определяет (182) информацию формата представления для каждой рабочей точки многослойных видеоданных в контейнере oinf для формата файла. Устройство чтения файла, возможно совместно с декодером видео, таким как декодер 30 видео, декодирует (184) многослойные видеоданные на основании определенной информации формата представления.

[0178] В одном или более примерах, описанные функции могут быть реализованы в аппаратном обеспечении, программном обеспечении, встроенном программном обеспечении, или любом их сочетании. При реализации в программном обеспечении, функции могут быть сохранены на или переданы через, в качестве одной или более инструкций или кода, машиночитаемый носитель информации и исполнены посредством основанного на аппаратном обеспечении

модуля обработки. Машиночитаемые носители информации могут включать в себя машиночитаемые запоминающие носители информации, которые соответствуют вещественному носителю информации, такому как носители информации для хранения данных, или средства связи, включающие в себя любые средства, которые способствуют переносу компьютерной программы из одного места в другое, например, в соответствии с протоколом связи. Таким образом, машиночитаемые носители информации главным образом соответствуют (1) вещественным машиночитаемым запоминающим носителям информации, которые не являются временными или (2) средствам связи, таким как сигнал или несущая волна. Носители информации для хранения данных могут быть любыми доступными носителями информации, доступ к которым может быть осуществлен посредством одного или более компьютеров или одного или более процессоров, чтобы извлекать инструкции, код и/или структуры данных для реализации методик, описываемых в данном раскрытии. Компьютерный программный продукт может включать в себя машиночитаемый носитель информации.

[0179] В качестве примера, а не ограничения, такие машиночитаемые запоминающие носители информации могут быть выполнены в виде RAM, ROM, EEPROM, CD-ROM или другого хранилища на оптическом диске, хранилища на магнитном диске, или других магнитных запоминающих устройств, флэш памяти, или любого другого носителя информации, который может быть использован, чтобы хранить требуемый программный код в форме инструкций или структур данных и доступ к которому может быть осуществлен посредством компьютера. Также, любое соединение правильно называть машиночитаемым носителем информации. Например, если инструкции передаются с web-сайта, сервера, или другого удаленного источника, используя коаксиальный кабель, оптоволоконный кабель, витую пару, цифровую абонентскую линию (DSL), или беспроводные технологии, такие как инфракрасные, радиосвязи, и микроволновые, тогда коаксиальный кабель, оптоволоконный кабель, витая пара, DSL, или беспроводные технологии, такие как инфракрасные, радиосвязи, и микроволновые, включены в определение носителя информации. Тем не менее,

следует понимать, что машиночитаемые запоминающие носители информации и носители информации для хранения данных не включают в себя соединения, несущие волны, сигналы, или другие временные носители информации, а вместо этого направлены на не временные, вещественные запоминающие носители информации. Используемый в данном документе магнитный диск и оптический диск включают в себя компакт-диск (CD), лазерный диск, оптический диск, цифровой универсальный диск (DVD), гибкий диск и Blu-ray диск, где магнитные диски обычно воспроизводят данные магнитным образом, тогда как оптические диски воспроизводят данные оптическим образом с помощью лазеров. Сочетания вышеприведенного также должно быть включено в объем машиночитаемых носителей информации.

[0180] Инструкции могут быть исполнены одним или более процессоров, таких как один или более цифровых сигнальных процессоров (DSP), микропроцессоров общего назначения, проблемно ориентированных интегральных микросхем (ASIC), программируемых вентильных матриц (FPGA), или других эквивалентных интегральных или дискретных логических схем. Соответственно, понятие «процессор», используемое в данном документе, может относиться к любой вышеупомянутой структуре или любой другой структуре, пригодной для реализации методик, описываемых в данном документе. В дополнение, в некоторых аспектах, функциональность, описываемая в данном документе может быть обеспечена в рамках предназначенных модулей аппаратного обеспечения и/или программного обеспечения, выполненных с возможностью кодирования и декодирования, или встроенных в объединенный кодек. Также, методики могут быть полностью реализованы в одной или более схем или логических элементов.

[0181] Методики данного раскрытия могут быть реализованы в широком многообразии устройств и аппаратур, включая беспроводную телефонную трубку, интегральную микросхему (IC) или набор IC (например, набор микросхем). Разнообразные компоненты, модули, или блоки описываются в данном раскрытии, чтобы подчеркнуть функциональные аспекты устройств, выполненных с возможностью выполнения раскрываемых методик, но не обязательно требуют

реализации посредством разных модулей аппаратного обеспечения. Наоборот, как описано выше, разнообразные модули могут быть объединены в модуль аппаратного обеспечения кодека или обеспечены посредством совокупности взаимодействующих модулей аппаратного обеспечения, включая один или более процессоров, как описано выше, совместно с пригодным программным обеспечением и/или встроенным программным обеспечением.

[0182] Были описаны разнообразные примеры. Эти и прочие примеры находятся в рамках объема нижеследующей формулы изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ обработки многослойных видеоданных, при этом способ содержит этапы, на которых:

получают многослойные видеоданные;

сохраняют многослойные видеоданные в формате файла;

сохраняют информацию формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и

генерируют файл видеоданных, отформатированный в соответствии с форматом файла.

2. Способ по п. 1, в котором информация формата представления содержит одно или более из следующего: пространственное разрешение, битовую глубину, или формат цвета.

3. Способ по п. 1, дополнительно содержащий этапы, на которых:

сохраняют информацию скорости передачи битов для каждой рабочей точки многослойных видеоданных в контейнере oinf формата файла; и

не сигнализируют контейнер скорости передачи битов после контейнера конфигурации формата файла.

4. Способ по п. 1, дополнительно содержащий этапы, на которых:

не сохраняют информацию профиля, яруса, и уровня (PTL), информацию формата представления, и информацию частоты кадров в записи конфигурации декодера формата файла; и

ассоциируют всю другую информацию в записи конфигурации декодера со всеми слоями многослойных видеоданных в дорожке.

