

(19)



**Евразийское  
патентное  
ведомство**

(11) **047124**(13) **B1**(12) **ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ**

(45) Дата публикации и выдачи патента  
**2024.06.04**

(51) Int. Cl. **H04N 19/30** (2014.01)  
**H04N 19/31** (2014.01)

(21) Номер заявки  
**202191199**

(22) Дата подачи заявки  
**2019.10.30**

---

(54) **СПОСОБ, УСТРОЙСТВО И МАШИНОЧИТАЕМЫЙ НОСИТЕЛЬ, СОДЕРЖАЩИЙ КОМАНДЫ ОБРАБОТКИ ИНФОРМАЦИИ О КОНФИГУРАЦИИ ДЛЯ ОБРАБОТКИ ОСТАТОЧНЫХ ДАННЫХ ПРИ ОБРАБОТКЕ ИЗОБРАЖЕНИЙ**

---

(31) **1817784.0**

(32) **2018.10.31**

(33) **GB**

(43) **2021.08.12**

(86) **PCT/GB2019/053070**

(87) **WO 2020/089618 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-A1-2018115778**

**WO-A1-2018046940**

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/sealable-hevc-encoding-how-to-setup-cfg-files-for-quality-scalability> [retrieved on 2020-01-21] page 2 - page 4

ROSEWARNE C ET AL: "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Update 2 of Encoder Description", 20. JCT-VC MEETING; 10-2-2105 - 18-2-2015; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: [HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/](http://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/), no. JCTVC-T1002, 2 June 2015 (2015-06-02), XP030117409, page 9; tables 33-2 page 11, paragraph 4.1.4

(57) Для заданного изображения в наборе изображений получают остаточные данные, которые могут быть использованы декодером для восстановления первого представления заданного изображения с первым уровнем качества с использованием второго представления заданного изображения с первым уровнем качества. Второе представление основано на представлении заданного изображения со вторым, более низким уровнем качества. Данные конфигурации, относящиеся к обработке остаточных данных, генерируются и выводятся для обработки с помощью декодера. Данные конфигурации содержат параметр (754, 854) временной обработки, который определяет степень временной обработки, связанной с восстановлением, для заданного изображения, первого представления с использованием второго представления и остаточных данных, причем выполнение временной обработки включает использование данных, основанных на множестве изображений в наборе изображений.

**047124**  
**B1**

**047124**  
**B1**

### **Область техники**

Настоящее изобретение относится к способам, устройствам, компьютерным программам и машиночитаемым носителям. В частности, но не исключительно, настоящее изобретение относится к способам, устройствам, компьютерным программам и машиночитаемым носителям, предназначенным для использования при обработке информации о конфигурации, которая относится к остаточным данным, пригодным для использования при восстановлении представления изображения с относительно высоким уровнем качества.

### **Уровень техники**

Сжатие и распаковка сигналов используются во многих известных системах. Сигналы многих типов, например, видеосигналы, могут быть сжаты и закодированы для передачи, например, по сети передачи данных. При декодировании такого сигнала может быть желательным повысить уровень качества сигнала и/или восстановить как можно больше информации, содержащейся в исходном сигнале.

В некоторых известных системах используют способы масштабируемого кодирования. Масштабируемое кодирование включает кодирование сигнала вместе с информацией, позволяющей восстановить сигнал с одним или более различными уровнями качества, например, в зависимости от возможностей декодера и доступной ширины полосы.

Приведем некоторые соображения, касающиеся восстановления сигналов в масштабируемой системе кодирования. Одно из таких соображений относится к объему хранимой, используемой и/или передаваемой информации. Количество информации может варьироваться, например, в зависимости от требуемого уровня качества восстановленного сигнала, характера информации, которая используется при восстановлении, и/или того, как такая информация сконфигурирована.

Еще одно соображение состоит в способности декодера точно и/или надежно обрабатывать полученную информацию. Одним из факторов, который может влиять на надежность декодера, является его способность обрабатывать ошибки и/или обрабатывать принятую информацию, которая является незапланированной, измененной и/или нераспознанной декодером. Другой фактор точности и/или надежности, с которой декодер может выполнять восстановление, связан с предоставлением декодеру актуальной информации, указывающей, как следует восстановить сигнал.

Еще одно соображение состоит в способности кодера и/или декодера эффективно обрабатывать информацию. Эффективность, с которой кодер и/или декодер обрабатывает информацию, может представлять собой фактор уровня производительности кодера и/или декодера.

### **Раскрытие сущности изобретения**

Различные аспекты настоящего изобретения изложены в прилагаемой формуле изобретения.

Дополнительные признаки и преимущества станут очевидными из нижеследующего описания предпочтительных вариантов осуществления, приведенных исключительно в качестве примера, со ссылкой на прилагаемые чертежи.

### **Краткое описание чертежей**

На фиг. 1 представлена структурная схема примера системы обработки сигналов в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 2А и 2В представлена структурная схема еще одного примера системы обработки сигналов в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 3 представлена упрощенная схема примера способа обработки данных в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 4 представлена упрощенная схема примера данных точки входа в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 5 представлена упрощенная схема примера сообщения с конфигурацией в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 6 представлена блок-схема, изображающая пример способа в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 7 представлена упрощенная схема еще одного примера сообщения с конфигурацией в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 8 представлена упрощенная схема еще одного примера сообщения с конфигурацией в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 9 представлена упрощенная схема еще одного примера сообщения с конфигурацией в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 10 представлена блок-схема, изображающая пример способа в соответствии с вариантом осуществления настоящего изобретения;

на фиг. 11 представлена блок-схема, изображающая пример еще одного способа в соответствии с вариантом осуществления настоящего изобретения; и

на фиг. 12 представлена структурная схема примера устройства в соответствии с вариантом осуществления настоящего изобретения.

### **Осуществление изобретения**

На фиг. 1 показан пример системы 100 обработки сигналов. Систему 100 обработки сигналов ис-

пользуют для обработки сигналов. Примеры типов сигналов включают, среди прочих, видеосигналы, сигналы изображения, аудиосигналы, волюметрические сигналы, например, используемые при медицинской, научной или голографической визуализации, 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 быть взаимосвязаны как клиент и сервер, причем 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. Один или более компонентов могут быть расположены в одном месте или могут быть расположены на расстоянии друг от друга в системе 100 обработки сигналов. Примеры типов устройств включают, среди прочих, компьютеризированные устройства, карманные или портативные 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 обмена данными со вторым устройством 104 посредством сети 106 передачи данных. Примеры сети 106 передачи данных включают, среди прочих, Интернет, локальную сеть (LAN) и глобальную сеть (WAN). 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 в дальнейшем упоминаются как "данные сигнала". Например, если сигнал представляет собой видеосигнал, кодер 108 выполнен с возможностью кодирования видеоданных. Видеоданные содержат последовательность из множества изображений или кадров. Кодер 108 может выполнять одну или более дополнительных функций в дополнение к кодированию данных сигнала. 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 и кодер 108 выполнены в виде отдельных устройств. В таких примерах первое устройство 102 соединено с возможностью обмена данными с кодером 108. Первое устройство 102 может быть реализовано в виде одной или более программных функций и/или одного или более аппаратных модулей.

Второе устройство 104 содержит декодер 110. Декодер 110 выполнен с возможностью декодирования данных сигнала. Декодер 110 может выполнять одну или более дополнительных функций в дополнение к декодированию данных сигнала. 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 и декодер 110 выполнены в виде отдельных устройств. В таких примерах второе устройство 104 соединено с возможностью обмена данными с декодером 110. Второе устройство 104 может быть реализовано в виде одной или более программных функций и/или одного или более аппаратных модулей.

Кодер 108 кодирует данные сигнала и передает закодированные данные сигнала на декодер 110 по сети 106 передачи данных. Декодер 110 декодирует принятые закодированные данные сигнала и генерирует декодированные данные сигнала. Декодер 110 может выводить декодированные данные сигнала или данные, полученные с использованием декодированных данных сигнала. Например, декодер 110 может выводить такие данные для их отображения на одном или более устройствах отображения, связанных со вторым устройством 104.

В некоторых описанных в настоящем документе примерах кодер 108 передает на декодер 110 представление сигнала с заданным уровнем качества и информацию, которую декодер 110 может использовать для восстановления представления сигнала с одним или более повышенными уровнями качества. Такая информация может упоминаться как "данные для восстановления". В некоторых примерах "восстановление" представления включает получение представления, которое не является точной копией исходного представления. Степень, в которой представление совпадает с исходным представлением, может зависеть от различных факторов, включая, среди прочего, уровни квантования. Представление сигнала с заданным уровнем качества может считаться воспроизведением, версией или отображением данных, содержащихся в сигнале, с заданным уровнем качества. В некоторых примерах данные для восстановления включают в данные сигнала, которые кодируют с помощью кодера 108 и передают на декодер 110. Например, данные для восстановления могут иметь форму метаданных. В некоторых примерах данные для восстановления кодируют и передают отдельно от данных сигнала.

Информация, которую декодер 110 использует для восстановления представления сигнала с одним или более повышенными уровнями качества, может содержать остаточные данные, как более подробно описано ниже. Остаточные данные представляют собой пример данных для восстановления. Информация, которую декодер 110 использует для восстановления представления сигнала с одним или более повышенными уровнями качества, также может содержать данные конфигурации, относящиеся к обработке остаточных данных. Данные конфигурации могут указывать, как остаточные данные были обработаны кодером 108 и/или как остаточные данные должны быть обработаны декодером 110. Данные конфигура-

ции могут быть переданы на декодер 110, например, в форме метаданных.

На фиг. 2А и 2В схематично показан пример системы 200 обработки сигналов. Система 200 обработки сигналов включает в себя первое устройство 202, содержащее кодер, и второе устройство 204, содержащее декодер. В каждом из первого устройства 202 и второго устройства 204 элементы показаны на двух логических уровнях. Указанные два уровня разделены пунктирной линией. Элементы первого, наивысшего уровня относятся к данным с относительно высоким уровнем качества. Элементы второго, самого низкого уровня относятся к данным с относительно низким уровнем качества. Относительно высокий и относительно низкий уровни качества относятся к многоуровневой иерархии, имеющей множество уровней качества. В некоторых примерах многоуровневая иерархия включает более двух уровней качества. В таких примерах первое устройство 202 и второе устройство 204 могут включать более двух разных уровней. Может существовать один или более других уровней, более высоких и/или более низких, чем уровни, изображенные на фиг. 2А и 2В.

