

(19)



**Евразийское
патентное
ведомство**

(11) **045133**

(13) **B1**

(12) **ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ**

(45) Дата публикации и выдачи патента
2023.10.30

(21) Номер заявки
202191197

(22) Дата подачи заявки
2019.10.30

(51) Int. Cl. **H04N 19/30** (2014.01)
H04N 19/463 (2014.01)
H04N 19/70 (2014.01)

(54) **СПОСОБ, УСТРОЙСТВО И МАШИНОЧИТАЕМЫЙ НОСИТЕЛЬ, СОДЕРЖАЩИЙ КОМПЬЮТЕРНУЮ ПРОГРАММУ ДЛЯ ФОРМИРОВАНИЯ КОНФИГУРАЦИОННОГО СООБЩЕНИЯ ДЛЯ ОБЕСПЕЧЕНИЯ ВОЗМОЖНОСТИ ДЕКОДЕРУ ВОССТАНАВЛИВАТЬ ПЕРВОЕ ПРЕДСТАВЛЕНИЕ ИЗОБРАЖЕНИЯ С ПЕРВЫМ УРОВНЕМ КАЧЕСТВА С ПОМОЩЬЮ ДАННЫХ КОНФИГУРАЦИИ**

(31) **1817781.6**

(32) **2018.10.31**

(33) **GB**

(43) **2021.07.26**

(86) **PCT/GB2019/053067**

(87) **WO 2020/089615 2020.05.07**

(71)(73) Заявитель и патентовладелец:
**В-НОВА ИНТЕРНЕСНЛ ЛИМИТЕД
(GB)**

(72) Изобретатель:
Джонсон Роберт (GB)

(74) Представитель:
**Харин А.В., Стойко Г.В., Буре Н.Н.,
Алексеев В.В. (RU)**

(56) **US-A1-2015264404**

US-A1-2015304665

US-A1-2002141409

EP-A1-3131296

Anonymous: "Scalable HEVC encoding: How to setup cfg files for quality scalability? - Stack Overflow", 26 June 2017 (2017-06-26), XP055660374, Retrieved from the Internet: URL:<https://stackoverflow.com/questions/44203320/scalable-hevc-encoding-how-to-setup-cfg-files-for-quality-scalability>, [retrieved on 2020-01-21], the whole document

(57) Остаточные данные получают, исходя из первого и второго представлений изображения с первым уровнем качества. Второе представление основано на представлении изображения со вторым, более низким уровнем качества. Остаточные данные могут быть применены в декодере для восстановления первого представления с помощью второго представления. Конфигурационное сообщение (300), содержащее часть (310) заголовка и часть (320) полезных данных, формируется и подается на вывод для обработки декодером. Часть (320) полезных данных содержит полученные данные (350) конфигурации, относящиеся к обработке остаточных данных. Часть (310) заголовка содержит: (i) параметр (312) типа сообщения, задающий тип конфигурационного сообщения и указывающий заданный формат полезных данных части (320) полезных данных, форматный размер части (320) полезных данных, имеющей указанный формат полезных данных заранее заданным или который может быть определен из содержимого части полезных данных, и (ii) параметр (314) размера полезных данных, определяющий фактический размер части (320) полезных данных.

B1

045133

**045133
B1**

Область техники

Изобретение относится к способам, устройствам, компьютерным программам и машиночитаемому носителю. В частности, но не исключительно, настоящее изобретение относится к способам, устройствам, компьютерным программам и машиночитаемому носителю для применения при обработке конфигурационной информации, которая относится к остаточным данным, применимым для восстановления представления изображения с относительно высоким уровнем качества.

Уровень техники

Сжатие и восстановление сигналов является важным элементом во многих известных системах. Многие типы сигналов, например видео, могут быть сжаты и закодированы для передачи, например, по сети передачи данных. Когда такой сигнал декодируют, может быть желательным повысить уровень качества такого сигнала и/или восстановить, в максимально возможной степени, как можно больше информации, содержащейся в исходном сигнале. В некоторых известных системах применяют способы масштабируемого кодирования. Масштабируемое кодирование включает в себя кодирование сигнала вместе с информацией, позволяющей восстанавливать сигнал с одним или несколькими различными уровнями качества, например, в зависимости от возможностей декодера и доступной пропускной способности. Существует несколько ограничений, касающихся восстановления сигналов в системе масштабируемого кодирования. Одним из таких ограничений является объем информации, которая хранится, применяется и/или передается. Количество информации может меняться, например, в зависимости от желаемого уровня качества восстановленного сигнала, характера информации, которая применяется при восстановлении, и/или от того, как такая информация сконфигурирована. Еще одно ограничение - это способность декодера надежно обрабатывать полученную информацию. Одним из факторов, который может повлиять на надежность декодера, является его способность обрабатывать ошибки и/или обрабатывать принятую информацию, которая является непредвиденной, измененной и/или нераспознанной декодером.

Сущность изобретения

Различные аспекты настоящего изобретения изложены в прилагаемой формуле изобретения.

Дополнительные признаки и преимущества станут очевидными из следующего описания предпочтительных вариантов осуществления, приведенного только в качестве примера, со ссылкой на прилагаемые графические материалы.

Краткое описание графических материалов

На фиг. 1 показана блок-схема примера системы обработки сигналов в соответствии с вариантом осуществления настоящего изобретения;

На фиг. 2А и 2В показана блок-схема другого примера системы обработки сигналов в соответствии с вариантом осуществления настоящего изобретения;

На фиг. 3 показана схема примера конфигурационного сообщения в соответствии с вариантом осуществления настоящего изобретения;

На фиг. 4 показана последовательность операций, представляющая пример способа согласно варианту осуществления настоящего изобретения;

На фиг. 5 показана схема еще одного примера конфигурационного сообщения в соответствии с вариантом осуществления настоящего изобретения;

На фиг. 6 показана последовательность операций, представляющая другой пример способа в соответствии с вариантом осуществления настоящего изобретения; и

На фиг. 7 показана общая блок-схема примера устройства в соответствии с вариантом осуществления настоящего изобретения.

Подробное описание

На фиг. 1 показан пример системы 100 обработки сигналов. Систему 100 обработки сигналов применяют для обработки сигналов. Примеры типов сигналов включают в себя, помимо прочего, видеосигналы, сигналы изображения, аудиосигналы, пространственные сигналы, такие как те, которые применяют в медицинских, научных или голографических изображениях, или другие многомерные сигналы.

Система 100 обработки сигналов включает в себя первое устройство 102 и второе устройство 104. Первое устройство 102 и второе устройство 104 могут иметь взаимосвязь клиент-сервер, при этом первое устройство 102 выполняет функции серверного устройства, а второе устройство 104 выполняет функции клиентского устройства. Система 100 обработки сигналов может включать в себя по меньшей мере одно дополнительное устройство (не показано). Первое устройство 102 и/или второе устройство 104 могут содержать один или более компонентов. Один или более компонентов могут быть реализованы аппаратно и/или программно. В системе 100 обработки сигналов один или более компонентов могут быть расположены совместно или могут быть расположены удаленно друг от друга. Примеры типов устройств включают в себя, помимо прочего, компьютеризированные устройства, карманные или переносные компьютеры, планшеты, мобильные устройства, игровые консоли, интеллектуальные телевизоры, телевизионные приставки, гарнитуры дополненной и/или виртуальной реальности и т. д.

Первое устройство 102 соединено с возможностью связи со вторым устройством 104 посредством сети 106 передачи данных. Примеры сети 106 передачи данных включают в себя, помимо прочего, Ин-

тернет, локальную компьютерную сеть (LAN - англ.: Local Area Network) и глобальную компьютерную сеть (WAN - англ.: Wide Area Network). Первое и/или второе устройство 102, 104 может иметь проводное и/или беспроводное соединение с сетью 106 передачи данных.

Первое устройство 102 содержит кодер 108. Кодер 108 выполнен с возможностью кодирования данных, содержащихся в сигнале, которые далее в настоящем документе называют "данные сигнала". Например, если такой сигнал является видеосигналом, кодер 108 выполнен с возможностью кодирования видеоданных. Видеоданные содержат последовательность из множества изображений или кадров. В дополнение к кодированию данных сигнала кодер 108 может выполнять одну или более дополнительных функций. Кодер 108 может быть реализован различными способами. Например, кодер 108 может быть реализован аппаратно и/или программно.

Хотя в этом примере первое устройство 102 содержит кодер 108, в других примерах первое устройство 102 отделено от кодера 108. В таких примерах первое устройство 102 соединено с возможностью связи с кодирующим устройством 108. Первое устройство 102 может быть реализовано как одна или более программных функций и/или аппаратных модулей.

Второе устройство 104 содержит декодер 110. Декодер 110 выполнен с возможностью декодирования данных сигнала. В дополнение к декодированию данных сигнала декодер 110 может выполнять одну или более дополнительных функций. Декодер 110 может быть реализован различными способами. Например, декодер 110 может быть реализован аппаратно и/или программно.

Хотя в этом примере второе устройство 104 содержит декодер 110, в других примерах второе устройство 104 отделено от декодера 110. В таких примерах второе устройство 104 соединено с возможностью связи с декодером 110. Второе устройство 104 может быть реализовано как одна или более программных функций и/или аппаратных модулей.