5. Способ по п. 1, дополнительно содержащий этап, на котором:

сохраняют информацию формата представления и информацию частоты кадров в записи конфигурации декодера для каждого слоя многослойных видеоданных.

6. Способ по п. 5, дополнительно содержащий этап, на котором:

сохраняют информацию параллелизма в записи конфигурации декодера для каждого слоя многослойных видеоданных.

7. Способ по п. 1, дополнительно содержащий этап, на котором:

не сохраняют индекс рабочей точки в записи конфигурации декодера формата файла.

8. Способ по п. 1, дополнительно содержащий этап, на котором:

сохраняют список индексов рабочей точки, ассоциированных с дорожкой многослойных видеоданных в записи конфигурации декодера формата файла.

9. Способ по п. 1, дополнительно содержащий этап, на котором:

сохраняют счетчик слоя в контейнере oinf формата файла, при этом счетчик слоя указывает количество необходимых слоев рабочей точки многослойных видеоданных.

10. Способ по п. 1, в котором контейнер oinf включается в контейнер информации мультимедиа.

11. Способ по п. 10, в котором контейнер oinf дополнительно включается в контейнер описания группы выборок, при этом контейнер описания группы выборок включается в контейнер таблицы выборок, и при этом контейнер таблицы выборок включается в контейнер информации мультимедиа.

12. Способ по п. 1, в котором каждая рабочая точка многослойных видеоданных содержит, соответственно, битовый поток, созданный из другого битового потока посредством операции в виде процесса извлечения битового суб-потoka в отношении другого битового потока.

13. Видеоустройство для обработки многослойных видеоданных, при этом устройство содержит:

носитель информации для хранения данных, выполненный с возможностью сохранения многослойных видеоданных; и

один или более процессоров, выполненных с возможностью:

получения многослойных видеоданных;

сохранения многослойных видеоданных в формате файла;

сохранения информации формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и

генерирования файла видеоданных, отформатированного в соответствии с форматом файла.

14. Устройство по п. 13, в котором информация формата представления содержит одно или более из следующего: пространственное разрешение, битовую глубину, или формат цвета.

15. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения информации скорости передачи битов для каждой рабочей точки многослойных видеоданных в контейнере oinf формата файла; и

не сигнализации контейнера скорости передачи битов после контейнера конфигурации формата файла.

16. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

не сохранения информации профиля, яруса, и уровня (PTL), информации формата представления, и информации частоты кадров в записи конфигурации декодера формата файла; и

ассоциации всей другой информации в записи конфигурации декодера со всеми слоями многослойных видеоданных в дорожке.

17. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения информации формата представления и информации частоты кадров в записи конфигурации декодера для каждого слоя многослойных видеоданных.

18. Устройство по п. 17, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения информации параллелизма в записи конфигурации декодера для каждого слоя многослойных видеоданных.

19. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

не сохранения индекса рабочей точки в записи конфигурации декодера формата файла.

20. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения списка индексов рабочей точки, ассоциированных с дорожкой многослойных видеоданных в записи конфигурации декодера

формата файла.

21. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения счетчика слоя в контейнере oinf формата файла, при этом счетчик слоя указывает количество необходимых слоев рабочей точки многослойных видеоданных.

22. Устройство по п. 13, в котором контейнер oinf включается в контейнер информации мультимедиа.

23. Устройство по п. 22, в котором контейнер oinf дополнительно включается в контейнер описания группы выборок, при этом контейнер описания группы выборок включается в контейнер таблицы выборок, и при этом контейнер таблицы выборок включается в контейнер информации мультимедиа.

24. Устройство по п. 13, в котором каждая рабочая точка многослойных видеоданных содержит, соответственно, битовый поток, созданный из другого битового потока посредством операции в виде процесса извлечения битового суб-потока в отношении другого битового потока.

25. Видеоустройство для обработки многослойных видеоданных, при этом устройство содержит:

средство для получения многослойных видеоданных;

средство для сохранения многослойных видеоданных в формате файла;

средство для сохранения информации формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и

средство для генерирования файла видеоданных, отформатированного в соответствии с форматом файла.

26. Устройство по п. 25, в котором контейнер oinf включается в контейнер информации мультимедиа.

27. Устройство по п. 26, в котором контейнер oinf дополнительно включается в контейнер описания группы выборок, при этом контейнер описания группы выборок включается в контейнер таблицы выборок, и при этом контейнер таблицы выборок включается в контейнер информации мультимедиа.

28. Машиночитаемый запоминающий носитель информации,

хранящий инструкции, которые, когда исполняются, предписывают одному или более процессорам:

получать многослойные видеоданные;

сохранять многослойные видеоданные в формате файла;

сохранять информацию формата представления для каждой рабочей точки многослойных видеоданных в контейнере информации рабочих точек (oinf) для формата файла; и

генерировать файл видеоданных, отформатированный в соответствии с форматом файла.

29. Машиночитаемый запоминающий носитель информации по п. 28, в котором контейнер oinf включается в контейнер информации мультимедиа.

30. Машиночитаемый запоминающий носитель информации по п. 29, в котором контейнер oinf дополнительно включается в контейнер описания группы выборок, при этом контейнер описания группы выборок включается в контейнер таблицы выборок, и при этом контейнер таблицы выборок включается в контейнер информации мультимедиа.

По доверенности

ИЗМЕНЕННАЯ ФОРМУЛА ПО СТ.34 РСТ

1. Способ обработки многослойных видеоданных, при этом способ содержит этапы, на которых:

получают многослойные видеоданные, содержащие более чем одну рабочую точку;

сохраняют многослойные видеоданные в формате файла, при этом формат файла включает в себя контейнер информации рабочих точек (oinf), который идентифицирует рабочие точки, включенные в многослойные видеоданные;

сохраняют информацию формата представления для каждой рабочей точки многослойных видеоданных в контейнере oinf; и

генерируют файл видеоданных, отформатированный в соответствии с форматом файла.