Как показано на фиг. 2А, первое устройство 202 получает первое представление изображения с относительно высоким уровнем качества 206. Представление заданного изображения представляет собой представление данных, содержащихся в изображении. Изображение может представлять собой заданный кадр видео. Первое представление изображения с относительно высоким уровнем качества 206 в дальнейшем будет упоминаться как "входные данные", поскольку в этом примере они представляют собой данные, предоставляемые в качестве входных данных на кодер в первом устройстве 202. Первое устройство 202 может принимать входные данные 206. Например, первое устройство 202 может принимать входные данные 206 по меньшей мере от одного другого устройства. Первое устройство 202 может быть выполнено с возможностью приема следующих друг за другом частей входных данных 206, например, следующих друг за другом кадров видео, и выполнения операций, описанных в настоящем документе, 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 в порядке 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 операции понижающей дискретизации входных данных 206, поэтому в дальнейшем они будут упоминаться как "данные с пониженной дискретизацией". В других примерах данные 212 получают путем выполнения операции, отличной от операции понижающей дискретизации входных данных 206.

В этом примере данные 212 с пониженной дискретизацией обрабатывают для генерации обработанных данных 213 с относительно низким уровнем качества. В других примерах данные 212 с пониженной дискретизацией не обрабатывают с относительно низким уровнем качества. Таким образом, первое устройство 202 выполнено с возможностью генерации данных с относительно низким уровнем качества, причем данные с относительно низким уровнем качества включают данные 212 с пониженной дискретизацией или обработанные данные 213.

В некоторых примерах генерация обработанных данных 213 включает кодирование данных 212 с пониженной дискретизацией. В результате кодирования данных 212 с пониженной дискретизацией получают закодированное изображение с относительно низким уровнем качества. Первое устройство 202 выполнено с возможностью вывода закодированного изображения, например, для передачи на второе устройство 204. Последовательность закодированных изображений, например, формирование закодированного видео, в качестве вывода для передачи на второе устройство 204 может упоминаться как "основной" поток. Создание закодированного изображения может происходить не на первом устройстве 202, а на устройстве кодирования, выполненном отдельно от первого устройства 202. Закодированное изображение может представлять собой часть видео, закодированного с применением стандарта H.264. Генерация обработанных данных 213 может, например, включать генерацию последовательных кадров видео в качестве выходных данных отдельного устройства кодирования, такого как видеокодер H.264. Промежуточный набор данных для генерации обработанных данных 213 может содержать выходные данные такого кодера в отличие от каких-либо промежуточных данных, генерируемых отдельным устройством кодирования.

Генерация обработанных данных 213 с относительно низким уровнем качества может дополнительно включать декодирование закодированного изображения с относительно низким уровнем качества. Операция декодирования может выполняться для эмуляции операции декодирования на втором устройстве 204, как станет очевидно далее. Декодирование закодированного изображения позволяет получить декодированное изображение с относительно низким уровнем качества. В некоторых примерах первое устройство 202 декодирует закодированное изображение с относительно низким уровнем качества с получением декодированного изображения с относительно низким уровнем качества. В других примерах первое устройство 202 принимает декодированное изображение с относительно низким уровнем качества, например, от устройства кодирования и/или декодирования, выполненного отдельно от первого устройства 202. Закодированное изображение может быть декодировано с использованием декодера H.264. Декодирование с помощью отдельного устройства декодирования может включать ввод закодированного видео, такого как поток закодированных данных, выполненный с возможностью передачи на удаленное

устройство декодирования, на отдельный функциональный декодер, реализованный совместно с первым устройством 202 для генерации последовательных декодированных кадров видео. Таким образом, обработанные данные 213 могут содержать кадр видеоданных, сгенерированный с применением сложного нелинейного процесса кодирования и декодирования, причем указанные процессы кодирования и декодирования могут включать моделирование пространственно-временных корреляций в соответствии с конкретным стандартом кодирования, таким как H.264. Однако, поскольку выходной сигнал любого кодера подается на соответствующий декодер, эта сложная обработка по существу скрыта от первого устройства 202.

В одном примере генерация обработанных данных 213 с относительно низким уровнем качества дополнительно включает получение данных коррекции, основанных на сравнении между данными 212 с пониженной дискретизацией и декодированным изображением, полученным первым устройством 202, например, основанных на разности между данными 212 с пониженной дискретизацией и декодированным изображением. Данные коррекции могут быть использованы для коррекции ошибок, внесенных при кодировании и декодировании данных 212 с пониженной дискретизацией. В некоторых примерах первое устройство 202 выводит данные коррекции, например, для передачи во второе устройство 204, а также закодированный сигнал. Это позволяет получателю исправить ошибки, внесенные при кодировании и декодировании данных 212 с пониженной дискретизацией. Указанные данные коррекции также могут упоминаться как "первый улучшенный" поток. Поскольку данные коррекции могут быть основаны на разности между данными 212 с пониженной дискретизацией и декодированным изображением, их можно рассматривать как форму остаточных данных (которые, например, отличаются от другого набора остаточных данных, описанного ниже).

В некоторых примерах генерация обработанных данных 213 с относительно низким уровнем качества дополнительно включает коррекцию декодированного изображения с использованием данных коррекции. Например, данным коррекции в качестве выходных данных для передачи может быть придана форма, подходящая для комбинирования с декодированным изображением, а затем они могут быть добавлены к декодированному изображению. Это может быть выполнено покадрово. В других примерах, вместо коррекции декодированного изображения с использованием данных коррекции первое устройство 202 использует данные 212 с пониженной дискретизацией. Например, в некоторых случаях могут быть использованы только закодированные, а затем декодированные данные, а в других случаях кодирование и декодирование могут быть заменены другой обработкой.

В некоторых примерах генерация обработанных данных 213 включает выполнение одной или более операций, отличных от операций кодирования, декодирования, получения и коррекции, описанных выше.

Первое устройство 202 получает данные 214, основанные на данных с относительно низким уровнем качества. Как указано выше, данные с относительно низким уровнем качества могут включать обработанные данные 213 или данные 212 с пониженной дискретизацией, причем данные 212 с пониженной дискретизацией не подлежат обработке с более низким уровнем. Как описано выше, в некоторых случаях обработанные данные 213 могут включать восстановленный видеопоток (например, на выходе операции кодирования-декодирования), который корректирует с использованием данных коррекции. В примере, показанном 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 представляет собой входные данные 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 выводит данные 214, выполняя операцию повышения дискретизации данных с относительно низким уровнем качества. Данные 214 в дальнейшем будут упоминаться как "данные с повышенной дискретизацией". Однако в других примерах для получения данных 214 могут использовать одну или более других операций, например, когда данные 212 не выводят с применением понижающей дискретизации входных данных 206. Следует отметить, что ссылки на относительно высокий и относительно низкий уровни качества могут соответствовать ссылкам на первый и второй уровни качества, причем второй уровень качества выше, чем первый уровень качества. Как описано в настоящем документе, в некоторых случаях уровни качества могут соответствовать разным пространственным разрешениям.

Входные данные 206 и данные 214 с повышенной дискретизацией используют для получения остаточных данных 216. Остаточные данные 216 связаны с изображением. Остаточные данные 216 могут иметь форму 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 заданный остаточный элемент получают путем вычитания значения элемента изображения в данных 214 с повышенной дискретизацией из значения соответствующего элемента изображения во входных данных 206. Таким образом, остаточные данные 216 могут быть использованы в сочетании с данными 214 с повышенной дискретизацией для восстановления входных данных 206. Остаточные данные 216 также могут упоминаться как "данные для восстановления" или "улучшенные данные". В одном случае остаточные данные 216 могут составлять часть "второго улучшенного" потока.

Первое устройство 202 получает данные конфигурации, относящиеся к обработке остаточных данных 216. Данные конфигурации указывают, как остаточные данные 216 были обработаны и/или сгенерированы 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 быть использованы для управления тем, как второе устройство 204 обрабатывает данные и/или восстанавливает входные данные 206 с использованием остаточных данных 216. Данные конфигурации могут относиться к одной или более характеристикам остаточных данных 216. Различные данные конфигурации могут обеспечивать различную обработку, выполняемую в отношении и/или с использованием остаточных данных 216. Таким образом, данные конфигурации могут быть использованы для восстановления входных данных 206 с использованием остаточных данных 216.

В этом примере первое устройство 202 передает на второе устройство 204 данные, основанные на данных 212 с пониженной дискретизацией, данные, основанные на остаточных данных 216, и данные конфигурации для обеспечения возможности восстановления вторым устройством 204 входных данных 206.

Как показано на фиг. 2В, второе устройство 204 принимает данные 220, основанные на (например, полученные из) данных 212 с пониженной дискретизацией. Второе устройство 204 также принимает данные, основанные на остаточных данных 216. Например, второе устройство 204 может принимать "основной" поток (данные 220), "первый улучшенный поток" (любые данные коррекции) и "второй улучшенный поток" (остаточные данные 216). Второе устройство 204 также принимает данные конфигурации, относящиеся к обработке остаточных данных 216. Данные 220, основанные на данных 212 с пониженной дискретизацией, могут представлять собой сами данные 212 с пониженной дискретизацией, обработанные данные 213 или данные, полученные из данных 212 с пониженной дискретизацией или обработанных данных 213. Данные, основанные на остаточных данных 216, могут представлять собой сами остаточные данные 216 или данные, полученные из 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, после обработки первым устройством 202 данных 212 с пониженной дискретизацией с генерацией обработанных данных 213 второе устройство 204 обрабатывает принятые данные 220 с генерацией обработанных данных 222. Такая обработка, выполняемая вторым устройством 204, может включать декодирование закодированного изображения (например, которое образует часть "основного" закодированного видеопотока) с получением декодированного изображения с относительно низким уровнем качества. В некоторых примерах обработка, выполняемая вторым устройством 204, включает коррекцию декодированного изображения с использованием 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 декодируют с помощью устройства декодирования, выполненного отдельно от второго устройства 204. Закодированное изображение с относительно низким уровнем качества может быть декодировано с использованием декодера H.264.