Кодер 108 кодирует данные сигнала и передает закодированные данные сигнала в декодер 110 через сеть 106 передачи данных. Декодер 110 декодирует принятые закодированные данные сигнала и формирует декодированные данные сигнала. Декодер 110 может выводить декодированные данные сигнала или данные, полученные с помощью декодированных данных сигнала. Например, декодер 110 может выводить такие данные для отображения на одном или более устройств отображения, связанных со вторым устройством 104.

В некоторых описанных здесь примерах кодер 108 передает в декодер 110 представление сигнала с заданным уровнем качества и информацию, которую декодер 110 может использовать для восстановления представления сигнала с одним или более повышенных уровней качества. Такая информация может называться "данными восстановления". В некоторых примерах "восстановление" представления подразумевает получение представления, которое не является точной копией исходного представления. Степень, в которой такое представление совпадает с исходным, может зависеть от различных факторов, включая, помимо прочего, уровни квантования. Представление сигнала с заданным уровнем качества можно рассматривать как отображение, версию или изображение данных, содержащихся в сигнале, с заданным уровнем качества. В некоторых примерах данные восстановления включены в данные сигнала, которые закодированы с помощью кодера 108 и переданы в декодер 110. Например, данные восстановления могут иметь форму метаданных. В некоторых примерах данные восстановления кодируют и передают отдельно от данных сигнала.

Информация, которую декодер 110 применяет для восстановления представления сигнала на одном или более повышенных уровнях качества, может содержать остаточные данные, как более подробно описано ниже. Остаточные данные являются примером данных восстановления. Информация, которую декодер 110 применяет для восстановления представления сигнала на одном или более повышенных уровнях качества, также может содержать данные конфигурации, относящиеся к обработке остаточных данных. Данные конфигурации могут указывать, как эти остаточные данные были обработаны кодирующим устройством 108 и/или как остаточные данные должны быть обработаны декодером 110. Данные конфигурации могут быть переданы в декодер 110, например, в форме метаданных.

На фиг. 2А и 2В схематически показан пример системы 200 обработки сигналов. Система 200 обработки сигналов включает в себя первое устройство 202, содержащее кодирующее устройство, и второе устройство 204, содержащее декодер. В каждом таком первом устройстве 202 и втором устройстве 204 элементы показаны на двух логических уровнях. Два уровня разделены пунктирной линией. Элементы первого, самого высокого уровня, связаны с данными относительно высокого уровня качества. Элементы второго, самого низкого уровня, связаны с данными относительно низкого уровня качества. Относительно высокий и относительно низкий уровни качества относятся к многоуровневой иерархии, имеющей несколько уровней качества. В некоторых примерах многоуровневая иерархия содержит более двух уровней качества. В каждом из примеров первое устройство 202 и второе устройство 204 могут включать в себя более двух разных уровней. Могут существовать один или более других уровней, выше и/или ниже тех, которые изображены на фиг. 2А и 2В.

Как показано на фиг. 2А, первое устройство 202 получает первое представление изображения с относительно высоким уровнем 206 качества. Представление данного изображения - это представление данных, содержащихся в изображении. Данное изображение может представлять собой заданный кадр

видео. Первое представление изображения с относительно высоким уровнем 206 качества далее будет называться "входными данными", поскольку, в данном примере, они являются данными, предоставленными в качестве входных данных кодера в первом устройстве 202. Первое устройство 202 может принимать входные данные 206. Например, первое устройство 202 может принимать входные данные 206 от по меньшей мере одного другого устройства. Первое устройство 202 может быть выполнено с возможностью приема последовательных фрагментов входных данных 206, например, последовательных кадров видео, и выполнения описанных здесь операций для каждого последующего кадра. Например, видео может содержать кадры F_1, F_2, \dots, F_T , а первое устройство 202 может обрабатывать каждый из них по очереди.

Первое устройство 202 получает данные 212, основанные на входных данных 206. В этом примере данные 212, основанные на входных данных 206, являются представлением 212 изображения с относительно низким уровнем качества. В этом примере данные 212 получают путем выполнения над входными данными 206 операции понижающей дискретизации, и поэтому далее в настоящем документе они называются "субдискретизированными данными". В других примерах данные 212 получают путем выполнения над входными данными 206 операции, отличной от операции понижающей дискретизации.

В этом примере субдискретизированные данные 212 обрабатывают для формирования обработанных данных 213 с относительно низким уровнем качества. В других примерах субдискретизированные данные 212 не обрабатывают с относительно низким уровнем качества. Таким образом, первое устройство 202 может формировать данные с относительно низким уровнем качества, при этом такие данные с относительно низким уровнем качества содержат субдискретизированные данные 212 или обработанные данные 213.

В некоторых примерах формирование обработанных данных 213 предусматривает кодирование субдискретизированных данных 212. Кодирование субдискретизированных данных 212 дает закодированное изображение с относительно низким уровнем качества. Первое устройство 202 может выводить закодированное изображение, например, для передачи на второе устройство 204. Ряд закодированных изображений, например, формирующий закодированное видео, который выводится для передачи на второе устройство 204, может называться "базовым" потоком. Вместо того, чтобы создаваться в первом устройстве 202, закодированное изображение может создаваться устройством кодирования, которое является отдельным от первого устройства 202. Закодированное изображение может быть частью видео, закодированного в формате H.264. Формирование обработанных данных 213 может, например, содержать создание последовательных кадров видео в виде выходных данных отдельного устройства кодирования, такого как видеокодер H.264. Промежуточный набор данных для формирования обработанных данных 213 может содержать выходные данные такого кодирующего устройства, в отличие от любых промежуточных данных, формируемых отдельным кодирующим устройством.

Формирование обработанных данных 213 с относительно низким уровнем качества может дополнительно предусматривать декодирование закодированного изображения с относительно низким уровнем качества. Операция декодирования может быть выполнена для эмуляции операции декодирования на втором устройстве 204, как станет очевидно далее. Декодирование закодированного изображения дает декодированное изображение с относительно низким уровнем качества. В некоторых примерах первое устройство 202 декодирует закодированное изображение с относительно низким уровнем качества для создания декодированного изображения с относительно низким уровнем качества. В других примерах первое устройство 202 принимает декодированное изображение с относительно низким уровнем качества, например, от кодирующего и/или декодирующего устройства, которое является отдельным от первого устройства 202. Закодированное изображение может быть декодировано с помощью декодера H.264. Декодирование с помощью отдельного декодирующего устройства может содержать входящее закодированное видео, такое как поток закодированных данных, выполненное с возможностью передачи на удаленное декодирующее устройство, в отдельный декодер с неизвестными свойствами, реализованный совместно с первым устройством 202 для формирования последовательных декодированных кадров видео. Таким образом, обработанные данные 213 могут содержать кадр видеоданных, который формируется посредством сложного нелинейного процесса кодирования и декодирования, при этом такой процесс кодирования и декодирования может предусматривать моделирование пространственно-временных корреляций в соответствии с конкретным стандартом кодирования, таким как H.264. Однако, поскольку выходной сигнал любого кодирующего устройства подают в соответствующий декодер, эта сложность эффективно скрыта от первого устройства 202.

В примере формирование обработанных данных 213 с относительно низким уровнем качества дополнительно предусматривает получение данных коррекции, основанных на сравнении между субдискретизированными данными 212 и декодированным изображением, полученным первым устройством 202, например, основанных на разнице между субдискретизированными данными 212 и декодированным изображением. Данные коррекции могут применять для коррекции ошибок, внесенных при кодировании и декодировании субдискретизированных данных 212. В некоторых примерах первое устройство 202 подает на выход данные коррекции, например, для передачи во второе устройство 204, а также закодированный сигнал. Это позволяет получателю исправить ошибки, внесенные при кодировании и декодиро-

вании субдискретизированных данных 212. Эти данные коррекции также могут быть названы потоком "первого улучшения". Поскольку данные коррекции могут быть основаны на разнице между субдискретизированными данными 212 и декодированным изображением, их можно рассматривать как форму остаточных данных (например, которые отличаются от другого набора остаточных данных, описанного далее).

В некоторых примерах формирование обработанных данных 213 с относительно низким уровнем качества дополнительно предусматривает коррекцию декодированного изображения с помощью данных коррекции. Например, данные коррекции как выходные данные для передачи могут быть помещены в форму, подходящую для объединения с декодированным изображением, а затем добавлены к декодированному изображению. Это может быть выполнено покадрово. В других примерах, вместо того, чтобы исправлять декодированное изображение с помощью данных коррекции, первое устройство 202 применяет субдискретизированные данные 212. Например, в определенных случаях могут применяться только кодированные, а затем декодированные данные, а в других случаях кодирование и декодирование могут быть заменены другой обработкой.

В некоторых примерах формирование обработанных данных 213 предусматривает выполнение одной или нескольких операций, отличных от действий по кодированию, декодированию, получению и коррекции, описанных выше.