2. Способ по п. 1, в котором информация формата представления содержит одно или более из следующего: пространственное разрешение, битовую глубину, или формат цвета.

3. Способ по п. 1, дополнительно содержащий этапы, на которых: сохраняют информацию скорости передачи битов для каждой рабочей точки многослойных видеоданных в контейнере oinf формата файла; и

не сигнализируют контейнер скорости передачи битов после контейнера конфигурации формата файла.

4. Способ по п. 1, дополнительно содержащий этапы, на которых: не сохраняют информацию профиля, яруса, и уровня (PTL), информацию формата представления, и информацию частоты кадров в записи конфигурации декодера формата файла; и

ассоциируют всю другую информацию в записи конфигурации декодера со всеми слоями многослойных видеоданных в дорожке.

5. Способ по п. 1, дополнительно содержащий этап, на котором: сохраняют информацию формата представления и информацию частоты кадров в записи конфигурации декодера для каждого слоя многослойных видеоданных.

6. Способ по п. 5, дополнительно содержащий этап, на котором: сохраняют информацию параллелизма в записи конфигурации декодера для каждого слоя многослойных видеоданных.

7. Способ по п. 1, дополнительно содержащий этап, на котором:

не сохраняют индекс рабочей точки в записи конфигурации декодера формата файла.

8. Способ по п. 1, дополнительно содержащий этап, на котором: сохраняют список индексов рабочей точки, ассоциированных с дорожкой многослойных видеоданных в записи конфигурации декодера формата файла.

9. Способ по п. 1, дополнительно содержащий этап, на котором: сохраняют счетчик слоя в контейнере oinf формата файла, при этом счетчик слоя указывает количество необходимых слоев рабочей точки многослойных видеоданных.

10. Способ по п. 1, в котором контейнер oinf включается в контейнер информации мультимедиа.

11. Способ по п. 10, в котором контейнер oinf дополнительно включается в контейнер описания группы выборок, при этом контейнер описания группы выборок включается в контейнер таблицы выборок, и при этом контейнер таблицы выборок включается в контейнер информации мультимедиа.

12. Способ по п. 1, в котором каждая рабочая точка многослойных видеоданных содержит, соответственно, битовый поток, созданный из другого битового потока посредством операции в виде процесса извлечения битового суб-потoka в отношении другого битового потока.

13. Видеоустройство для обработки многослойных видеоданных, при этом устройство содержит:

носитель информации для хранения данных, выполненный с возможностью сохранения многослойных видеоданных; и

один или более процессоров, выполненных с возможностью:

получения многослойных видеоданных, содержащих более чем одну рабочую точку;

сохранения многослойных видеоданных в формате файла, при этом формат файла включает в себя контейнер информации рабочих точек (oinf), который идентифицирует рабочие точки, включенные в многослойные видеоданные;

сохранения информации формата представления для каждой рабочей точки многослойных видеоданных в контейнере oinf; и

генерирования файла видеоданных, отформатированного в соответствии с форматом файла.

14. Устройство по п. 13, в котором информация формата представления содержит одно или более из следующего: пространственное разрешение, битовую глубину, или формат цвета.

15. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения информации скорости передачи битов для каждой рабочей точки многослойных видеоданных в контейнере oinf формата файла; и

не сигнализации контейнера скорости передачи битов после контейнера конфигурации формата файла.

16. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

не сохранения информации профиля, яруса, и уровня (PTL), информации формата представления, и информации частоты кадров в записи конфигурации декодера формата файла; и

ассоциации всей другой информации в записи конфигурации декодера со всеми слоями многослойных видеоданных в дорожке.

17. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения информации формата представления и информации частоты кадров в записи конфигурации декодера для каждого слоя многослойных видеоданных.

18. Устройство по п. 17, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения информации параллелизма в записи конфигурации декодера для каждого слоя многослойных видеоданных.

19. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

не сохранения индекса рабочей точки в записи конфигурации декодера формата файла.

20. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения списка индексов рабочей точки, ассоциированных с

дорожкой многослойных видеоданных в записи конфигурации декодера формата файла.

21. Устройство по п. 13, в котором один или более процессоров дополнительно выполнены с возможностью:

сохранения счетчика слоя в контейнере oinf формата файла, при этом счетчик слоя указывает количество необходимых слоев рабочей точки многослойных видеоданных.

22. Устройство по п. 13, в котором контейнер oinf включается в контейнер информации мультимедиа.

23. Устройство по п. 22, в котором контейнер oinf дополнительно включается в контейнер описания группы выборок, при этом контейнер описания группы выборок включается в контейнер таблицы выборок, и при этом контейнер таблицы выборок включается в контейнер информации мультимедиа.

24. Устройство по п. 13, в котором каждая рабочая точка многослойных видеоданных содержит, соответственно, битовый поток, созданный из другого битового потока посредством операции в виде процесса извлечения битового суб-потока в отношении другого битового потока.

25. Видеоустройство для обработки многослойных видеоданных, при этом устройство содержит:

средство для получения многослойных видеоданных, содержащих более чем одну рабочую точку;

средство для сохранения многослойных видеоданных в формате файла, при этом формат файла включает в себя контейнер информации рабочих точек (oinf), который идентифицирует рабочие точки, включенные в многослойные видеоданные;

средство для сохранения информации формата представления для каждой рабочей точки многослойных видеоданных в контейнере oinf;
и

средство для генерирования файла видеоданных, отформатированного в соответствии с форматом файла.

26. Устройство по п. 25, в котором контейнер oinf включается в контейнер информации мультимедиа.