В других примерах принятые данные 220 содержат данные 212 с пониженной дискретизацией и не содержат обработанные данные 213. В некоторых таких примерах второе устройство 204 не обрабатывает принятые данные 220, генерируя обработанные данные 222.

Второе устройство 204 использует данные с относительно низким уровнем качества для получения данных 214 с повышенной дискретизацией. Как указано выше, данные с относительно низким уровнем качества могут включать обработанные данные 222 или принятые данные 220, причем второе устройство 204 не обрабатывает принятые данные 220 с относительно низким уровнем качества. Данные 214 с повышенной дискретизацией представляют собой предварительное представление изображения с относительно высоким уровнем качества. Данные 214 с повышенной дискретизацией могут быть получены путем выполнения операции повышения дискретизации данных с относительно низким уровнем качества.

Второе устройство 204 получает остаточные данные 216. Остаточные данные 216 могут быть использованы с данными 214 с повышенной дискретизацией для восстановления входных данных 206. Остаточные данные 216 представляют собой результат сравнения между входными данными 206 и данными 214 с повышенной дискретизацией.

Второе устройство 204 также получает данные конфигурации, относящиеся к обработке остаточных данных 216. Данные конфигурации могут быть использованы вторым устройством 204 для восстановления входных данных 206. Например, данные конфигурации могут указывать характеристику или свойство (относящееся к остаточным данным 216), которые указывают, как следует использовать и/или обрабатывать остаточные данные 216, или следует ли вообще использовать остаточные данные 216. В некоторых примерах данные конфигурации включают остаточные данные 216.

Приведем некоторые соображения, касающиеся использования таких данных конфигурации. Одно из таких соображений относится к объему генерируемой, хранимой, передаваемой и/или обрабатываемой информации. Чем больше информации используется, тем больше ресурсов может быть задействовано для обработки такой информации. Примеры таких ресурсов включают в себя ресурсы передачи, ресурсы

хранения и ресурсы обработки. По сравнению с некоторыми известными способами примеры, описанные в настоящем документе, позволяют использовать относительно небольшой объем информации. Это может позволить уменьшить объем данных, передаваемых по сети 106 передачи данных. Экономия ресурсов может быть, в частности, актуальной, если данные относятся к видеоданным высокого качества, когда объем информации, передаваемой в известных системах, может быть очень большим.

Еще одно соображение, касающееся использования таких данных конфигурации, относится к возможности адаптации к различным типам устройств декодирования. Например, данный кодер может передавать информацию на множество декодеров, каждый из которых имеет разные характеристики, такие как возможности по обработке и/или хранению. Желательно, чтобы использование данных конфигурации при восстановлении изображения не зависело от типа и/или возможностей устройства декодирования, используемых для выполнения восстановления. По сравнению с некоторыми известными способами, в примерах, описанных в настоящем документе, надежная обработка данных конфигурации может быть осуществлена с применением декодеров различных типов, имеющих разные рабочие характеристики.

Дополнительное соображение, касающееся использования таких данных конфигурации, относится к объему и/или относительной сложности обработки, осуществляемой в кодере и/или декодере при обработке данных конфигурации. По сравнению с некоторыми известными способами, в примерах, описанных в настоящем документе, уменьшен объем и/или сложность обработки, выполняемой кодером и/или декодером для получения и/или обработки данных конфигурации. Упрощение обработки, выполняемой кодером и/или декодером, обуславливает повышение эффективности кодера и/или декодера.

Другие соображения относятся к способности декодера точно и надежно выполнять восстановление изображения. Точное и надежное выполнение восстановления изображения может повлиять на конечное визуальное качество отображаемого изображения и, следовательно, может повлиять на восприятие зрителем изображения и/или видео, содержащего изображение. Одним из факторов, который может влиять на точность и/или надежность, с которыми декодер может выполнять восстановление, является предоставление декодеру актуальной и/или точной информации, относящейся к тому, как должно быть выполнено восстановление. Например, данные конфигурации могут изменяться со временем и/или могут быть различными для разных изображений. Если декодер не будет своевременно проинформирован о таких изменениях, это может повлиять на результирующее визуальное качество восстановленных изображений. По сравнению с некоторыми известными способами, примеры, описанные в настоящем документе, позволяют относительно часто передавать информацию о конфигурации на декодер, благодаря чему декодер использует относительно актуальную информацию о конфигурации при восстановлении изображения.

Одно из соображений относится к обработке ошибок. Например, способность декодера обрабатывать потенциальные ошибки при обработке данных конфигурации может представлять собой фактор надежности декодера. По сравнению с некоторыми известными способами, в примерах, описанных в настоящем документе, декодер выполнен с возможностью обработки потенциальных ошибок, а также получения незапланированных или нераспознанных данных. Это может позволить существенно повысить надежность декодера.

На фиг. 3 схематично показан пример способа 300 обработки данных. Способ 300 обработки данных может выполняться устройством, содержащим кодер, таким как первое устройство 102, описанное выше. Способ 300 обработки данных включает побайтовую обработку данных конфигурации, как более подробно описано ниже.

Получают данные 310 конфигурации. Данные 310 конфигурации относятся к обработке остаточных данных, например, остаточных данных 216, как описано выше. Следует понимать, что данные 310 конфигурации, изображенные на фиг. 3, являются иллюстративным представлением содержимого таких данных конфигурации. При практическом использовании данные конфигурации могут считываться и/или записываться в виде последовательности единиц и нулей, например, [01000010].

Данные 310 конфигурации содержат набор значений 320. Набор значений 320 соответствует набору параметров конфигурации. Значение заданного параметра конфигурации имеет заданную длину в битах. Длина в битах представляет собой количество битов, используемых для представления значения заданного параметра конфигурации. Длина в битах значения заданного параметра конфигурации может быть фиксированной или может быть переменной. Для переменной длины в битах количество битов, используемых для представления значения заданного параметра конфигурации, зависит от значения заданного параметра конфигурации.

В некоторых примерах данные 310 конфигурации содержат ссылку на справочную таблицу, в которой хранятся возможные значения заданного параметра конфигурации. Например, данные 310 конфигурации могут содержать значение, которое может быть использовано для получения значения параметра конфигурации из справочной таблицы.

Данные 310 конфигурации организованы в последовательность байтов 330. Последовательность байтов 330 содержит целое количество байтов. В этом примере значения параметров 320 конфигурации организованы в последовательность байтов, содержащую 5 байтов, хотя следует понимать, что в других примерах может быть использовано другое количество байтов. В этом примере заданный байт в после-

довательности байтов 330 содержит 8 битов. Последовательность байтов 330 может быть частью более крупной последовательности байтов или побайтового потока. Расположение данных 310 конфигурации в виде последовательности байтов 330 позволяет выполнять побайтовую обработку данных 310 конфигурации. При побайтовой обработке данные обрабатываются на побайтовой основе.

Обработка данных 310 конфигурации побайтовым способом может быть более эффективной, чем обработка данных 310 конфигурации побитовым способом. Побитовая обработка может включать отслеживание того, какой бит в данном байте будет следующим битом для чтения или записи. Если группа битов, которая представляет заданный параметр, охватывает границу байта, может выполняться дополнительная обработка для считывания группы битов и получения заданного параметра. Такая обработка может быть исключена в случае осуществления побайтовой обработки. В некоторых примерах количество циклов выполнения может быть уменьшено при побайтовой обработке данных по сравнению с побитовой обработкой.

В некоторых примерах последовательность байтов 330 записывается в запоминающее устройство. При сохранении данных 310 конфигурации в запоминающее устройство и/или при извлечении данных 310 конфигурации из запоминающего устройства может потребоваться меньший объем обработки, если данные 310 конфигурации обрабатываются побайтно по сравнению со случаем, когда данные 310 конфигурации обрабатываются побитно. Наименьшая единица адресуемого запоминающего устройства может содержать один байт. Таким образом, информация может храниться в запоминающем устройстве побайтно. Следовательно, если данные 310 конфигурации упакованы в последовательность байтов, может потребоваться меньше этапов для сохранения данных 310 конфигурации в запоминающем устройстве и/или извлечения данных 310 конфигурации из запоминающего устройства. Величина заполнения, используемая для упаковки данных 310 конфигурации в адресуемые блоки запоминающего устройства, также может быть уменьшена за счет обеспечения данных 310 конфигурации в запоминающем устройстве в побайтовой форме.

Кроме того, передача данных 310 конфигурации по сети, такой как Интернет, может выполняться более эффективно, если данные 310 конфигурации организованы в последовательность байтов. Данные могут передаваться по сети Интернет с использованием протокола управления передачей (Transmission Control Protocol, TCP).

TCP работает на транспортном уровне стека протоколов. TCP получает данные из потока, упаковывает данные в TCP-сегменты и использует Интернет-модуль, например, протокол Интернета (Internet Protocol, IP), для передачи каждого TCP-сегмента получателю по сети Интернет. Поток данных может быть получен из запоминающего устройства. TCP-сегмент может содержать целое количество байтов. Таким образом, может потребоваться меньший объем обработки, если TCP получает данные, подлежащие упаковке в TCP-сегменты, из побайтового потока, чем в случае их получения из побитового потока. Величина заполнения, используемая для упаковки данных в TCP-сегменты, также может быть уменьшена за счет обеспечения данных в TCP в побайтовой форме.

В этом примере размещение данных 310 конфигурации в виде последовательности байтов 330 выполняется на уровне приложений стека протоколов. На уровне приложений данные 310 конфигурации могут обрабатываться побайтно, а не побитно.

В некоторых примерах данные 310 конфигурации содержат значения множества параметров конфигурации, которые имеют разную длину в битах. Данные 310 конфигурации могут быть упакованы в последовательность байтов 330 путем расположения значений в заданном порядке. Заданный порядок может отличаться от порядка, в котором получены значения. Таким образом, значения могут быть переупорядочены.

Значения параметров 320 конфигурации могут быть организованы таким образом, чтобы заданный байт в последовательности байтов 330 заполнялся значениями с разными длинами в битах. За счет заполнения заданного байта в последовательности байтов 330 значениями с разными длинами в битах, расположенными в заданном порядке, величина заполнения, используемая в последовательности байтов 330, может быть уменьшена. Таким образом, данные могут быть упакованы в последовательность байтов 330 более эффективно по сравнению со случаем, когда значения с разными длинами в битах не расположены в заданном порядке. Более эффективная упаковка данных в последовательность байтов 330 может позволить уменьшить количество байтов, используемых для хранения данных 310 конфигурации. Следовательно, количество информации, которую необходимо передавать по сети, может быть уменьшено.