Первое устройство 202 получает данные 214, основанные на данных с относительно низким уровнем качества. Как указано выше, такие данные с относительно низким уровнем качества могут содержать обработанные данные 213 или субдискретизированные данные 212, при этом такие субдискретизированные данные 212 не обрабатывают на более низком уровне. Как описано выше, в определенных случаях обработанные данные 213 могут содержать поток восстановленного видео (например, от операции кодирования-декодирования), который скорректирован с помощью данных коррекции. В примере на фиг. 2А и 2В данные 214 представляют собой второе представление изображения с относительно высоким уровнем качества, причем первое представление изображения с относительно высоким уровнем качества является входными данными 206. Второе представление с относительно высоким уровнем качества можно рассматривать как предварительное или прогнозируемое представление изображения с относительно высоким уровнем качества. В этом примере первое устройство 202 получает данные 214 путем выполнения операции повышающей дискретизации данных с относительно низким уровнем качества. Данные 214 в дальнейшем будут называться "передискретизированными данными". Однако в других примерах одна или более других операций могут применяться для выработки данных 214, например, когда данные 212 не получают с помощью понижающей дискретизации входных данных 206. Следует отметить, что ссылки на относительно высокие и относительно низкие уровни качества могут соответствовать ссылкам на первый и второй уровень качества, причем второй уровень качества выше первого уровня качества. Как описано в настоящем документе, в некоторых случаях уровни качества могут соответствовать разным пространственным разрешениям.

Входные данные 206 и передискретизированные данные 214 применяют для получения остаточных данных 216. Остаточные данные 216 связаны с изображением. Остаточные данные 216 могут существовать в форме набора остаточных элементов. Остаточный элемент в наборе остаточных элементов 216 может быть связан с соответствующим элементом изображения во входных данных 206. Примером элемента изображения является пиксель.

В этом примере данный остаточный элемент получен путем вычитания значения элемента изображения в передискретизированных данных 214 из значения соответствующего элемента изображения во входных данных 206. По существу, для восстановления входных данных 206 применимы остаточные данные 216 в комбинации с передискретизированными данными 214. Остаточные данные 216 также могут называться "данными восстановления" или "данными улучшения". В одном случае остаточные данные 216 могут образовывать часть потока "второго улучшения".

Первое устройство 202 получает данные конфигурации, относящиеся к обработке остаточных данных 216. Данные конфигурации указывают, как остаточные данные 216 были обработаны и/или сформированы первым устройством 202, и/или как остаточные данные 216 должны быть обработаны вторым устройством 204. Данные конфигурации могут содержать набор параметров конфигурации. Данные конфигурации могут применяться для управления тем, как второе устройство 204 обрабатывает данные и/или восстанавливает входные данные 206 с помощью остаточных данных 216. Данные конфигурации могут относиться к одной или большему количеству характеристик остаточных данных 216. Отличные данные конфигурации могут приводить к различной обработке, выполняемой над остаточными данными 216 и/или с их использованием. Таким образом, данные конфигурации применимы для восстановления входных данных 206 с помощью остаточных данных 216.

В этом примере первое устройство 202 передает на второе устройство 204 данные, основанные на субдискретизированных данных 212, данные, основанные на остаточных данных 216 и данных конфигурации, давая возможность второму устройству 204 восстановить входные данные 206.

Обращаясь теперь к фиг. 2В, второе устройство 204 принимает данные 220, основанные (например, выработанные) на субдискретизированных данных 212. Второе устройство 204 также принимает данные,

основанные на остаточных данных 216. Например, второе устройство 204 может принимать "базовый" поток (данные 220), "поток первого улучшения" (любые данные коррекции) и "поток второго улучшения" (остаточные данные 216). Второе устройство 204 также принимает данные конфигурации, относящиеся к обработке остаточных данных 216. Данные 220, основанные на субдискретизированных данных 212, могут быть собственно субдискретизированными данными 212, обработанными данными 213 или данными, выработанными из таких субдискретизированных данных 212 или обработанных данных 213. Данные, основанные на остаточных данных 216, могут быть самими остаточными данными 216 или данными, выработанными из таких остаточных данных 216.

В некоторых примерах принятые данные 220 содержат обработанные данные 213, которые могут содержать закодированное изображение с относительно низким уровнем качества и/или данные коррекции. В некоторых примерах, например, когда первое устройство 202 обработало субдискретизированные данные 212 для формирования обработанных данных 213, второе устройство 204 обрабатывает принятые данные 220 для формирования обработанных данных 222. Такая обработка вторым устройством 204 может содержать декодирование закодированного изображения (например, которое формирует часть "базового" потока кодированного видео) для создания декодированного изображения с относительно низким уровнем качества. В некоторых примерах обработка вторым устройством 204 содержит коррекцию декодированного изображения с помощью полученных данных коррекции. Следовательно, обработанные данные 222 могут содержать кадр скорректированных данных с первым или относительно низким уровнем качества. В некоторых примерах закодированное изображение с относительно низким уровнем качества декодируется декодирующим устройством, которое отделено от второго устройства 204. Закодированное изображение с относительно низким уровнем качества может быть декодировано с помощью декодера H.264.

В других примерах принятые данные 220 содержат субдискретизированные данные 212 и не содержат обработанные данные 213. В некоторых таких примерах второе устройство 204 не обрабатывает принятые данные 220 для формирования обработанных данных 222.

Для получения передискретизированных данных 214 во втором устройстве 204 применяются данные с относительно низким уровнем качества. Как указано выше, данные с относительно низким уровнем качества могут содержать обработанные данные 222 или принятые данные 220, причем второе устройство 204 не обрабатывает принятые данные 220 с относительно низким уровнем качества. Передискретизированные данные 214 представляют собой предварительное представление изображения с относительно высоким уровнем качества. Передискретизированные данные 214 могут быть выработаны путем выполнения операции повышающей дискретизации над данными с относительно низким уровнем качества.

Второе устройство 204 получает остаточные данные 216. Для восстановления входных данных 206 применимы остаточные данные 216 с передискретизированными данными 214. Остаточные данные 216 являются признаком сравнения между входными данными 206 и передискретизированными данными 214.

Второе устройство 204 также получает данные конфигурации, относящиеся к обработке остаточных данных 216. Данные конфигурации могут использоваться вторым устройством 204 для восстановления входных данных 206. Например, данные конфигурации могут указывать на характеристику или свойство, относящееся к остаточным данным 216, которые влияют на то, как остаточные данные 216 должны применяться и/или обрабатываться, или должны ли вообще применяться остаточные данные 216. В некоторых примерах данные конфигурации содержат остаточные данные 216.

Имеются несколько соображений, касающихся применения таких данных конфигурации. Одним из таких соображений является объем информации, которая формируется, хранится, передается и/или обрабатывается. Чем больше информации применяется, тем большее количество ресурсов может быть предусмотрено для обработки такой информации. Примеры таких ресурсов включают в себя ресурсы для передачи, ресурсы для хранения и ресурсы для обработки. По сравнению с некоторыми известными технологиями, описанные здесь примеры позволяют использовать относительно небольшой объем информации. Это может уменьшить объем данных, передаваемых через сеть 106 передачи данных. Экономия может быть особенно значимой, когда данные относятся к данным видео высокого качества, где объем информации, передаваемой в известных системах, может быть особенно большим.

Еще одним соображением, касающимся использования таких данных конфигурации, является возможность адаптации к декодирующим устройствам различных типов. Например, данное кодирующее устройство может передавать информацию множеству декодеров, каждый из которых имеет отличающиеся характеристики, такие как возможности по обработке и/или хранению. Желательно, чтобы применение данных конфигурации при восстановлении изображения не зависело от типа и/или возможностей декодирующего устройства, которое применяется для выполнения восстановления. По сравнению с некоторыми известными технологиями, примеры, описанные в настоящем документе, позволяют различным типам декодеров, имеющих отличающиеся рабочие характеристики, надежно обрабатывать данные конфигурации.

Дополнительным соображением, касающимся использования таких данных конфигурации, является

объем и/или сложность обработки, предусмотренной в декодере для обработки данных конфигурации. По сравнению с некоторыми известными технологиями, примеры, описанные в настоящем документе, уменьшают объем обработки, выполняемой декодером для получения и/или обработки данных конфигурации.

Также следует учитывать и обработку ошибок. Например, способность декодера обрабатывать потенциальные ошибки при обработке данных конфигурации может являться фактором надежности декодера. По сравнению с некоторыми известными технологиями, описанные в настоящем документе примеры позволяют декодеру обрабатывать потенциальные ошибки, а также получать непредусмотренные или нераспознанные данные. Это может эффективно повысить надежность декодера.

На фиг. 3 схематически показан пример конфигурационного сообщения 300. Устройство, например первое устройство 102, содержащее кодер 108, формирует конфигурационное сообщение 300 и выводит такое конфигурационное сообщение 300 для обработки дополнительным устройством, например вторым устройством 104, которое содержит декодер 110.

Конфигурационное сообщение 300 содержит часть 310 заголовка и часть 320 полезных данных. Часть 320 полезных данных содержит данные 350 конфигурации. Данные 350 конфигурации относятся к обработке остаточных данных, например, остаточных данных 216, как описано выше. В некоторых примерах длина конфигурационного сообщения 300 представляет собой целое число байтов.