27. Устройство по п. 26, в котором контейнер oinf

дополнительно включается в контейнер описания группы выборок, при этом контейнер описания группы выборок включается в контейнер таблицы выборок, и при этом контейнер таблицы выборок включается в контейнер информации мультимедиа.

28. Машиночитаемый запоминающий носитель информации, хранящий инструкции, которые, когда исполняются, предписывают одному или более процессорам:

получать многослойные видеоданные, содержащие более чем одну рабочую точку;

сохранять многослойные видеоданные в формате файла, при этом формат файла включает в себя контейнер информации рабочих точек (oinf), который идентифицирует рабочие точки, включенные в многослойные видеоданные;

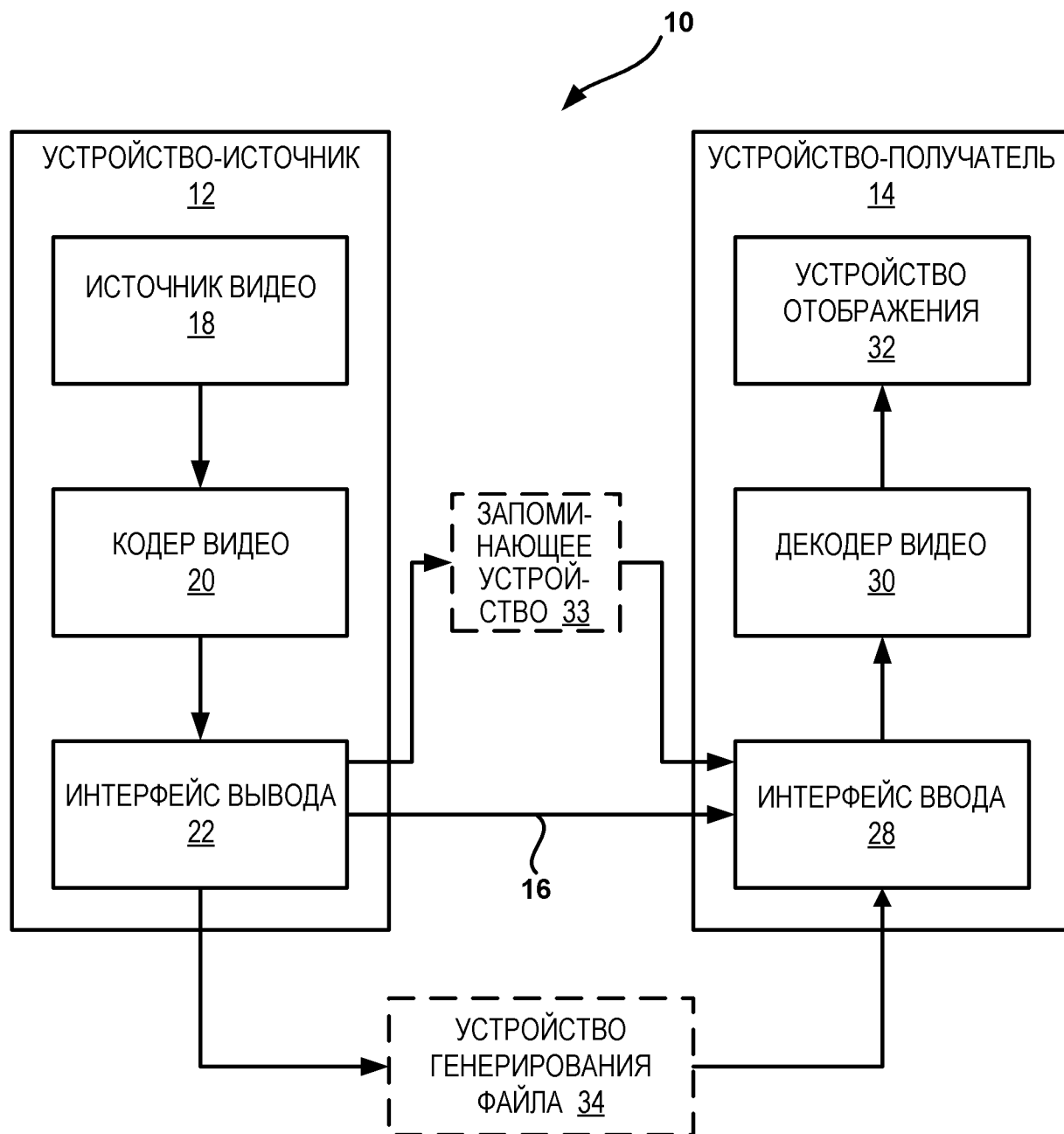
сохранять информацию формата представления для каждой рабочей точки многослойных видеоданных в контейнере oinf; и

генерировать файл видеоданных, отформатированный в соответствии с форматом файла.