Последовательность байтов 330 выводится для обработки декодером, таким как декодер 110, для обеспечения возможности восстановления декодером входных данных 206 с использованием данных 310 конфигурации.

На фиг. 4 схематично показан пример данных 400 точки входа. Данные 400 точки входа могут выводиться кодером 108 и приниматься декодером 110. Данные 400 точки входа представляют собой один из типов данных конфигурации. В некоторых примерах данные 400 точки входа представляют собой часть данных 310 конфигурации. Данные 400 точки входа могут считываться декодером 110 до того, как будут считаны другие данные конфигурации. Таким образом, данные 400 точки входа могут быть первыми данными конфигурации, которые обрабатываются декодером 110 для данного принятого видео-



потока.

Данные 400 точки входа содержат параметр 420 версии. Параметр 420 версии может указывать номер версии. Версия может представлять собой версию синтаксиса, в соответствии с которым упорядочены данные конфигурации. Синтаксис можно рассматривать как структуру, которая определяет, как следует обрабатывать данные конфигурации, и/или определяет параметры конфигурации, которые должны быть включены в данные конфигурации. Кодер 108 и/или декодер 110 могут быть выполнены с возможностью функционирования в соответствии с версией синтаксиса, указанной параметром 420 версии.

Значение параметра 420 версии имеет фиксированную длину в битах. В некоторых примерах значение параметра 420 версии имеет длину в битах, равную 8 битов. Таким образом, значение параметра 420 версии может быть сохранено в элементе фиксированной длины в последовательности байтов 330. Элемент фиксированной длины содержит заданное целое количество байтов.

В этом примере данные 400 точки входа содержат параметр 425 расширенной версии. Значение параметра 425 расширенной версии имеет переменную длину в битах. Значение параметра 425 расширенной версии может быть сохранено в элементе переменной длины в последовательности байтов 330.

Элемент переменной длины содержит целое количество байтов в последовательности байтов 330. Элемент переменной длины может упоминаться как "многобайтовый" элемент, хотя в некоторых примерах элемент переменной длины содержит один байт. Один или более байтов элемента переменной длины могут быть смежными в последовательности байтов 330. В некоторых примерах один или более байтов элемента переменной длины не являются смежными в последовательности байтов 330.

По меньшей мере один заданный байт элемента переменной длины имеет один или более заданных битов, с помощью которых указывают, содержит ли элемент переменной длины один или более дополнительных байтов по меньшей мере к указанному одному заданному байту. В некоторых примерах один или более заданных битов содержат один заданный бит. В других примерах один или более заданных битов включают более одного заданного бита. В некоторых примерах каждый байт элемента переменной длины содержит один или более заданных битов. В других примерах по меньшей мере один байт элемента переменной длины не содержит один или более заданных битов.

В том случае, если один или более заданных битов включают один заданный бит, этот один заданный бит может считаться флагом, который указывает, присутствуют ли один или более дополнительных байтов в элементе переменной длины. Например, если элемент переменной длины содержит дополнительный байт по меньшей мере к одному заданному байту, значение заданного бита может быть установлено равным 1. Если, с другой стороны, элемент переменной длины не содержит дополнительный байт к по меньшей мере одному заданному байту, значение заданного бита может быть установлено равным 0. Другими словами, заданный бит указывает, выходит ли элемент переменной длины за рамки по меньшей мере одного заданного байта.

Один или более заданных битов могут быть расположены в конце по меньшей мере одного заданного байта. Например, один или более заданных битов могут включать один или более последних битов по меньшей мере одного заданного байта. В некоторых примерах один или более заданных битов включают один или более первых битов по меньшей мере одного заданного байта. В некоторых примерах каждый из по меньшей мере одного заданного байта содержит семь битов, предназначенных для хранения значения параметра 425 расширенной версии, и один бит, размещенный с возможностью указания того, содержит ли элемент переменной длины один или более дополнительных байтов по меньшей мере к указанному одному заданному байту.

Таким образом, значение параметра 425 расширенной версии может быть представлено с использованием элемента переменной длины независимо от длины в битах значения параметра 425 расширенной версии. Элемент переменной длины содержит минимальное количество байтов, которые могут быть использованы для хранения значения параметра 425 расширенной версии. Таким образом, при использовании элемента переменной длины может быть уменьшен объем хранимой, обрабатываемой и/или передаваемой информации по сравнению со случаем использования элемента фиксированной длины, размер которого соответствует максимально возможной длине в битах значения параметра 425 расширенной версии. Элемент переменной длины позволяет гибко и по мере необходимости увеличивать количество возможных значений, которые могут быть использованы для представления параметра 425 расширенной версии. Таким образом, элемент переменной длины может быть использован для обеспечения будущих разработок и/или изменений данных 310 конфигурации, например, позволяя увеличить длину в битах значения параметра 425 расширенной версии без необходимости осуществления относительно сложных изменений структуры данных 310 конфигурации.

В этом примере параметр 425 расширенной версии используется выборочно. Это показано пунктирными линиями на фиг. 4. Параметр 425 расширенной версии используют, если определено, что параметра 420 версии фиксированной длины недостаточно для хранения значения номера версии. В некоторых примерах значение номера версии сохраняется в параметре 425 расширенной версии в ответ на определение того, что значение номера версии имеет заданное соотношение с заданным пороговым значением. В некоторых примерах заданное соотношение включает значение номера версии, большее или равное заданному пороговому значению. В некоторых примерах, если определено, что значение номера

версии не имеет заданного соотношения с заданным пороговым значением, значение номера версии сохраняется в параметре 420 версии, а не в параметре 425 расширенной версии переменной длины. Выборочное использование элемента переменной длины обуславливает уменьшение объема используемой, хранимой и/или передаваемой информации по сравнению со случаем, когда элемент переменной длины используют независимо от значения номера версии.

В некоторых примерах заданное пороговое значение связано с длиной в битах элемента фиксированной длины, а именно с длиной в битах параметра 420 версии. Например, если элемент фиксированной длины содержит восемь битов, значения от 0 до 255 могут быть представлены с помощью элемента фиксированной длины. Заданное пороговое значение в таком случае может быть равно, например, 255. Таким образом, если значение номера версии больше 255, для сохранения значения номера версии параметр 425 расширенной версии переменной длины может быть использован вместо параметра 420 версии фиксированной длины или в дополнение к нему.

В некоторых примерах данные сохраняются в параметре 420 версии для указания того, что номер версии хранится в параметре 425 расширенной версии. В некоторых примерах данные содержат по меньшей мере один заданный бит в параметре 420 версии. Данные, хранящиеся в параметре 420 версии, позволяют декодеру 110 определить, что значение номера версии хранится в параметре 425 расширенной версии.

Вышеописанное использование элементов переменной длины и/или выборочное включение таких элементов переменной длины также может быть применимым в отношении других параметров конфигурации для данных конфигурации, выводимых кодером 108.

Дополнительно или в качестве альтернативы выборочному включению параметров конфигурации, соответствующих элементам переменной длины, кодер 108 может выборочно включать параметры конфигурации, соответствующие элементам фиксированной длины. В некоторых примерах данные конфигурации, генерируемые кодером 108, содержат индикатор наличия параметра. Индикатор наличия параметра указывает, содержится ли заданный набор параметров конфигурации в заданной части данных конфигурации. Заданный набор параметров конфигурации имеет заданный формат. В примере, относящемся к номеру версии, описанному выше, индикатор наличия параметра содержит заданный бит в параметре 420 версии, который указывает на включение параметра 425 расширенной версии. Индикаторы наличия параметра соответствуют вложенным логическим операторам, таким как операторы "если" (IF). Другими словами, если индикатор наличия параметра имеет конкретное значение или если ему задано конкретное значение, заданный набор параметров конфигурации включается в данные конфигурации. Выборочное включение заданного набора параметров конфигурации позволяет уменьшить количество используемой, хранимой и/или передаваемой информации по сравнению со случаем, когда заданный набор параметров конфигурации не включается выборочно.

На фиг. 5 схематично показан пример сообщения 500 с конфигурацией. Сообщение 500 с конфигурацией может представлять собой часть данных 310 конфигурации. Сообщение 500 с конфигурацией может быть сгенерировано кодером 108. Сообщение 500 с конфигурацией может быть принято и/или считано декодером 110 после того, как будут приняты и/или считаны данные 400 точки входа.

Сообщение 500 с конфигурацией содержит часть-заголовок 510 и часть-полезную нагрузку 520. Часть-полезная нагрузка 520 содержит набор параметров 550 конфигурации. Набор параметров 550 конфигурации относится к обработке остаточных данных, например остаточных данных 216, как описано выше. В некоторых примерах сообщение 500 с конфигурацией имеет длину, равную целому количеству байтов.

Часть-заголовок 510 содержит параметр 512 типа сообщения. Параметр 512 типа сообщения определяет тип сообщения 500 с конфигурацией. Тип сообщения, определенный параметром 500 типа сообщения, может указывать, как следует обрабатывать сообщение 500 с конфигурацией. Примеры типов сообщений с конфигурацией описаны ниже.

Параметр 512 типа сообщения указывает на то, что часть-полезная нагрузка 520 имеет заданный формат полезной нагрузки. Формат полезной нагрузки части-полезной нагрузки 520 может быть определен на основании типа сообщения 500 с конфигурацией. Формат полезной нагрузки может определять набор параметров конфигурации, которые содержатся в части-полезной нагрузке 520. В некоторых примерах формат полезной нагрузки части-полезной нагрузки 520 определяет конкретную логику обработки, которая определяет порядок обработки содержимого части-полезной нагрузки 520. Часть-полезная нагрузка 520 имеет размер формата. Размер формата части-полезной нагрузки 520 соответствует формату полезной нагрузки части-полезной нагрузки 520. Размер формата части-полезной нагрузки, имеющей заданный формат полезной нагрузки, задан или может быть определен из содержимого части-полезной нагрузки. Например, размер формата может быть определен на основании объединенного размера отдельных параметров конфигурации, которые определены форматом полезной нагрузки. Размер формата может рассматриваться как возможный или предполагаемый размер части-полезной нагрузки 520.