Часть 310 заголовка содержит параметр 312 типа сообщения. Параметр 312 типа сообщения задает тип конфигурационного сообщения 300. Тип сообщения, заданный параметром 312 типа сообщения, может указывать, как должно быть обработано конфигурационное сообщение 300. Примером типа сообщения является конфигурационное сообщение набора изображений. Конфигурационное сообщение набора изображений может соответствовать множеству изображений в наборе изображений. Другим примером типа сообщения является конфигурационное сообщение изображения. Конфигурационное сообщение изображения может соответствовать заданному изображению только в наборе изображений. Другим примером типа сообщения является сообщение с закодированными данными. Сообщение с закодированными данными может быть выполнено с возможностью содержания закодированных остаточных данных для данного изображения. Еще одним примером типа сообщения является сообщение метаданных кодирующего устройства. Сообщение метаданных кодирующего устройства может содержать метаданные, относящиеся к одной или большему количеству характеристик кодирующего устройства. Еще одним примером типа сообщения является сообщение управления декодером. Сообщение управления декодером может содержать команды для управления декодером.

Параметр 312 типа сообщения указывает на то, что часть 320 полезных данных имеет заданный формат полезных данных. Формат полезных данных части 320 полезных данных может быть определен на основании сообщения конфигурационного сообщения 300. Формат полезных данных может определять набор параметров конфигурации данных 350 конфигурации, которые содержатся в части 320 полезных данных. В некоторых примерах формат полезных данных части 320 полезных данных определяет конкретную логику обработки, которая задает, как следует обрабатывать содержимое части 320 полезных данных. Часть 320 полезных данных имеет форматный размер. Форматный размер части 320 полезных данных соответствует формату полезных данных части 320 полезных данных. Форматный размер части полезных данных, имеющей заданный формат полезных данных, определен заранее или может быть определен из содержимого части полезных данных. Например, форматный размер может быть определен, исходя из объединенного размера отдельных параметров конфигурации, которые определены форматом полезных данных. Форматный размер может рассматриваться как возможный размер или предполагаемый размер части 320 полезных данных.

Часть 310 заголовка содержит параметр 314 размера полезных данных. Параметр 314 размера полезных данных задает фактический размер части 320 полезных данных. Фактический размер части 320 полезных данных может быть выражен целым числом байтов.

В некоторых примерах значение параметра 314 размера полезных данных хранится в элементе фиксированной длины. Элемент фиксированной длины, хранящий значение параметра 314 размера полезных данных, может содержать заранее определенное количество битов в заданном байте части 310 заголовка. Например, значение параметра 314 размера полезных данных может быть сохранено в элементе фиксированной длины, который содержит три бита в однобайтовой части заголовка.

В некоторых примерах параметр 314 размера полезных данных содержит ссылку на таблицу соответствия размера полезных данных, в которой хранятся возможные значения фактического размера части 320 полезных данных. Например, параметр 314 размера полезных данных может содержать значение, применимое для получения фактического размера части 320 полезных данных из таблицы соответствия размера полезных данных.

В некоторых примерах значение параметра 314 размера полезных данных хранится в элементе переменной длины. Это может иметь место, например, когда фактический размер части 310 полезных данных не является одним из значений, указанных в таблице соответствия размера полезных данных. Фактический размер части 310 полезных данных, определенный параметром 314 размера полезных данных, может иметь переменную длину в битах. Элемент переменной длины содержит целое число байтов. В

некоторых примерах по меньшей мере один байт элемента переменной длины имеет один или более заранее определенных битов, выполненных с возможностью указания того, содержит ли элемент переменной длины один или более дополнительных байтов для такого по меньшей мере одного заданного байта. Например, заданный байт элемента переменной длины может содержать семь битов, которые выполнены с возможностью хранения параметра 314 размера полезных данных, и один бит, который применяется, чтобы указывать, имеется ли дополнительный байт после данного байта в элементе переменной длины или нет. По существу, параметр 314 размера полезных данных может быть представлен с помощью элемента переменной длины независимо от длины в битах параметра 314 размера полезных данных. Элемент переменной длины содержит минимальное количество байтов, которые применимы для хранения значения параметра 314 размера полезных данных.

В некоторых примерах данные хранятся в элементе фиксированной длины, который указывает, что значение параметра 314 размера полезных данных хранится в элементе переменной длины. Например, если определено, что значение параметра 314 размера полезных данных является одним из заранее определенного набора значений, такое значение может быть сохранено в элементе фиксированной длины и/или элемент фиксированной длины может включать в себя ссылку на таблицу соответствия размера полезных данных. С другой стороны, если определено, что значение параметра 314 размера полезных данных не является одним значением из заранее определенного набора значений, элемент переменной длины может применяться для хранения значения, а элемент фиксированной длины может быть выполнен с возможностью указания, например, с помощью флага, содержащегося в элементе фиксированной длины, что для хранения значения применяется элемент переменной длины.

В этом примере часть 310 заголовка содержит только параметр 312 типа сообщения и параметр 314 размера полезных данных. В других примерах часть 310 заголовка содержит один или более дополнительных параметров. В некоторых примерах длина части 310 заголовка составляет один байт. Например, параметр 312 типа сообщения может быть представлен пятью битами, а параметр 314 размера полезных данных может быть представлен оставшимися тремя битами однобайтовой части 310 заголовка. В других примерах, например, где для представления фактического размера части 320 полезных данных применяется элемент переменной длины, часть 310 заголовка содержит более одного байта.

Фактический размер части 320 полезных данных может быть таким же или отличаться от форматного размера части 320 полезных данных. Независимо от того, являются ли фактический размер и форматный размер одинаковыми, параметр 314 размера полезных данных, задающий фактический размер, включен в часть 310 заголовка конфигурационного сообщения 300.

Включение параметра 314 размера полезных данных, который задает фактический размер части 320 полезных данных, когда возможный размер части 320 полезных данных, а именно форматный размер, может быть логически выведен из параметра 312 типа сообщения, может показаться неэффективным и/или дублирующим информацию, особенно если форматный размер и фактический размер совпадают. Включение параметра 314 размера полезных данных в конфигурационное сообщение 300 предусматривает применение большего количества данных для конфигурационного сообщения 300 по сравнению со случаем, в котором такой параметр не является включенным. Однако включение параметра 314 размера полезных данных в конфигурационное сообщение 300 способствует более эффективной и надежной обработке декодером 110 и/или улучшенной обработке ошибок по сравнению со случаем, в котором такой параметр не применяется, что будет более подробно описано ниже.

На фиг. 4 показан пример способа 400 обработки данных конфигурации. Способ 400 может быть осуществлен устройством, содержащим декодер, таким как второе устройство 104, описанное выше.

В некоторых примерах приняты остаточные данные. В этом примере принятые остаточные данные представляют собой остаточные данные 216, описанные выше. Остаточные данные 216 могут быть приняты от кодирующего устройства, такого как кодер 108. Остаточные данные 216 являются применимыми для восстановления первого представления 206 изображения с относительно высоким уровнем качества в многоуровневой иерархии с помощью второго представления 214 изображения с относительно высоким уровнем качества. Второе представление 214 основано на представлении 222 изображения с относительно низким уровнем качества в многоуровневой иерархии. В некоторых примерах данные, применимые для получения представления 222 с относительно низким уровнем качества, принимают от кодера 108.

В позиции 410 принимается конфигурационное сообщение. В этом примере принятое конфигурационное сообщение представляет собой конфигурационное сообщение 300, описанное выше. Конфигурационное сообщение 300 может быть принято от кодера 108.

В позиции 420 принятое конфигурационное сообщение 300 обрабатывают с помощью параметра 312 типа сообщения и параметра 314 размера полезных данных. Конфигурационное сообщение 300 обрабатывают для получения данных 350 конфигурации, хранящихся в части 320 полезных данных конфигурационного сообщения 300.

В некоторых примерах обработка конфигурационного сообщения 300 содержит синтаксический анализ части 310 заголовка конфигурационного сообщения 300 для получения параметра 312 типа сообщения и параметра 314 размера полезных данных. Исходя из параметра 314 размера полезных данных определяют фактический размер части 320 полезных данных. На основании параметра 312 типа сообще-

ния определяют тип конфигурационного сообщения 300. Конфигурационное сообщение 300 может быть обработано в соответствии с определенным типом сообщения и определенным фактическим размером части 320 полезных данных.

Обработка конфигурационного сообщения 300 может содержать синтаксический анализ части 320 полезных данных для получения данных 350 конфигурации. В некоторых примерах обработка конфигурационного сообщения 300 содержит отслеживание количества байтов конфигурационного сообщения 300, которые были прочитаны при анализе содержимого части 320 полезных данных. Декодер 110 может обнаруживать ситуации неполного считывания, если имеется определенное расхождение между количеством байтов, считанных при синтаксическом анализе части 320 полезных данных, и фактическим размером части 320 полезных данных, как задано параметром 314 размера полезных данных. Неполное считывание может быть потенциальным источником ошибок при обработке конфигурационных сообщений.