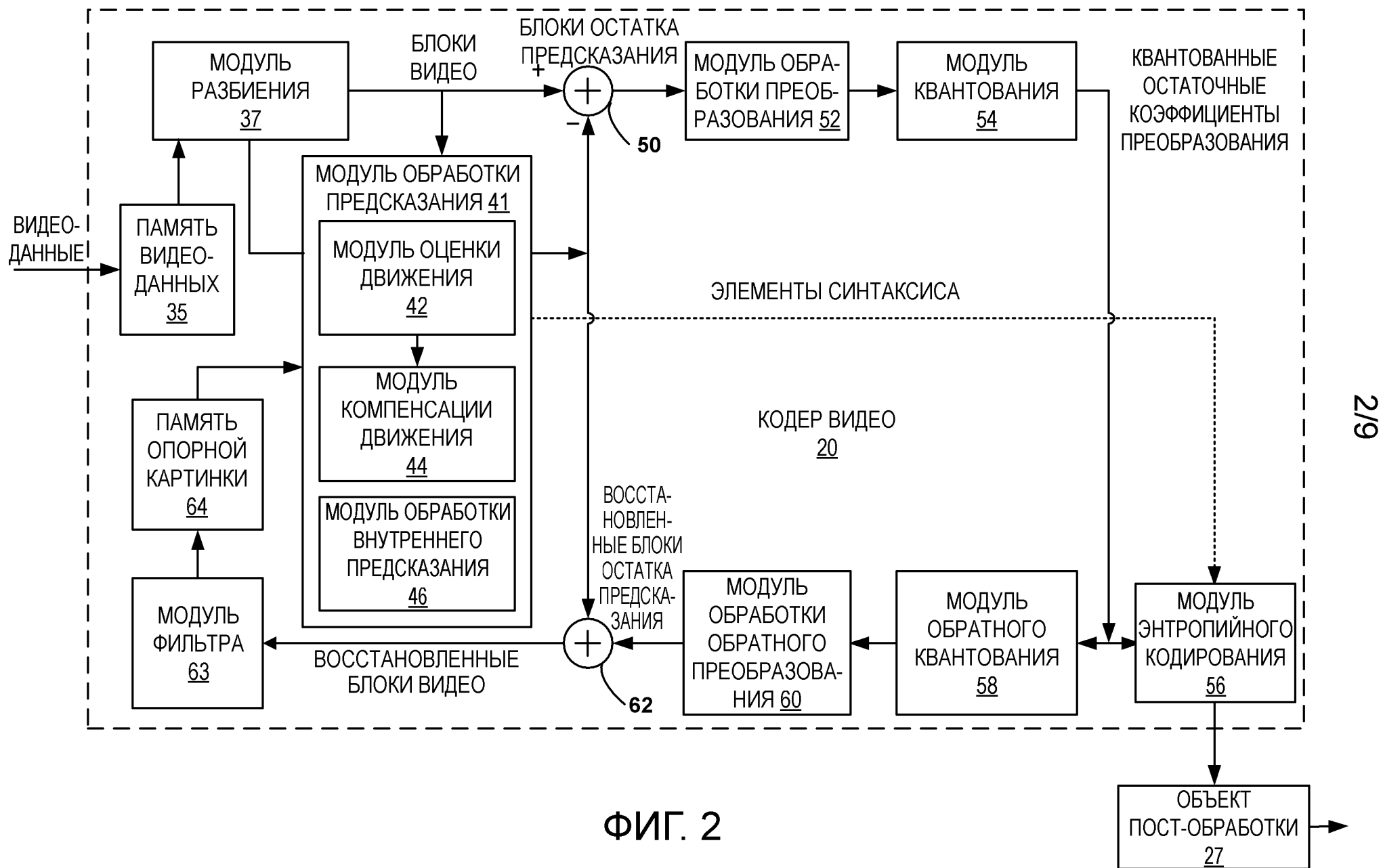
29. Машиночитаемый запоминающий носитель информации по п. 28, в котором контейнер oinf включается в контейнер информации мультимедиа.

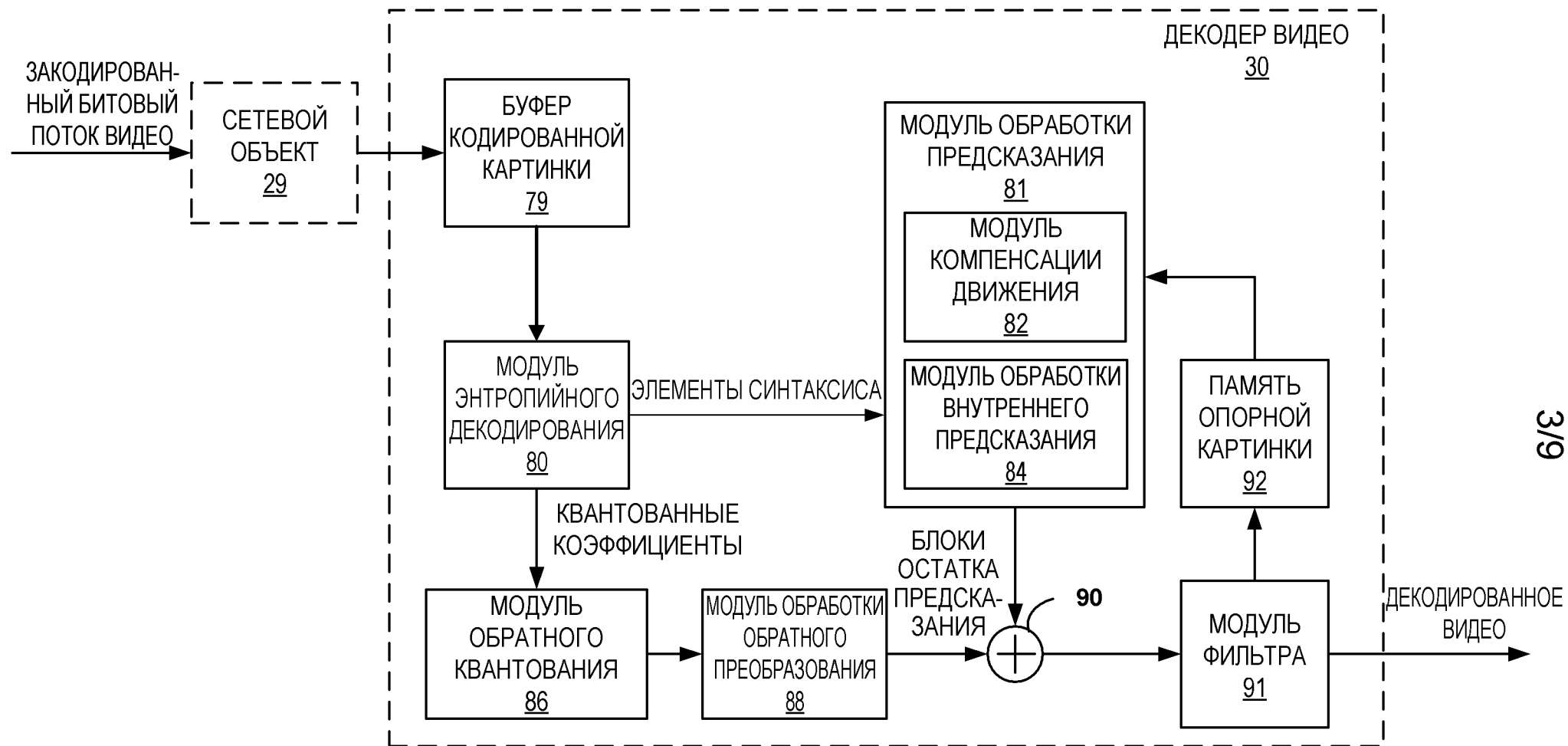
30. Машиночитаемый запоминающий носитель информации по п. 29, в котором контейнер oinf дополнительно включается в контейнер описания группы выборок, при этом контейнер описания группы выборок включается в контейнер таблицы выборок, и при этом контейнер таблицы выборок включается в контейнер информации мультимедиа.