Часть-заголовок 510 также содержит параметр 514 размера полезной нагрузки. Параметр 514 размера полезной нагрузки указывает фактический размер части-полезной нагрузки 520. Фактический размер части-полезной нагрузки 520 может быть равен целому количеству байтов.

В некоторых примерах значение параметра 514 размера полезной нагрузки сохраняется в элементе фиксированной длины. Элемент фиксированной длины, хранящий значение параметра 514 размера полезной нагрузки, может содержать заданное количество битов в заданном байте части-заголовка 510. Например, значение параметра 514 размера полезной нагрузки может быть сохранено в элементе фиксированной длины, содержащем три бита в однобайтовой части-заголовке.

В некоторых примерах параметр 514 размера полезной нагрузки содержит ссылку на справочную таблицу с размерами полезной нагрузки, в которой хранятся возможные значения фактического размера части-полезной нагрузки 520. Например, параметр 514 размера полезной нагрузки может содержать значение, которое может быть использовано для определения фактического размера части-полезной нагрузки 520 из справочной таблицы с размерами полезной нагрузки.

В некоторых примерах значение параметра 514 размера полезной нагрузки сохраняется в элементе переменной длины. Это, например, может иметь место, если фактический размер части-полезной нагрузки 520 не соответствует ни одному из значений, указанных в справочной таблице с размерами полезной нагрузки. Фактический размер части-полезной нагрузки 520, определяемый параметром 514 размера полезной нагрузки, может иметь переменную длину в битах. Таким образом, параметр 514 размера полезной нагрузки может быть представлен с использованием элемента переменной длины независимо от длины в битах параметра 514 размера полезной нагрузки. Элемент переменной длины содержит минимальное количество байтов, которые могут быть использованы для хранения значения параметра 514 размера полезной нагрузки.

В некоторых примерах данные хранятся в элементе фиксированной длины, который указывает, что значение параметра 514 размера полезной нагрузки хранится в элементе переменной длины. Например, если определено, что значение параметра 514 размера полезной нагрузки соответствует одному значению из заданного набора значений, это значение может быть сохранено в элементе фиксированной длины и/или элемент фиксированной длины может содержать ссылку на справочную таблицу с размерами полезной нагрузки. С другой стороны, если определено, что значение параметра 514 размера полезной нагрузки не соответствует ни одному из значений из заданного набора значений, для хранения значения может быть использован элемент переменной длины, а элемент фиксированной длины может быть предназначен для указания, например, с помощью флага, содержащегося в элементе фиксированной длины, того, что для хранения значения используется элемент переменной длины.

В этом примере часть-заголовок 510 содержит только параметр 512 типа сообщения и параметр 514 размера полезной нагрузки. В других примерах часть-заголовок 510 содержит один или более дополнительных параметров. В некоторых примерах часть-заголовок 510 имеет длину, равную одному байту. Например, параметр 512 типа сообщения может быть представлен пятью битами, а параметр 514 размера полезной нагрузки может быть представлен оставшимися тремя битами однобайтовой части-заголовка 510. В других примерах, например, если элемент переменной длины используют для представления фактического размера части-полезной нагрузки 520, часть-заголовок 510 содержит более одного байта.

Фактический размер части-полезной нагрузки 520 может быть таким же, как размер формата части-полезной нагрузки 520, или может отличаться от него. Независимо от того, совпадают ли фактический размер и размер формата, параметр 514 размера полезной нагрузки, определяющий фактический размер, включается в часть-заголовок 510 сообщения 500 с конфигурацией.

Включение параметра 514 размера полезной нагрузки, который определяет фактический размер части-полезной нагрузки 520, когда возможный размер части-полезной нагрузки 520, а именно размер формата, может быть определен на основании параметра 512 типа сообщения, может выглядеть как неэффективное действие и/или дублирование информации, в частности, если размер формата и фактический размер совпадают. Включение параметра 514 размера полезной нагрузки в сообщение 500 с конфигурацией предполагает использование большего количества данных для сообщения 500 с конфигурацией по сравнению со случаем, в котором включение такого параметра не происходит. Однако включение параметра 514 размера полезной нагрузки в сообщение 500 с конфигурацией обеспечивает более эффективную и надежную обработку в декодере 110 и/или улучшенную обработку ошибок по сравнению со случаем, когда такой параметр не используется, как будет более подробно описано ниже.

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

В некоторых примерах осуществляется прием остаточных данных. В этом примере принятые остаточные данные представляют собой остаточные данные 216, описанные выше. Остаточные данные 216 могут быть приняты от кодера, такого как кодер 108.

На этапе 610 принимается сообщение с конфигурацией. В этом примере полученное сообщение с конфигурацией представляет собой сообщение 500 с конфигурацией, описанное выше. Сообщение 500 с конфигурацией может быть принято от кодера 108.

На этапе 620 принятое сообщение 500 с конфигурацией обрабатывается с использованием параметра 512 типа сообщения и параметра 514 размера полезной нагрузки. Сообщение 500 с конфигурацией обрабатывается с получением набора параметров 550 конфигурации, сохраняемого в части-полезной нагрузки 520 сообщения 500 с конфигурацией.

В некоторых примерах обработка сообщения 500 с конфигурацией включает синтаксический анализ части-заголовка 510 сообщения 500 с конфигурацией с получением параметра 512 типа сообщения и параметра 512 размера полезной нагрузки. На основании параметра 514 размера полезной нагрузки определяется фактический размер части-полезной нагрузки 520. На основании параметра 512 типа сообщения определяется тип сообщения 500 с конфигурацией. Сообщение 500 с конфигурацией обрабатывается в соответствии с определенным типом сообщения и определенным фактическим размером части-полезной нагрузки 520.

Обработка сообщения 500 с конфигурацией включает синтаксический анализ части-полезной нагрузки 520 с получением данных 550 конфигурации. В некоторых примерах обработка сообщения 500 с конфигурацией включает отслеживание количества байтов сообщения 500 с конфигурацией, данные о котором были считаны при синтаксическом анализе части-полезной нагрузки 520. Декодер 110 может обнаруживать неполностью считанные значения при определении несоответствия между количеством байтов, считанных при синтаксическом анализе части-полезной нагрузки 520, и фактическим размером части-полезной нагрузки 520, указанной параметром 514 размера полезной нагрузки. Неполностью считанные значения могут быть потенциальным источником ошибок при обработке сообщений с конфигурацией.

В некоторых примерах параметр 514 размера полезной нагрузки используется для определения начала следующего сообщения с конфигурацией. Следующее сообщение с конфигурацией может представлять собой следующее последовательное сообщение с конфигурацией относительно сообщения 500 с конфигурацией в последовательности сообщений с конфигурацией, принятых декодером 110. Последовательность сообщений с конфигурацией может представлять собой часть данных конфигурации. Сообщение 500 с конфигурацией и следующее сообщение с конфигурацией могут соответствовать одному и тому же изображению или разным изображениям. Например, сообщение 500 с конфигурацией может соответствовать первому изображению в последовательности изображений, а следующее сообщение с конфигурацией может соответствовать второму, другому изображению в последовательности изображений.

В некоторых примерах обработка сообщения 500 с конфигурацией включает отбрасывание по меньшей мере части содержимого части-полезной нагрузки 520. По меньшей мере часть содержимого части-полезной нагрузки 520 может быть отброшена с использованием параметра 514 размера полезной нагрузки. Например, декодер 110 может перейти к следующему сообщению с конфигурацией, причем начало следующего сообщения с конфигурацией было определено с использованием параметра 514 размера полезной нагрузки без обработки по меньшей мере части содержимого части-полезной нагрузки 520.

Таким образом, использование параметра 514 размера полезной нагрузки позволяет декодеру 110 эффективно обрабатывать возможные неполностью считанные значения. Использование параметра 514 размера полезной нагрузки для определения начала следующего сообщения также позволяет декодеру 110 обеспечить правильное согласование или синхронизацию с потоком байтов, содержащим множество сообщений с конфигурацией. Например, даже если содержимое и/или формат данного сообщения с конфигурацией не распознаны, являются незапланированными и/или были обновлены без уведомления декодера 110, декодер 110 все же может определить начало следующего сообщения с конфигурацией и, таким образом, обеспечивает синхронизацию с потоком байтов. Кроме того, если поток байтов нарушается, декодер 110 может быть выполнен с возможностью определения того, когда должно произойти перечитывание, и реагирования соответствующим образом. Например, декодер 110 может прервать обработку текущего сообщения с конфигурацией и сбросить поток байтов к определенному местоположению начала следующего сообщения с конфигурацией. Таким образом, обработка сообщения 500 с конфигурацией с использованием параметра 514 размера полезной нагрузки повышает надежность и/или адаптируемость декодера 110 по сравнению со случаем, когда такой параметр не используется, поскольку декодер 110 выполнен с возможностью обработки сообщений с конфигурацией, которые были изменены, и/или не были распознаны и/или оказались незапланированными. Декодер 110 также выполнен с возможностью обработки потенциальных неполностью считанных или перечитанных значений без потери синхронизации с потоком байтов, в результате чего улучшается способность декодера 110 к обработке потенциальных ошибок.

В некоторых примерах обработка сообщения 500 с конфигурацией включает синтаксический анализ первого содержимого части-полезной нагрузки 520. Первое содержимое соответствует размеру формата части-полезной нагрузки 520. Первое содержимое может содержать набор параметров 550 конфигурации. Первое содержимое может содержать первые данные конфигурации. Обработка сообщения 500 с конфигурацией может дополнительно включать отбрасывание оставшегося содержимого части-полезной нагрузки 520 с использованием параметра 514 размера полезной нагрузки. Оставшееся содержимое части-полезной нагрузки 520 включает содержимое части-полезной нагрузки 520, которое не содержится в первом содержимом. Оставшееся содержимое содержит дополнительные данные конфигурации. Наличие оставшегося содержимого может быть определено путем обнаружения несоответствия между размером формата и фактическим размером части-полезной нагрузки 520. Другими словами, если определено,

что фактический размер превышает размер формата, определяется, что в части-полезной нагрузке 520 помимо содержимого, которое ожидается на основании размера формата, присутствует дополнительное содержимое.