В некоторых примерах параметр 314 размера полезных данных применяют для определения начала следующего конфигурационного сообщения. Следующее конфигурационное сообщение может быть следующим последовательным конфигурационным сообщением по отношению к конфигурационному сообщению 300 в последовательности конфигурационных сообщений, принятых декодером 110. Последовательность конфигурационных сообщений может образовывать часть данных конфигурации. Конфигурационное сообщение 300 и следующее конфигурационное сообщение могут соответствовать одному и тому же изображению или разным изображениям. Например, конфигурационное сообщение 300 может соответствовать первому изображению в последовательности изображений, а следующее конфигурационное сообщение может соответствовать второму, отличающемуся изображению в последовательности изображений.

В некоторых примерах обработка конфигурационного сообщения 300 включает в себя отбрасывание по меньшей мере части содержимого части 320 полезных данных. По меньшей мере, часть содержимого части 320 полезных данных может быть отброшена с помощью параметра 314 размера полезных данных. Например, декодер 110 может перейти к следующему конфигурационному сообщению, причем начало следующего конфигурационного сообщения было определено с помощью параметра 314 размера полезных данных, без обработки по меньшей мере части содержимого части 320 полезных данных.

Таким образом, использование параметра 314 размера полезных данных позволяет декодеру 110 эффективно обрабатывать возможные неполные считывания. Применение параметра 314 размера полезных данных для определения положения начала следующего сообщения также позволяет декодеру 110 поддерживать правильное выравнивание или синхронизацию с потоком байтов, содержащим несколько конфигурационных сообщений. Например, даже если содержимое и/или формат данного конфигурационного сообщения являются нераспознанными, непредусмотренными и/или были обновлены без оповещения об этом декодера 110, декодер 110 все еще способен определить начало следующего конфигурационного сообщения и, таким образом, остается синхронизированным с потоком байтов. Кроме того, если поток байтов становится искаженным, декодер 110 способен выявлять ситуации, когда должно произойти избыточное считывание, и отреагировать на них соответствующим образом. Например, декодер 110 может прервать обработку текущего конфигурационного сообщения и сбросить поток байтов в определенное положение начала следующего конфигурационного сообщения. Таким образом, обработка конфигурационного сообщения 300 с помощью параметра 314 размера полезных данных повышает надежность и/или способность к адаптации декодера 110 по сравнению со случаем, в котором такой параметр не применяют, то есть декодер 110 способен обрабатывать конфигурационные сообщения, которые были изменены и/или не распознаны и/или оказались непредусмотренными. Декодер 110 также способен обрабатывать потенциальные неполные и избыточные считывания без потери синхронизации с потоком байтов, за счет чего повышается способность декодера 110 обрабатывать потенциальные ошибки.

В некоторых примерах обработка конфигурационного сообщения 300 содержит синтаксический анализ первого содержимого части 320 полезных данных. Первое содержимое соответствует форматному размеру части 320 полезных данных. Первое содержимое может содержать данные 350 конфигурации. Первое содержимое может содержать первый набор параметров конфигурации. Обработка конфигурационного сообщения 300 может дополнительно содержать отбрасывание остаточного содержимого части 320 полезных данных с помощью параметра 314 размера полезных данных. Остаточное содержимое части 320 полезных данных содержит тот контент части 320 полезных данных, который не содержится в первом содержимом. Остаточное содержимое содержит дополнительные данные конфигурации. Остаточное содержимое может содержать дополнительный набор параметров конфигурации. Наличие остаточного содержимого может быть определено путем выявления расхождения между форматным размером и фактическим размером части 320 полезных данных. Другими словами, если определено, что фактический размер больше, чем форматный размер, это свидетельствует о том, что в части 320 полезных данных, помимо содержимого, которое предусмотрено, исходя из форматного размера, присутствует дополнительное содержимое.

В позиции 430 первое представление 206 восстанавливают с помощью полученных данных 350 конфигурации. Первое представление 206 может быть восстановлено с помощью остаточных данных 216 в соответствии с полученными данными 350 конфигурации.

На фиг. 5 схематично показан пример конфигурационного сообщения 500. Устройство, например первое устройство 102, содержащее кодер 108, может формировать конфигурационное сообщение 500 и выводить такое конфигурационное сообщение 500 для обработки дополнительным устройством, например, вторым устройством 104, которое содержит декодер 110. Некоторые элементы, изображенные на фиг. 5, аналогичны элементам, показанным на фиг. 3, поэтому для обозначения аналогичных элементов были использованы соответствующие ссылочные позиции, увеличенные на 200.

Конфигурационное сообщение 500 содержит часть 510 заголовка и часть 520 полезных данных. Часть 510 заголовка содержит параметр 512 типа сообщения. Параметр 512 типа сообщения задает тип конфигурационного сообщения 500. Тип сообщения указывает на то, как следует обрабатывать конфигурационное сообщение 500. Часть 510 заголовка содержит параметр 514 размера полезных данных. Параметр 514 размера полезных данных задает фактический размер части 520 полезных данных.

В данном примере часть 520 полезных данных содержит первый фрагмент 522 полезных данных. Первый фрагмент 522 полезных данных содержит первые данные 552 конфигурации, относящиеся к обработке остаточных данных 216. Часть 520 полезных данных также содержит второй фрагмент 524 полезных данных. В этом примере второй фрагмент 524 полезных данных расположен последовательно относительно первого фрагмента 522 полезных данных в части 520 полезных данных. Расположение второго фрагмента 524 полезных данных последовательно относительно первого фрагмента 522 полезных данных может позволить декодеру 110 принять, обработать и/или выполнить синтаксический анализ первого фрагмента 522 полезных данных до того, как будет принят, обработан и/или синтаксически проанализирован второй фрагмент 524 полезных данных. Второй фрагмент 524 полезных данных содержит вторые данные 554 конфигурации, относящиеся к обработке остаточных данных 216.

Первый и второй фрагменты 522, 524 полезных данных вместе образуют часть 520 полезных данных конфигурационного сообщения 500. Следовательно, параметр 514 размера полезных данных указывает суммарный размер первого и второго фрагментов 522, 524 полезных данных.

В некоторых примерах первый фрагмент 522 полезных данных содержит все первые данные 552 конфигурации. В некоторых примерах второй фрагмент 524 полезных данных содержит все вторые данные 554 конфигурации. По существу, первые и вторые данные 552, 554 конфигурации могут храниться в отдельных фрагментах части 520 полезных данных конфигурационного сообщения 500. Сохранение первых данных 552 конфигурации в отдельном фрагменте части 520 полезных данных по отношению ко вторым данным 554 конфигурации способствует эффективной обработке декодером содержимого части 520 полезных данных, в частности, для декодера, который выполнен с возможностью не использовать и/или не обрабатывать одни из первых и вторых данных 552, 554 конфигурации.

В некоторых примерах часть 510 заголовка не задает размер первого фрагмента 522 полезных данных и/или размер второго фрагмента 524 полезных данных. В некоторых примерах размер первого фрагмента 522 полезных данных может быть определен, исходя из форматного размера, указанного параметром 512 типа сообщения. По существу, форматный размер, указанный параметром 512 типа сообщения, может отражать содержимое первого фрагмента 522 полезных данных, но не содержимое второго фрагмента 524 полезных данных.

Одни или оба из первых данных 552 конфигурации и вторых данных 554 конфигурации могут быть применимыми для восстановления входных данных 206 с помощью остаточных данных 216. В некоторых примерах вторые данные 554 конфигурации содержат дополнительные данные конфигурации. Дополнительные данные конфигурации предназначены для дополнения первых данных 552 конфигурации.

Первые данные 552 конфигурации относятся к первому набору из одного или большего количества декодеров. Первый набор из одного или большего количества декодеров имеет первую рабочую характеристику. Примером рабочей характеристики является возможность обработки. Вторые данные 554 конфигурации относятся ко второму набору из одного или большего количества декодеров. Второй набор из одного или большего количества декодеров имеет вторую рабочую характеристику. Декодер 110, на который передается конфигурационное сообщение 500, может содержаться в первом наборе или втором наборе. Кодер 108, который формирует конфигурационное сообщение 500, может не иметь информации, в каком наборе декодеров содержится декодер 110. Кодер 108 может передавать конфигурационное сообщение 500 множеству декодеров через оба набора декодеров.

В некоторых примерах первая рабочая характеристика соответствует первой версии, согласно которой первый набор из одного или большего количества декодеров настроен для работы, а вторая рабочая характеристика соответствует второй, отличающейся версии, согласно которой второй набор из одного или большего количества декодеров настроен для работы. Вторая версия может быть более поздней по отношению к первой версии. Первая и вторая версии могут быть разными версиями по структуре синтаксиса, которая определяет, как должны обрабатываться конфигурационные сообщения, и/или которая определяет содержимое таких конфигурационных сообщений. Следовательно, когда синтаксис обновлен от первой версии до второй версии, к конфигурационному сообщению, которое содержит первые данные 552 конфигурации из первой версии, добавляются вторые данные 554 конфигурации. Вторые данные 554 конфигурации могут, например, быть добавлены к первым данным 552 конфигурации.