По доверенности

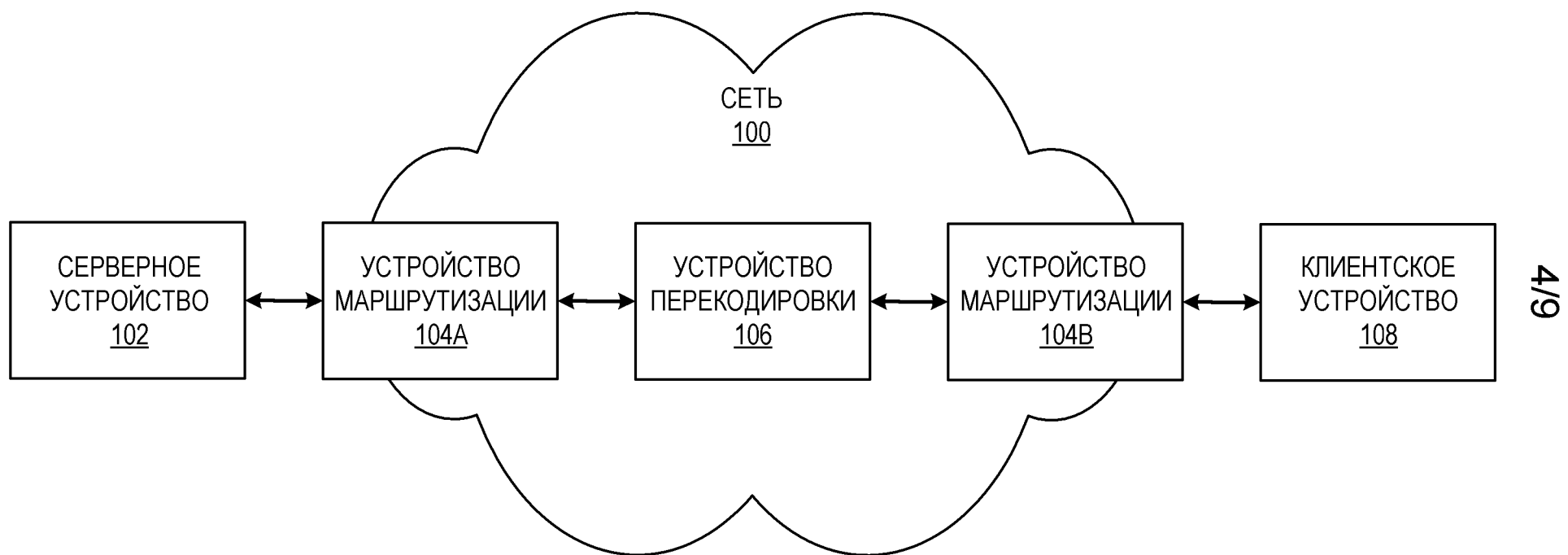


ФИГ. 1



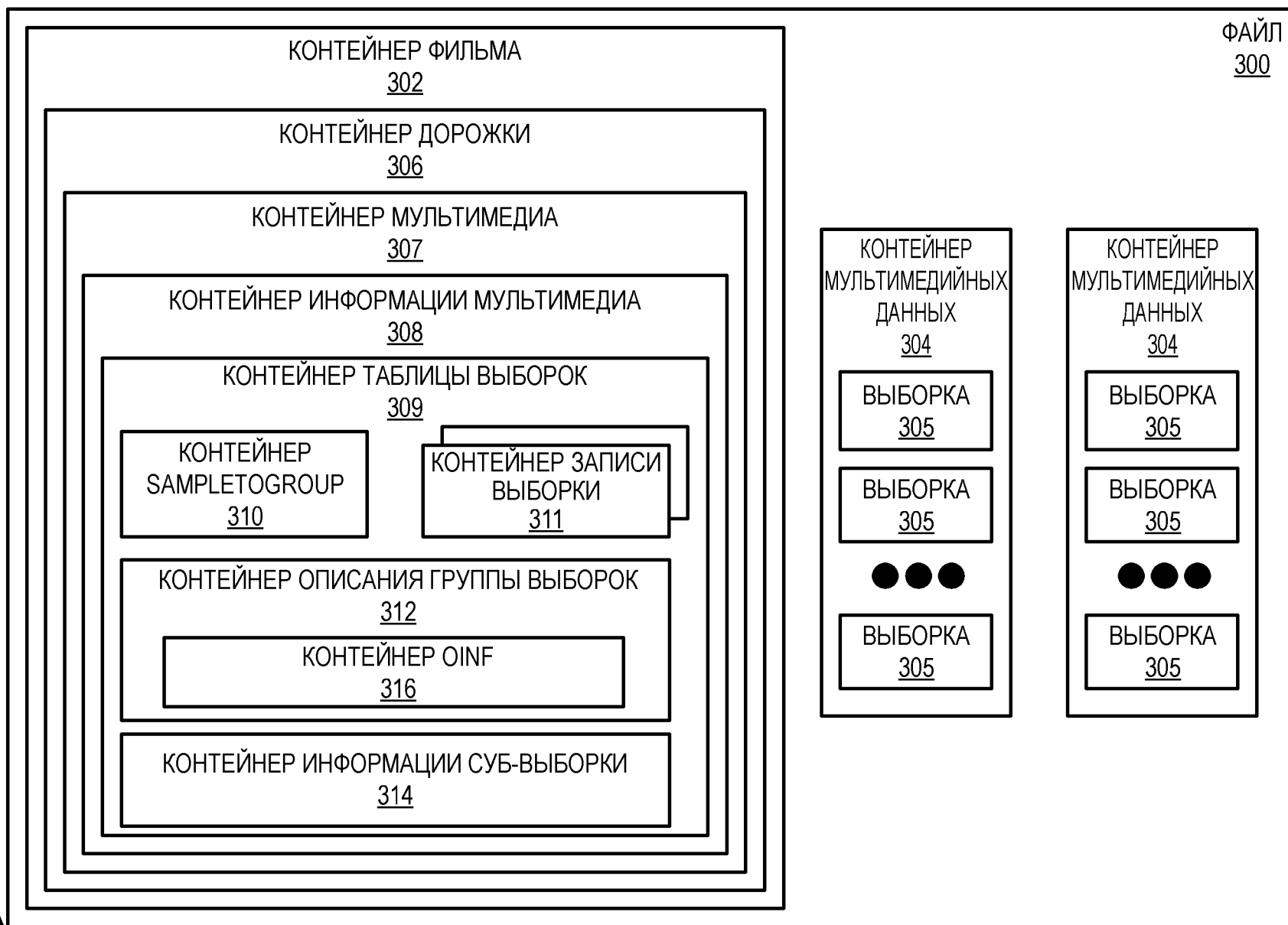