На этапе 630 входные данные 206 восстанавливаются с использованием полученного набора параметров 550 конфигурации. Входные данные 206 могут быть восстановлены с использованием остаточных данных 216 в соответствии с полученным набором параметров 550 конфигурации.

В некоторых примерах часть-полезная нагрузка 520 сообщения 500 с конфигурацией содержит первую и вторую порции полезной нагрузки. Параметр 514 размера полезной нагрузки может указывать общий размер первой и второй порций полезной нагрузки. Первая и вторая порции полезной нагрузки могут содержать, соответственно, первые и вторые данные конфигурации. В некоторых примерах вторые данные конфигурации включают дополнительные данные конфигурации. Дополнительные данные конфигурации предназначены для дополнения первых данных конфигурации. Соответственно, вторая порция полезной нагрузки может быть размещена после первой порции полезной нагрузки. Таким образом, первая порция полезной нагрузки может быть принята, обработана и/или синтаксически проанализирована декодером 110 до того, как будет принята, обработана и/или синтаксически проанализирована вторая порция полезной нагрузки.

В некоторых примерах в части-заголовке 510 не определен размер первой порции полезной нагрузки и/или размер второй порции полезной нагрузки. В некоторых примерах размер первой порции полезной нагрузки может быть определен на основании размера формата, указанного с помощью параметра 512 типа сообщения. Таким образом, размер формата, указанный с помощью параметра 512 типа сообщения, может отражать содержимое первой порции полезной нагрузки, но не содержимое второй порции полезной нагрузки.

Первые данные конфигурации относятся к первому набору из одного или более декодеров. Первый набор из одного или более декодеров имеет первую рабочую характеристику. Примером рабочей характеристики являются возможности по обработке. Вторые данные конфигурации относятся ко второму набору из одного или более декодеров. Второй набор из одного или более декодеров имеет вторую рабочую характеристику. Декодер 110, на который передается сообщение 500 с конфигурацией, может войти в первый набор или второй набор. Кодер 108, который генерирует сообщение 500 с конфигурацией, может не иметь данных о том, в каком наборе декодеров содержится декодер 110. Кодер 108 может передать сообщение 500 с конфигурацией на множество декодеров в обоих наборах декодеров.

В некоторых примерах первая рабочая характеристика соответствует первой версии, согласно которой для работы сконфигурирован первый набор из одного или более декодеров, а вторая рабочая характеристика соответствует второй, другой версии, согласно которой для работы сконфигурирован второй набор из одного или более декодеров. Вторая версия может быть более поздней версией 110 сравнению с первой версией. Первая и вторая версии могут представлять собой разные версии структуры синтаксиса, определяющей, как должны обрабатываться сообщения с конфигурацией, и/или определяющей содержимое таких сообщений с конфигурацией. Таким образом, при обновлении синтаксиса с первой версии до второй версии вторые данные конфигурации добавляются в сообщение с конфигурацией, которое содержит первые данные конфигурации из первой версии. Вторые данные конфигурации могут, например, быть добавлены к первым данным конфигурации.

Добавление вторых данных конфигурации в сообщение с конфигурацией существующего типа вместо создания нового типа сообщения для размещения вторых данных конфигурации позволяет уменьшить объем хранимых, передаваемых и/или обрабатываемых данных по сравнению со случаем генерирования новых типов сообщений. Изменение конфигурации существующих типов сообщений вместо определения новых типов сообщений позволяет уменьшить общее количество сообщений с конфигурацией, используемых для передачи всех соответствующих данных конфигурации. Каждое сообщение с конфигурацией имеет соответствующий заголовок, длина которого может составлять по меньшей мере один байт. Таким образом, за счет уменьшения общего количества сообщений с конфигурацией может быть уменьшен объем хранимых, передаваемых и/или обрабатываемых данных.

В некоторых примерах декодер 110 не выполнен с возможностью обработки как первой, так и второй порции полезной нагрузки. Например, может быть выполнено обновление кодера 108, согласно которому вторые данные конфигурации будут включены в сообщение 500 с конфигурацией, но на декодере 110 может не проводиться такое обновление. Другими словами, кодер 108 и декодер 110 могут работать в соответствии с разными версиями синтаксиса. В таких примерах может возникнуть несоответствие в интерпретации размера формата части-полезной нагрузки 520 кодером 108 и декодером 110. Например, размер формата, интерпретируемый кодером 108, с учетом дополнительного содержимого части-полезной нагрузки 520 и обновленного формата полезной нагрузки после обновления версии отличается от размера формата, интерпретируемого декодером 110, поскольку декодер 110 не имеет данных о дополнительном содержимом и обновленном формате полезной нагрузки, и функционирует в соответствии с предыдущей версией. Для кодера 108 размер формата равен фактическому размеру обновленной части-полезной нагрузки 520, но для декодера 110 это не так. Таким образом, использование только размера формата может быть ненадежным индикатором фактического размера части-полезной нагрузки 520. Та-

ким образом, включение параметра 514 размера полезной нагрузки, который определяет фактический размер части-полезной нагрузки 520, обеспечивает более надежное выполнение обработки декодером 110.

В некоторых примерах, если декодер 110, например, состоит в первом наборе декодеров, он выполняет синтаксический анализ первой порции полезной нагрузки и не выполняет синтаксический анализ второй порции полезной нагрузки. При выполнении синтаксического анализа первой порции полезной нагрузки и невыполнении синтаксического анализа второй порции полезной нагрузки декодер 110 получает первые данные конфигурации, но при этом не будут получены вторые данные конфигурации. В некоторых примерах, если декодер 110, например, состоит во втором наборе декодеров, он выполняет синтаксический анализ второй порции полезной нагрузки и не выполняет синтаксический анализ первой порции полезной нагрузки. При выполнении синтаксического анализа второй порции полезной нагрузки и невыполнении синтаксического анализа первой порции 522 полезной нагрузки декодер 110 получает вторые данные конфигурации, но при этом не будут получены первые данные конфигурации. В некоторых примерах, если декодер 110, например, состоит во втором наборе декодеров, он выполняет синтаксический анализ как первой, так и второй порций полезной нагрузки с получением как первых, так и вторых данных конфигурации.

Как описано выше, параметр 514 размера полезной нагрузки может быть использован для определения начала следующего сообщения с конфигурацией. В некоторых примерах, если декодер 110, например, состоит в первом наборе декодеров, содержимое второй порции полезной нагрузки отбрасывается. Содержимое второй порции полезной нагрузки может быть отброшено на основании данных об общем размере части-полезной нагрузки 520, указанном в параметре 514 размера полезной нагрузки. Использование параметра 514 размера полезной нагрузки для определения начала следующего сообщения позволяет декодеру 110 поддерживать синхронизацию с потоком байтов независимо от того, состоит ли декодер 110 в первом наборе декодеров или во втором наборе декодеров.

В том случае, если декодер 110 получает как первые, так и вторые данные конфигурации, восстановление первого представления 206 изображения происходит с использованием как первых, так и вторых данных конфигурации. В том случае, если декодер 110 обрабатывает только одну из первой и второй порций полезной нагрузки, первое представление 206 изображения может быть восстановлено с использованием полученных одних из первых и вторых данных конфигурации. Таким образом, использование сообщения 500 с конфигурацией для восстановления первого представления 206 может не зависеть от типа декодера и/или номера версии, в соответствии с которой сконфигурирован для работы декодер 110.

На фиг. 7 представлена упрощенная схема примера сообщения 700 с конфигурацией. Сообщение 700 с конфигурацией может представлять собой часть данных 310 конфигурации. Сообщение 700 с конфигурацией может быть сгенерировано кодером 108. Некоторые элементы, изображенные на фиг. 7, аналогичны элементам, показанным на фиг. 5. Числовые значения соответствующих ссылочных позиций увеличены на 200 и, таким образом, используются для аналогичных элементов.

Сообщение 700 с конфигурацией является более конкретным примером сообщения 500 с конфигурацией. Сообщение 700 с конфигурацией содержит часть-заголовок 710 и часть-полезную нагрузку 720. Часть-заголовок 710 содержит параметр 712 типа сообщения и параметр 714 размера полезной нагрузки. Часть-полезная нагрузка 720 содержит набор параметров конфигурации. Набор параметров конфигурации относится к обработке остаточных данных 216. В некоторых примерах сообщение 700 с конфигурацией имеет длину, равную целому количеству байтов.

Сообщение 700 с конфигурацией относится к типу сообщений с конфигурацией набора изображений. Сообщение с конфигурацией набора изображений соответствует множеству изображений в наборе изображений. В некоторых примерах набор изображений включает 75 изображений. Набор изображений может представлять собой часть видеосигнала. Например, набор изображений может содержать последовательные кадры видео. Видео может быть потоковым видео. Примером видео является трансляция видео. Например, видео может транслироваться по сети 106 передачи данных. В некоторых примерах набор изображений кодируется как группа изображений (group of pictures, GOP). Группа изображений представляет собой закодированную последовательность изображений, каждое из которых может быть декодировано с использованием информации, полностью содержащейся в группе изображений. Видео может содержать множество таких групп изображений.

В некоторых примерах сообщение с конфигурацией набора изображений соответствует каждому изображению в наборе изображений. Часть-полезная нагрузка 720 содержит набор параметров конфигурации набора изображений. Набор параметров конфигурации набора изображений имеет заданный формат.

В этом примере набор параметров конфигурации набора изображений содержит параметр 751 типа преобразования. Параметр 751 типа преобразования относится к операции преобразования, выполняемой в отношении остаточных данных 216. Операция преобразования может представлять собой операцию направленной декомпозиции. Одним из примеров операции направленной декомпозиции является преобразование Адамара (Hadamard). В операции преобразования может быть использована пространственная корреляция и/или корреляция по направлениям между остаточными элементами в остаточных дан-

ных 216. Выполнение операции преобразования остаточных данных 216 приводит к преобразованию остаточных данных. Преобразованные остаточные данные могут иметь меньший объем, чем остаточные данные 216. Преобразованные остаточные данные могут иметь более низкую энтропию кодирования, чем остаточные данные 216. Таким образом, для кодера 108 может быть более эффективным передавать преобразованные остаточные данные, чем остаточные данные 216. Декодер 110, принимая преобразованные остаточные данные, может затем выполнить операцию обратного преобразования преобразованных остаточных данных, чтобы получить остаточные данные 216. Таким образом, кодер 108 сообщает декодеру 110, какая операция преобразования была применена к остаточным данным 216, с помощью параметра 751 типа преобразования. В некоторых примерах параметр 751 типа преобразования определяет операцию преобразования из множества возможных операций преобразования, которые могут быть применены к остаточным данным 216. В некоторых примерах значение параметра 751 типа преобразования имеет фиксированную длину в битах. Например, значение параметра 751 типа преобразования может иметь длину, равную двум битам.