Добавление вторых данных 554 конфигурации к конфигурационному сообщению существующего

типа сообщения вместо создания нового типа сообщения для размещения вторых данных 554 конфигурации уменьшает объем данных, которые хранятся, передаются и/или обрабатываются, по сравнению со случаем, когда формируются новые типы сообщений. Путем изменения конфигурации существующих типов сообщений вместо определения новых типов сообщений может быть уменьшено общее количество конфигурационных сообщений, используемых для передачи всех соответствующих данных конфигурации. Каждое конфигурационное сообщение имеет соответствующий заголовок, который, например, может иметь длину по меньшей мере один байт. Следовательно, за счет уменьшения общего количества конфигурационных сообщений может быть уменьшен объем данных, которые хранятся, передаются и/или обрабатываются.

В некоторых примерах декодер 110 не может обрабатывать и первый, и второй фрагменты 522, 524 полезных данных. В одном случае кодер 108 может подвергнуться обновлению, в соответствии с которыми вторые данные 554 конфигурации включены в конфигурационное сообщение 500, но декодер 110 может не подвергнуться такому обновлению. Другими словами, кодер 108 и декодер 110 могут работать согласно отличающимся версиям синтаксиса. В таких примерах может существовать расхождение в интерпретации форматного размера части 520 полезных данных кодирующим устройством 108 и декодером 110. Например, форматный размер согласно кодирующему устройству 108, в котором учтено дополнительное содержимое части 520 полезных данных и обновленный формат полезных данных после обновления версии, отличается от форматного размера согласно декодеру 110, поскольку декодер 110 не учитывает дополнительное содержимое и обновленный формат полезных данных и работает в соответствии с предыдущей версией. Для кодера 108 форматный размер равен фактическому размеру обновленной части 520 полезных данных, но для декодера 110 это не так. Следовательно, один только форматный размер может быть ненадежным индикатором фактического размера части 520 полезных данных. Таким образом, включение параметра 514 размера полезных данных, который определяет фактический размер части 520 полезных данных, способствует более надежной обработке в декодере 110, как более подробно описано ниже.

На фиг. 6 показан пример способа 600 обработки данных конфигурации. Способ 600 может быть осуществлен устройством, содержащим декодер, таким как второе устройство 104, описанное выше.

В позиции 610 принято конфигурационное сообщение 500.

В позиции 620 конфигурационное сообщение 500 обрабатывается для получения одних или обоих из первых данных 552 конфигурации и вторых данных 554 конфигурации.

В некоторых примерах, например, если декодер 110 находится в первом наборе декодеров, первый фрагмент 522 полезных данных синтаксически анализируется декодером 110, а для второго фрагмента 524 полезных данных синтаксический анализ декодером 110 не производится. Синтаксический анализ первого фрагмента 522 полезных данных, но не второго фрагмента 524 полезных данных позволяет декодеру 110 получить первые данные 552 конфигурации, но не вторые данные 554 конфигурации. В некоторых примерах, например, если декодер 110 находится во втором наборе декодеров, второй фрагмент 524 полезных данных синтаксически анализируется декодером 110, а для первого фрагмента 522 полезных данных синтаксический анализ декодером 110 не производится. Синтаксический анализ второго фрагмента 524 полезных данных, но не первого фрагмента 522 полезных данных позволяет декодеру 110 получить вторые данные 554 конфигурации, но не первые данные 552 конфигурации. В некоторых примерах, например, если декодер 110 находится во втором наборе декодеров, осуществляется синтаксический анализ как первого фрагмента 522 полезных данных, так и второго фрагмента 524 полезных данных для получения как первых данных 552 конфигурации, так и вторых данных 554 конфигурации.

Параметр 514 размера полезных данных может применяться для определения начала следующего конфигурационного сообщения. В некоторых примерах, например, если декодер 110 находится в первом наборе декодеров, содержимое второго фрагмента 524 полезных данных отбрасывается. Содержимое второго фрагмента 524 полезных данных содержит вторые данные 554 конфигурации. Содержимое второго фрагмента 524 полезных данных может быть отброшено с помощью общего размера части 520 полезных данных, указанного в параметре 514 размера полезных данных. Например, декодер 110 может считать содержимое первого фрагмента 522 полезных данных, а затем перейти к следующему конфигурационному сообщению, начало которого было локализовано с помощью параметра 514 размера полезных данных, без считывания содержимого второго фрагмента 524 полезных данных.

Применение параметра 514 размера полезных данных для определения начала следующего сообщения позволяет декодеру 110 оставаться синхронизированным с потоком байтов, содержащим несколько конфигурационных сообщений. Таким образом, декодер 110 может обработать конфигурационное сообщение 500 и определить начало следующего конфигурационного сообщения независимо от того, находится ли декодер 110 в первом наборе декодеров или во втором наборе декодеров.

В позиции 630 восстанавливается первое представление 206 изображения с относительно высоким уровнем качества. Первое представление 206 восстанавливается с помощью одних или обоих из первых данных 552 конфигурации и вторых данных 554 конфигурации, которые получены в позиции 620. В случае, когда декодер 110 получает как первые, так и вторые данные 552, 554 конфигурации, первое представление 206 изображения восстанавливают с помощью как первых, так и вторых данных 552, 554 кон-

фигурации. В случае, когда декодер 110 обрабатывает один из первого и второго фрагментов 522, 524 полезных данных, но не другой из первого и второго фрагментов 522, 524 полезных данных, первое представление 206 изображения может быть восстановлено с помощью одних из полученных первых и вторых данных 552, 554 конфигурации. По существу, применение конфигурационного сообщения 500 для восстановления первого представления 206 может не зависеть от типа декодера и/или номера версии, в соответствии с которой декодер 110 настроен для работы.

В некоторых примерах в конфигурационном сообщении 500 выполняют побайтовую обработку. При побайтовой обработке байты данных обрабатывают поочередно. Побайтовая обработка данных конфигурации может быть более эффективной, чем побитовая обработка данных конфигурации. Побитовая обработка может предусматривать отслеживание того, какой бит в заданном байте является следующим битом для чтения или записи. Когда группа битов, представляющая данный параметр, захватывает границу байта, может быть выполнена дополнительная обработка, чтобы осуществить считывание группы битов и получить данный параметр. Такая обработка может быть сокращена в случае, когда осуществляется побайтовая обработка. В некоторых примерах при побайтовой обработке данных количество циклов выполнения может быть уменьшено, если сравнивать с побитовой обработкой.

В некоторых примерах данные конфигурации записывают в память. Если данные конфигурации обрабатываются побайтно, то при хранении данных конфигурации в памяти и/или при извлечении данных конфигурации из памяти может потребоваться меньший объем обработки по сравнению со случаем, когда данные конфигурации обрабатываются побитно. Наименьшая единица адресуемой памяти может содержать один байт. То есть информация может храниться в памяти побайтно. Соответственно, если данные конфигурации упакованы в последовательность байтов, для сохранения данных конфигурации в памяти и/или для извлечения данных конфигурации из памяти может потребоваться меньше этапов. Объем заполнения, применяемый для упаковки данных конфигурации в адресуемые блоки памяти, также может быть уменьшен путем внесения данных конфигурации в память в побайтовой форме.

Кроме того, передача данных конфигурации через сеть, такую как Интернет, может быть выполнена более эффективно, если данные конфигурации скомпонованы в последовательность байтов. Данные можно передавать через Интернет посредством протокола управления передачей данных (TCP). TCP действует на транспортном уровне стека протокола. TCP получает данные из потока, упаковывает данные в сегменты TCP и применяет Интернет-модуль, например Интернет-протокол (IP), для передачи получателю каждого сегмента TCP через Интернет. Поток данных может быть получен из памяти. Сегмент TCP может содержать целое число байтов. То есть если TCP получает данные, которые должны быть упакованы в сегменты TCP, из побайтного потока, для этого может потребоваться меньший объем обработки по сравнению с побитовым потоком. Объем заполнения, применяемый для упаковки данных конфигурации в сегменты TCP, также может быть уменьшен путем внесения данных конфигурации в TCP в побайтовой форме.

На фиг. 7 показана блок-схема примера устройства 700.

В примере устройство 700 содержит кодирующее устройство. В другом примере устройство 700 содержит декодер.

Примеры устройства 700 включают в себя, помимо прочего, мобильный компьютер, персональную компьютерную систему, беспроводное устройство, базовую станцию, телефонное устройство, настольный компьютер, переносной компьютер, ноутбук, нетбук, универсальную компьютерную систему, карманный компьютер, рабочую станцию, сетевой компьютер, сервер приложений, запоминающее устройство, бытовое электронное устройство, такое как фотокамера, видеокамера, мобильное устройство, игровая консоль, портативное игровое устройство, или вообще любой тип вычислительного или электронного устройства.

В данном примере устройство 700 содержит один или более процессоров 701, выполненных с возможностью обработки информации и/или команд. Один или более процессоров 701 могут содержать центральный процессор (ЦП). Один или более процессоров 701 сопряжены с шиной 702. Операции, выполняемые одним или более процессоров 701, могут быть осуществлены аппаратными средствами и/или программным обеспечением. Один или более процессоров 701 могут содержать несколько совместно расположенных процессоров или несколько отдельно расположенных процессоров.