ФИГ. 3

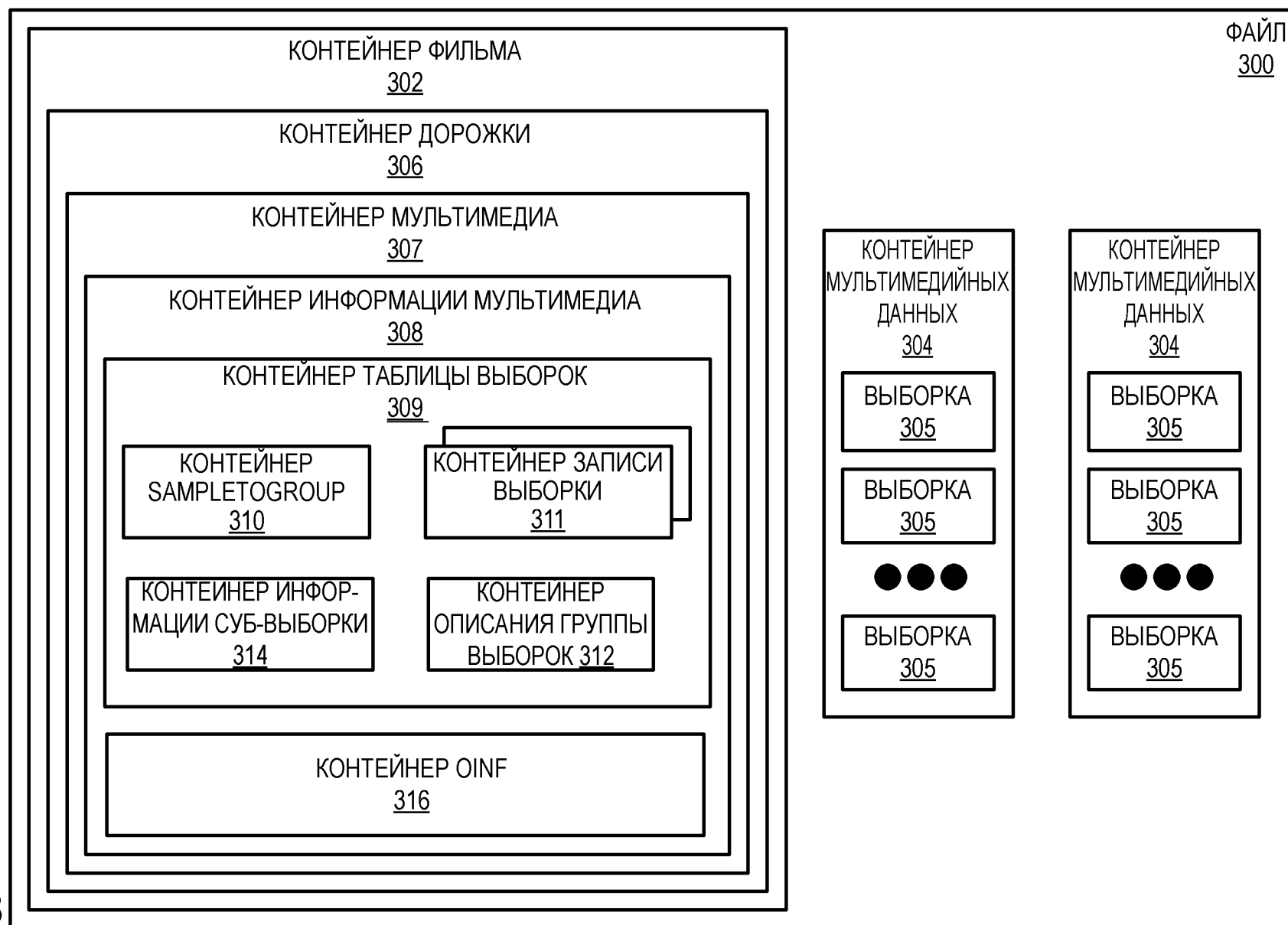


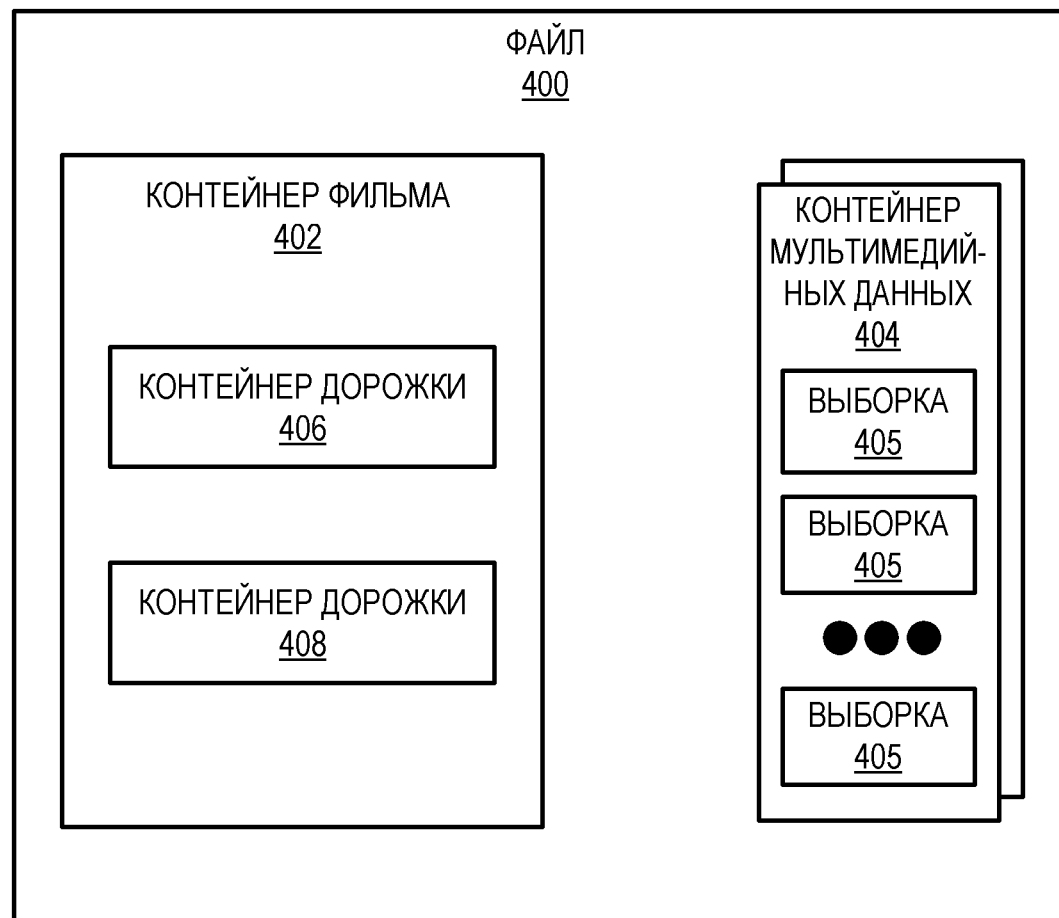