В этом примере набор параметров конфигурации набора изображений содержит параметр 752 типа повышения дискретизации. Параметр 752 типа повышения дискретизации относится к операции повышения дискретизации. Операцию повышения дискретизации выполняют для каждого изображения в наборе изображений по представлению изображения с относительно низким уровнем качества. Операцию повышения дискретизации выполняют по представлению с относительно низким уровнем качества с получением данных 214 с повышенной дискретизацией. Кодер 108 выполняет такую операцию повышения дискретизации с получением остаточных данных 216. Декодер 110 может выполнить соответствующую операцию повышения дискретизации, чтобы восстановить входные данные 206, а именно объединить данные 214 с повышенной дискретизацией с остаточными данными 216. Таким образом, кодер 108 сообщает декодеру 110, какую операцию повышения дискретизации следует использовать, с помощью параметра 752 типа повышения дискретизации. В некоторых примерах параметр 752 типа повышения дискретизации определяет операцию интерполяции из множества возможных операций интерполяции. Примеры возможных операций интерполяции включают интерполяцию ближайшего соседа, билинейную интерполяцию, бикубическую интерполяцию и интерполяцию Ланцоша (Lanczos). В некоторых примерах параметр типа повышения дискретизации указывает, выполняется ли повышающая дискретизация в одном измерении или в двух измерениях. Другими словами, параметр 752 типа повышения дискретизации может указывать, выполняется ли одно или оба из горизонтального и вертикального повышения дискретизации по представлению с относительно низким уровнем качества. В некоторых примерах параметр конфигурации, относящийся к операции повышения дискретизации, может указывать, выполняется ли горизонтальное или как горизонтальное, так и вертикальное повышение дискретизации по представлению с относительно низким уровнем качества. В некоторых примерах значение параметра 752 типа повышения дискретизации имеет фиксированную длину в битах. В одном примере значение параметра 752 типа повышения дискретизации имеет длину, равную трем битам. Еще в одном примере значение параметра 752 типа повышения дискретизации имеет длину, равную четырем битам.

В этом примере набор параметров конфигурации набора изображений содержит параметр 753 типа глубины. Параметр 753 типа глубины относится к глубине в битах. Глубина в битах связана с остаточными данными 216 для заданного изображения в наборе изображений. Глубина в битах также может упоминаться как глубина цвета. В некоторых примерах глубина в битах указывает количество битов, используемых для обозначения цвета заданного пикселя в заданном изображении. Таким образом, глубина в битах может указывать на то, насколько точно цвет может быть представлен в заданном изображении. В некоторых примерах глубина в битах указывает количество битов, используемых для каждого компонента цвета заданного пикселя в заданном изображении. В некоторых примерах заданный остаточный элемент в остаточных данных 216 для заданного изображения содержит множество компонентов цвета (например, для данных YUV или RGB). Глубина в битах может указывать количество битов, используемых для каждого компонента цвета заданного остаточного элемента. Разные компоненты цветов могут иметь разную длину в битах. В некоторых примерах значение параметра 753 типа глубины имеет фиксированную длину в битах. В одном примере значение параметра 753 типа глубины имеет длину, равную двум битам. Еще в одном примере значение параметра 753 типа глубины имеет длину, равную четырем битам.

В этом примере набор параметров конфигурации набора изображений содержит параметр 754 временной обработки. Параметр 754 временной обработки определяет степень временной обработки, связанной с восстановлением (для заданного изображения в наборе изображений) входных данных 206. Выполнение временной обработки включает использование данных, основанных на множестве изображений в наборе изображений. В некоторых примерах каждое из изображений в наборе изображений соответствует разной временной выборке видео. Например, с помощью временной обработки можно скорректировать данные для заданного кадра видео, например, данные, полученные из остаточных данных для кадра, с использованием данных для другого кадра видео, таких как данные, полученные из остаточных данных для другого кадра. Данные, основанные на множестве изображений, могут указывать степень временной корреляции между остаточными данными для заданного изображения и остаточными

данными для одного или более дополнительных изображений в наборе изображений. Одно или более дополнительных изображений могут соответствовать одной или более предыдущим временным выборкам в видео относительно временной выборки, соответствующей заданному изображению. Например, данные, по меньшей мере полученные из остаточных данных, которые связаны с предыдущим кадром видео, могут быть использованы для восстановления текущего кадра видео путем изменения данных, по меньшей мере полученных из остаточных данных, которые связаны с текущим кадром.

В некоторых примерах данные, основанные на множестве изображений, содержат данные временной корреляции. Данные временной корреляции могут включать данные, основанные на разнице между остаточными данными для заданного изображения и соответствующими остаточными данными для одного или более дополнительных изображений. Это различие может быть основано на данных для текущего кадра и данных для предыдущего кадра. Для последовательностей изображений с относительно высокой степенью временной корреляции может быть более эффективным использовать и/или выводить данные временной корреляции, а не остаточные данные для данного изображения.

В некоторых примерах значение параметра 754 временной обработки основано на показателе оптимизации. Операция оптимизации может выполняться для определения того, являются ли остаточные данные или данные временной корреляции более эффективными для использования и/или вывода для данного изображения. На основании показателя оптимизации с помощью значения параметра 754 временной обработки декодеру 110 может быть указано восстановить заданный кадр видео на основании одного из остаточных данных и данных временной корреляции. Эта операция может отражать обработку, которая была выполнена на соответствующем кодере. Операция оптимизации может включать анализ искажения в зависимости от скорости передачи, более подробно описанный ниже.

Значение параметра 754 временной обработки может иметь длину, равную одному биту. Например, оно может указывать, следует ли выполнять временную обработку (двоичное определение), например, в отношении заданного кадра видео. Таким образом, параметр 754 временной обработки может быть сохранен в заданном бите в заданном байте данных конфигурации.

В этом примере набор параметров конфигурации набора изображений содержит параметр 755 типа разрешения, определяющий разрешение входных данных 206 для данного изображения в наборе изображений. В некоторых примерах значение параметра 755 типа разрешения имеет фиксированную длину в битах. В одном примере значение параметра 755 типа разрешения имеет длину, равную семи битам.

В некоторых примерах набор параметров конфигурации набора изображений содержит параметр 756 настраиваемого разрешения. Параметр настраиваемого разрешения относится к разрешению входных данных 206. Параметр 756 настраиваемого разрешения может выборочно включаться в сообщение 700 с конфигурацией набора изображений. Это показано пунктирными линиями на фиг. 7. Например, параметр 756 настраиваемого разрешения может быть включен, если определено, что разрешение входных данных 206 не входит в заданный набор значений, которые могут быть указаны с помощью параметра 755 типа разрешения. Данные могут быть сохранены в параметре 755 типа разрешения для указания того, что параметр 756 настраиваемого разрешения включен в сообщение 700 с конфигурацией набора изображений. Такие данные могут содержать заданный бит в параметре 755 типа разрешения. Таким образом, заданный бит содержит индикатор наличия параметра, который указывает присутствие параметра 756 настраиваемого разрешения в сообщении 700 с конфигурацией набора изображений.

Набор параметров конфигурации набора изображений может содержать больше, меньше параметров или параметры, отличные от описанных выше со ссылкой на фиг. 7.

Сообщение 700 с конфигурацией набора изображений соответствует множеству изображений в наборе изображений. В этом примере сообщение 700 с конфигурацией набора изображений соответствует всем изображениям в наборе изображений. Таким образом, один набор параметров конфигурации набора изображений применим к каждому изображению в наборе изображений. Данные конфигурации набора изображений также могут упоминаться как "данные глобальной конфигурации".

Сообщение 700 с конфигурацией набора изображений выводится на декодер 110, чтобы декодер 110 восстановил (для каждого изображения в наборе изображений) входные данные 206 с использованием данных 214 с повышенной дискретизацией и остаточных данных 216.

Количество раз, которое необходимо вывести сообщение 700 с конфигурацией набора изображений для набора изображений, меньше количества изображений в наборе изображений. Благодаря тому, что сообщение 700 с конфигурацией набора изображений выводят для количества изображений, которое меньше количества изображений в наборе изображений, уменьшается количество информации, передаваемой в течение заданного периода времени. В этом примере количество раз, когда сообщение 700 с конфигурацией набора изображений выводят для набора изображений, равно единице. Сообщение 700 с конфигурацией набора изображений может быть выведено только для первого изображения в наборе изображений. Первое изображение в наборе изображений может быть первым выводимым, принимаемым, обрабатываемым и/или отображаемым изображением в наборе изображений. Вывод сообщения 700 с конфигурацией набора изображений для первого изображения в наборе изображений может позволить получателю как можно раньше получить данные конфигурации набора изображений и/или может позволить получателю по возможности применить данные конфигурации набора изображений в отношении



как можно большего количества изображений в наборе изображений. В некоторых примерах сообщение 700 с конфигурацией набора изображений выводится вместе или параллельно с первым изображением в наборе изображений. В некоторых примерах сообщение 700 с конфигурацией набора изображений выводится для изображения, отличного от первого изображения в наборе изображений. В некоторых примерах количество раз, когда сообщение 700 с конфигурацией набора изображений выводится для набора изображений, больше одного, но меньше количества изображений в наборе изображений. В одном примере сообщение 700 с конфигурацией набора изображений выводится один раз для набора изображений, содержащего 75 изображений.

В некоторых примерах количество раз, когда сообщение 700 с конфигурацией набора изображений выводится для набора изображений, зависит от показателя оптимизации. Примером показателя оптимизации является показатель искажения в зависимости от скорости передачи. Оптимизация искажения в зависимости от скорости передачи обеспечивает оптимизацию объема передаваемых данных относительно величины потери визуального качества. Чем реже выводится сообщение 700 с конфигурацией набора изображений, тем меньше передается данных. Однако слишком частый вывод сообщения 700 с конфигурацией набора изображений может привести к нежелательной потере качества вследствие того, что декодеру 110 не будут предоставлены обновленные данные конфигурации. Таким образом, количество раз, когда выводится сообщение 700 с конфигурацией набора изображений, оптимизируют таким образом, чтобы уменьшить объем отправляемых данных и одновременно уменьшить любые потери в визуальном качестве.