В этом примере устройство 700 содержит пригодную для использования в компьютере энергозависимую память 703, выполненную с возможностью хранения информации и/или команд для одного или большего количества процессоров 701. Энергозависимая память 703, пригодная для использования в компьютере, сопряжена с шиной 702. Энергозависимая память 703, пригодная для использования в компьютере, может содержать оперативное запоминающее устройство (ОЗУ).

В этом примере устройство 700 содержит энергонезависимую память 704, пригодную для использования в компьютере, выполненную с возможностью хранения информации и/или команд для одного или большего количества процессоров 701. Энергонезависимая память 704, пригодная для использования в компьютере, сопряжена с шиной 702. Энергонезависимая память 704, пригодная для использования в компьютере, может содержать постоянное запоминающее устройство (ПЗУ).

В данном примере устройство 700 содержит один или более устройств 705 для хранения данных,

выполненных с возможностью хранения информации и/или команд. Одно или более устройств 705 для хранения данных сопряжены с шиной 702. Одно или более устройств 705 для хранения данных могут, например, содержать магнитный или оптический диск и дисковод или твердотельный накопитель (SSD - англ.: solid-state drive).

В этом примере устройство 700 содержит одно или более устройств 706 ввода/вывода (I/O), выполненных с возможностью передачи информации в один или более процессоров 701 и/или от них. Одно или более устройств 706 ввода/вывода (I/O) сопряжены с шиной 702. Одно или более устройств 706 ввода/вывода (I/O) содержат по меньшей мере один сетевой интерфейс. Такой по меньшей мере один сетевой интерфейс может позволять устройству 700 осуществлять связь через одну или более сетей передачи данных. Примеры сетей передачи данных включают в себя, помимо прочего, Интернет и локальную компьютерную сеть (LAN). Одно или более устройств 706 ввода-вывода могут позволить пользователю предоставлять входные данные в устройство 700 посредством одного или большего количества устройств ввода (не показаны). Одно или более устройств ввода могут включать в себя, например, пульт дистанционного управления, одну или более физических кнопок и т. д. Одно или более устройств 706 ввода/вывода могут обеспечивать возможность предоставления информации пользователю через одно или более устройств вывода (не показаны). Одно или более устройств вывода могут, например, включать в себя экран дисплея.

Для устройства 700 изображены разнообразные другие модули. Примером могут служить, если присутствуют, операционная система 707, модуль 708 обработки сигналов данных, один или более дополнительных модулей 709 и данных 710, которые показаны как единый блок, или их объединение с энергозависимой памятью 703, пригодной для использования в компьютере, энергонезависимой памятью 704, пригодной для использования в компьютере, и с одним или более модулей 705 хранения данных. Модуль 708 обработки сигналов данных может быть реализован посредством компьютерного программного кода, который хранится в ячейках памяти в пределах используемой компьютером энергонезависимой памяти 704, машиночитаемых носителей данных в одном или более модулей 705 хранения данных и/или других материальных машиночитаемых носителей. Примеры материальных машиночитаемых носителей информации включают в себя, помимо прочего, оптический носитель (например, CD-ROM, DVD-ROM или Blu-ray), модуль флэш-памяти, дискету, жесткий диск или любой другой носитель, способный хранить машиночитаемые команды, такие как микропрограммное обеспечение или микрокод по меньшей мере в одной микросхеме ПЗУ, ОЗУ или программируемого ПЗУ (ППЗУ) или в специализированной интегральной схеме (ASIC).

Соответственно, устройство 700 может содержать модуль 708 обработки сигналов данных, который может выполняться одним или более процессоров 701. Модуль 708 обработки сигнала данных может быть выполнен с возможностью включения в себя команд для реализации по меньшей мере некоторых операций, описанных в настоящем документе. Во время работы один или более процессоров 701 запускают, осуществляют, исполняют, интерпретируют или иным образом выполняют команды в модуле 708 обработки сигналов.

Хотя по меньшей мере некоторые аспекты примеров, описанных в настоящем документе со ссылкой на чертежи, содержат компьютерные процессы, выполняемые в системах обработки или процессорах, примеры, описанные в настоящем документе, также распространяются на компьютерные программы, например компьютерные программы на или в носителе, выполненном с возможностью практической реализации таких примеров.носителем может быть любой объект или устройство, способное переносить программу.

Следует понимать, что устройство 700 может содержать больше или меньше компонентов и/или компоненты, отличающиеся от тех, что изображены на фиг. 7.

Устройство 700 может быть расположено в одном месте или может быть распределено в нескольких местоположениях. Такие местоположения могут быть локальными или удаленными.

Описанные в настоящем документе технологии могут быть реализованы в виде программного или аппаратного обеспечения или могут быть реализованы с помощью комбинации программных и аппаратных компонентов. Они могут включать в себя настройку устройства для осуществления и/или поддержки любой технологии или всех технологий, описанных в настоящем документе.

Следует понимать, что любой признак, описанный в отношении любого варианта осуществления, может применяться отдельно или в комбинации с другими описанными признаками, а также может применяться в сочетании с одним или несколькими признаками любого другого варианта осуществления или любой комбинацией любых других вариантов осуществления. Кроме того, без отклонения от объема изобретения, который ограничен прилагаемой формулой изобретения, также могут быть применены не описанные выше эквиваленты и модификации.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ формирования конфигурационного сообщения для обеспечения возможности декодеру восстанавливать первое представление изображения с первым уровнем качества с помощью данных конфигурации, остаточных данных и второго представления изображения с первым уровнем качества, выполняемый устройством, содержащим кодер, причем способ включает в себя:

получение остаточных данных, основанных на первом представлении изображения с первым уровнем качества и втором представлении изображения с первым уровнем качества, при этом первое представление и второе представление находятся в многоуровневой иерархии, имеющей несколько уровней качества,

причем второе представление основано на представлении изображения со вторым, более низким уровнем качества в многоуровневой иерархии, причем второе представление получают путем повышающей дискретизации представления изображения со вторым, более низким, уровнем качества, при этом остаточные данные получают путем вычитания элемента изображения во втором представлении изображения из соответствующего элемента изображения в первом представлении изображения, и остаточные данные подходят для применения декодером для восстановления первого представления с помощью второго представления;

получение данных конфигурации, относящихся к обработке остаточных данных, причем данные конфигурации подходят для применения декодером для восстановления первого представления изображения;

формирование конфигурационного сообщения, содержащего заголовок и полезные данные, причем полезные данные конфигурационного сообщения содержат данные конфигурации, а заголовок конфигурационного сообщения содержит:

параметр типа сообщения, который задает тип сообщения конфигурационного сообщения и указывает на то, что полезные данные имеют заданный формат полезных данных, в котором полезные данные, имеющие указанный формат полезных данных, имеют форматный размер; и

параметр размера полезных данных, который задает фактический размер полезных данных; и вывод конфигурационного сообщения для обработки декодером, чтобы обеспечить декодеру возможность восстановления первого представления с помощью данных конфигурации, остаточных данных и второго представления.

2. Способ по п.1, в котором форматный размер и фактический размер являются одинаковыми.

3. Способ по п.1, в котором форматный размер и фактический размер являются отличающимися.

4. Способ по любому из пп.1-3, в котором тип сообщения, заданный параметром типа сообщения, указывает, как должно быть обработано конфигурационное сообщение.

5. Способ по любому из пп.1-4, в котором длина конфигурационного сообщения представляет собой целое число байтов.

6. Способ по любому из пп.1-5, в котором фактический размер полезных данных представляет собой целое число байтов.

7. Способ по любому из пп.1-6, в котором значение параметра размера полезных данных хранится в элементе фиксированной длины.

8. Способ по п.7, в котором значение параметра размера полезных данных хранится в элементе фиксированной длины, который содержит три бита в однобайтовой части заголовка.

9. Способ по п.7 или 8, в котором параметр размера полезных данных содержит ссылку на таблицу соответствия размера полезных данных, в которой хранятся возможные значения фактического размера полезных данных.

10. Способ по п.9, в котором для получения фактического размера полезных данных из таблицы соответствия размера полезных данных применяется указанное значение.

11. Способ по любому из пп.1-6, в котором значение параметра размера полезных данных имеет переменную длину в битах, и причем значение параметра размера полезных данных хранится в элементе переменной длины, содержащем целое число байтов.

12. Способ по п.11, в котором элемент переменной длины содержит по меньшей мере один заданный байт, имеющий один или более заранее определенных битов, выполненных с возможностью указания того, содержит ли элемент переменной длины один или более дополнительных байтов для по меньшей мере одного заданного байта.

13. Способ по п.11 или 12, в котором значение не является одним из множества значений, заданных в таблице соответствия размера полезных данных.

14. Способ по любому из пп.1-13, в котором заголовок содержит только параметр типа сообщения и параметр размера полезных данных.