ФИГ. 4

ФИГ. 5А



ФИГ. 5В

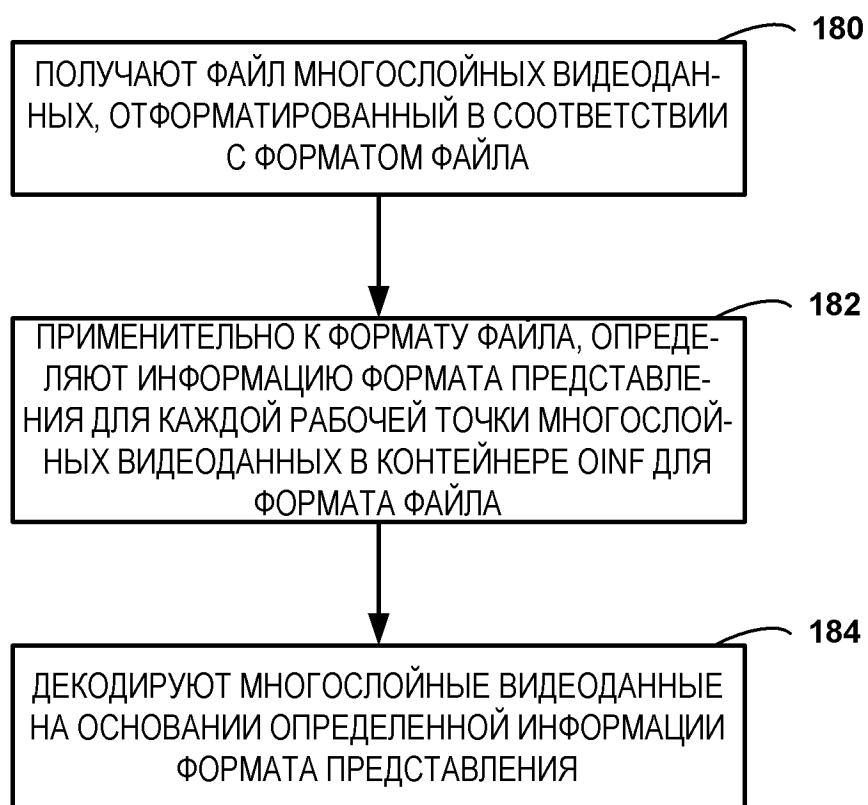




ФИГ. 6



ФИГ. 7



ФИГ. 8