15. Способ восстановления первого представления изображения с первым уровнем качества с помощью конфигурационного сообщения, остаточных данных и второго представления изображения с первым уровнем качества, выполняемый устройством, содержащим декодер, причем способ включает в

себя:

прием конфигурационного сообщения, содержащего заголовок и полезные данные, причем полезные данные содержат данные конфигурации, относящиеся к обработке остаточных данных, применимых для восстановления первого представления изображения с первым уровнем качества в многоуровневой иерархии, имеющей несколько уровней качества, с помощью второго представления изображения с первым уровнем качества,

причем второе представление основано на представлении изображения со вторым, более низким, уровнем качества в многоуровневой иерархии, причем второе представление получают путем повышающей дискретизации представления изображения со вторым, более низким, уровнем качества, при этом остаточные данные получают путем вычитания элемента изображения во втором представлении изображения из соответствующего элемента изображения в первом представлении изображения,

причем заголовок содержит:

параметр типа сообщения, который задает тип сообщения конфигурационного сообщения и указывает на то, что полезные данные имеют заданный формат полезных данных, в котором полезные данные, имеющие указанный формат полезных данных, имеют форматный размер; и

параметр размера полезных данных, который задает фактический размер полезных данных;

обработку конфигурационного сообщения с помощью параметра типа сообщения и параметра размера полезных данных для получения данных конфигурации; и

восстановление первого представления с помощью полученных данных конфигурации, остаточных данных и второго представления.

16. Способ по п.15, в котором обработка конфигурационного сообщения содержит:

определение фактического размера полезных данных на основе параметра размера полезных данных;

определение типа сообщения на основе параметра типа сообщения; и

обработку конфигурационного сообщения в соответствии с определенным типом сообщения и определенным фактическим размером полезных данных.

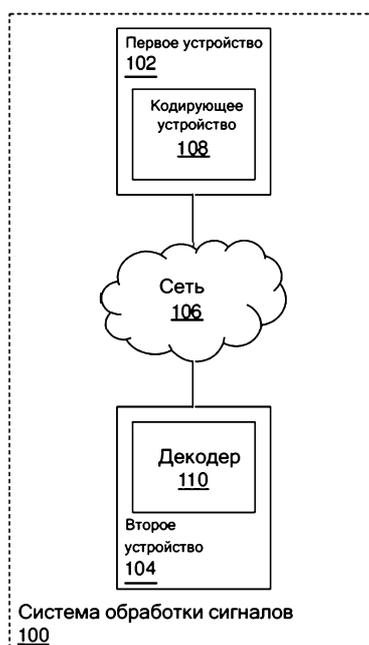
17. Способ по п.15 или 16, в котором форматный размер и фактический размер являются одинаковыми.

18. Способ по п.15 или 16, в котором форматный размер и фактический размер являются отличающимися.

19. Способ по любому из пп.15-18, включающий в себя: прием остаточных данных.

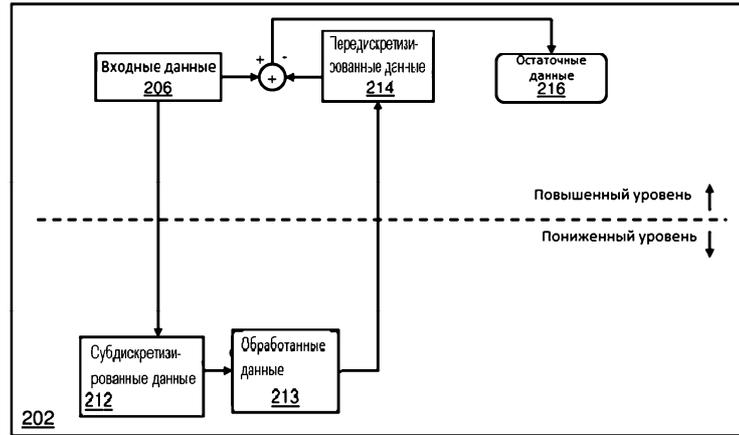
20. Вычислительное устройство для формирования конфигурационного сообщения для обеспечения возможности декодеру восстанавливать первое представление изображения с первым уровнем качества с помощью данных конфигурации, выполненное с возможностью осуществления способа по любому из пп.1-19.

21. Машиночитаемый носитель, содержащий компьютерную программу, содержащую команды, исполняемые вычислительным устройством по п.20 для осуществления операций способа по любому из пп.1-19.



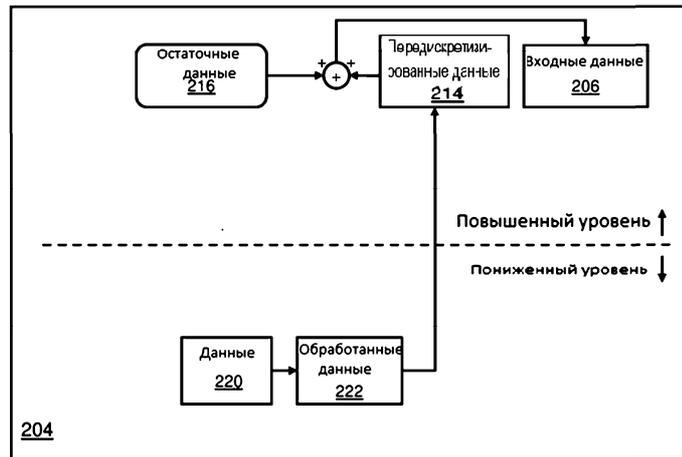
Фиг. 1

200



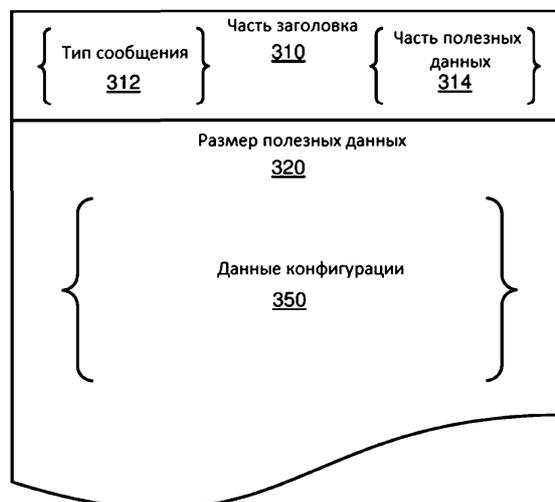
Фиг. 2А

200



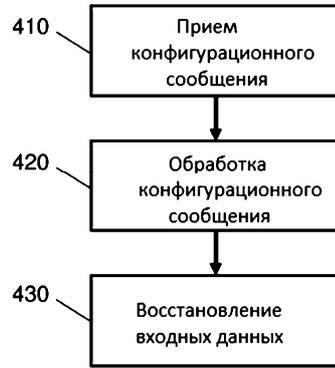
Фиг. 2В

300



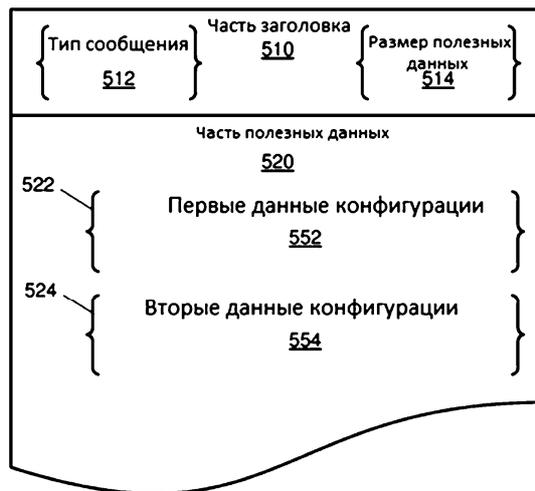
Фиг. 3

400 ↘



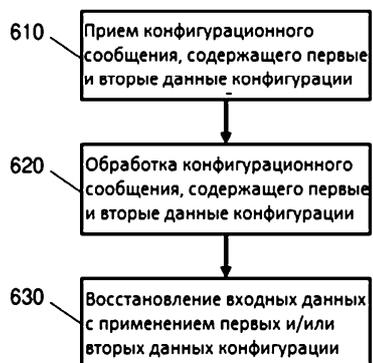
Фиг. 4

500 ↘

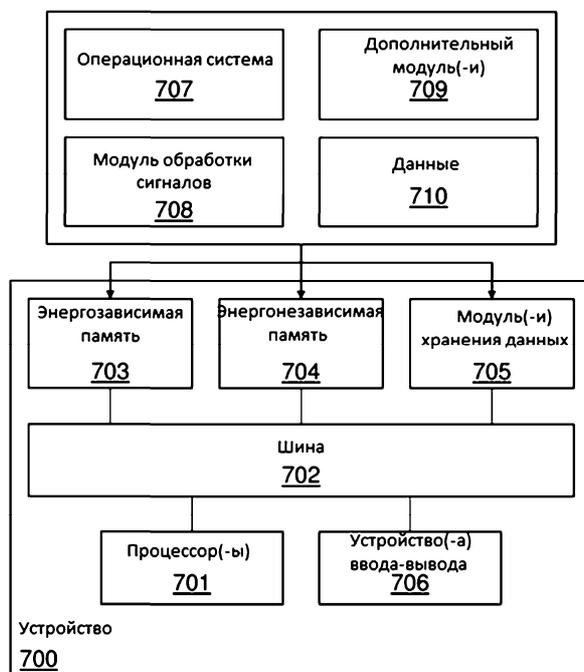


Фиг. 5

600



Фиг. 6



Фиг. 7