В некоторых примерах отношение количества раз, когда выводится сообщение 700 с конфигурацией набора изображений для набора изображений, к количеству изображений в наборе изображений меньше заданного порогового значения. Заданное пороговое значение может быть основано на показателе оптимизации, таком как показатель искажения в зависимости от скорости передачи, описанный выше.

На фиг. 8 схематично показан пример сообщения 800 с конфигурацией. Сообщение 800 с конфигурацией может представлять собой часть данных 310 конфигурации. Сообщение 800 с конфигурацией может быть сгенерировано кодером 108. Некоторые элементы, изображенные на фиг. 8, аналогичны элементам, показанным на фиг. 5. Числовые значения соответствующих ссылочных позиций увеличены на 300 и, таким образом, используются для аналогичных элементов.

Сообщение 800 с конфигурацией является более конкретным примером сообщения 500 с конфигурацией. Сообщение 800 с конфигурацией содержит часть-заголовок 810 и часть-полезную нагрузку 820. Часть-заголовок 810 содержит параметр 812 типа сообщения и параметр 814 размера полезной нагрузки. Часть-полезная нагрузка 820 содержит набор параметров конфигурации. Набор параметров конфигурации относится к обработке остаточных данных 216. В некоторых примерах сообщение 800 с конфигурацией имеет длину, равную целому количеству байтов.

Сообщение 800 с конфигурацией относится к типу сообщений с конфигурацией изображения. Сообщение с конфигурацией изображения может соответствовать заданному изображению только из некоторого набора изображений. Таким образом, для каждого изображения в наборе изображений может быть сгенерировано и/или выведено другое сообщение с конфигурацией изображения. Часть-полезная нагрузка 820 содержит набор параметров конфигурации изображения.

В этом примере набор параметров конфигурации изображения содержит параметр 851 использования остаточных данных. Значение параметра 851 использования остаточных данных указывает, следует ли восстанавливать входные данные 206 для заданного изображения с использованием остаточных данных 216 для заданного изображения. Если входные данные 206 должны быть восстановлены, в сообщение 800 с конфигурацией изображения включают дополнительный набор параметров конфигурации. Если входные данные 206 не должны быть восстановлены, в сообщение 800 с конфигурацией изображения не включают дополнительный набор параметров конфигурации. Таким образом, дополнительный набор параметров конфигурации включают выборочно. Это изображено пунктирными линиями на фиг. 8.

В этом примере значение параметра 851 использования остаточных данных указывает, включен ли дополнительный набор параметров конфигурации в сообщение 800 с конфигурацией изображения. В этом примере значение параметра 851 использования остаточных данных может, таким образом, указывать, включен ли дополнительный набор параметров конфигурации в сообщение 800 с конфигурацией изображения. Таким образом, параметр 851 использования остаточных данных может содержать индикатор наличия параметра. Параметр 851 использования остаточных данных может быть использован декодером 110 для определения того, следует ли восстанавливать входные данные 206 с использованием остаточных данных 216, и того, включен ли дополнительный набор параметров конфигурации в сообщение 800 с конфигурацией.

В некоторых примерах, если дополнительный набор параметров конфигурации включен в сообщение 800 с конфигурацией изображения, часть-полезная нагрузка 820 сообщения 800 с конфигурацией имеет длину, равную 5 байтам. Если дополнительный набор параметров конфигурации не включен в сообщение 800 с конфигурацией, часть-полезная нагрузка 820 сообщения 800 с конфигурацией может иметь длину, равную 1 байту.

В этом примере набор параметров конфигурации изображения содержит данные 852 квантования.

Данные 852 квантования содержатся в дополнительном наборе параметров конфигурации, включение которых указано параметром 851 использования остаточных данных. Данные 852 квантования указывают уровень квантования данных, которые могут быть использованы декодером 110 для восстановления (для заданного изображения) входных данных 206 с использованием данных 214 с повышенной дискретизацией.

Квантование включает аппроксимацию относительно большого набора значений с применением относительно небольшого набора значений, например, путем округления, определения порога или усечения значений в относительно большом наборе. Задачей операции квантования является уменьшение количества данных в квантованных выходных данных по сравнению с количеством данных во входных данных для операции квантования. Уровень квантования заданных данных может указывать на то, насколько эффективно квантованы данные. Уровень квантования может указывать на степень детализации квантованных данных, полученных в процессе квантования. Другими словами, уровень квантования может указывать на то, какое количество информации теряется в ходе такого квантования. Например, уровень квантования может указывать одно или более пороговых значений или интервалов квантования, которые указывают, как будут квантованы данные.

В некоторых примерах данные 852 квантования указывают уровень квантования остаточных данных 216. Таким образом, данные, которые могут быть использованы декодером 216 для восстановления входных данных 206, могут представлять собой данные с относительно высоким уровнем качества. На основании уровня квантования декодер 110 может принимать решение в отношении того, как выполнять обратное квантование по уже квантованным остаточным данным 216. Например, в декодере 110 принятый набор остаточных данных 216 может иметь значения в первом диапазоне квантования, который затем расширяется до большего диапазона значений после обратного квантования. Данные 852 квантования могут быть использованы для выполнения этой операции.

В некоторых примерах данные 852 квантования указывают уровень квантования данных коррекции. Для заданного изображения данные коррекции могут быть использованы для исправления ошибок при кодировании и/или декодировании представления с относительно низким уровнем качества, как описано со ссылкой на фиг. 2 выше. Таким образом, данные, которые могут быть использованы декодером 110 для восстановления входных данных 206, могут представлять собой данные с относительно низким уровнем качества.

В некоторых примерах данные 852 квантования указывают первый уровень квантования и второй уровень квантования. Первый уровень квантования представляет собой уровень квантования, применяемый к остаточным данным 216. Второй уровень квантования представляет собой уровень квантования, применяемый к данным коррекции. Таким образом, как остаточные данные 216, так и данные коррекции могут быть квантованы с использованием индивидуально определенных уровней квантования, которые указаны с помощью данных 852 квантования. Таким образом, данные 852 квантования могут содержать уровни, с которыми одни или более из данных коррекции и остаточных данных были квантованы в кодере (например, после преобразования, как описано в настоящем документе), которые затем отправляют в декодер, чтобы обеспечить обратное квантование и восстановление исходных значений, связанных с данными коррекции и остаточными данными (восстановленные значения могут не совпадать с исходными значениями до квантования в кодере из-за потери информации в ходе процесса квантования).

Данные 852 квантования могут содержать целое количество байтов. В некоторых примерах данные 852 квантования содержат два байта. В некоторых примерах данные 852 квантования содержат четыре байта.

В этом примере набор параметров конфигурации изображения содержит параметр 853 типа изображения. Параметр 853 типа изображения содержится в дополнительном наборе параметров конфигурации, включение которых указано параметром 851 использования остаточных данных. Параметр 853 типа изображения определяет, соответствует ли заданное изображение кадру или полю. Другими словами, параметр 853 типа изображения указывает, является ли заданное изображение прогрессивным, и в этом случае параметр 853 типа изображения определяет кадр, или чересстрочным, а в этом случае параметр 853 типа изображения определяет поле. В том случае, если заданное изображение является чересстрочным, параметр 853 типа изображения также может определять тип поля, например верхнее поле или нижнее поле. Параметр 853 типа изображения может иметь длину в битах, равную одному или двум битам.

В этом примере набор параметров конфигурации изображения включает параметр 854 временного обновления. Параметр 854 временного обновления содержится в дополнительном наборе параметров конфигурации, включение которых указано параметром 851 использования остаточных данных. Параметр 854 временного обновления используют для указания того, когда должны быть обновлены временные данные. Например, значение параметра 854 временного обновления может указывать, следует ли быть очищен буфер, хранящий временные данные. Например, как описано в настоящем документе, остаточные данные (или преобразованные остаточные данные) для текущего кадра видео могут быть изменены на основании остаточных данных (или преобразованных остаточных данных) для предыдущего кадра видео. Последние данные могут храниться в буфере и могут быть использованы для изменения

предшествующих данных. Это может итеративно повторяться для каждого кадра. В одном случае в кодере может быть выполнено сравнение текущего набора данных и содержимого буфера, и вычислена разность. В декодере временная обработка может включать добавление содержимого буфера к принятым данным для восстановления остаточных или преобразованных остаточных данных. В данном случае значение параметра 854 временного обновления может указывать декодеру, что данные, относящиеся к предыдущему кадру, которые хранятся в буфере, должны быть очищены. Это фактически может указывать на то, что временную обработку не нужно применять (например, значение 0, добавленное к принятым данным, не изменит принятые данные). Временные данные включают данные, которые могут быть использованы в ходе временной обработки, например, данные, основанные на множестве изображений. Длина в битах значения параметра 854 временного обновления может составлять один бит.

Параметры конфигурации набора изображений могут содержать больше, меньше параметров или параметры, отличные от описанных выше со ссылкой на фиг. 8.

Сообщение 800 с конфигурацией изображения выводится для множества изображений в наборе изображений. В некоторых примерах количество изображений в указанном множестве изображений равно количеству изображений в наборе изображений. Другими словами, сообщение 800 с конфигурацией изображения может выводиться для каждого изображения в наборе изображений. В других примерах количество изображений во множестве изображений меньше количества изображений в наборе изображений.

Количество изображений во множестве изображений может быть основано на показателе оптимизации. Показатель оптимизации может представлять собой показатель искажения в зависимости от скорости передачи.

Сообщение 700 с конфигурацией набора изображений и сообщение 800 с конфигурацией изображения могут выводиться для заданного набора изображений. Количество раз, которое необходимо вывести сообщение 700 с конфигурацией набора изображений для набора изображений, меньше количества раз, когда необходимо вывести сообщение 800 с конфигурацией изображения для набора изображений. Например, сообщение 700 с конфигурацией набора изображений может выводиться только один раз для набора изображений, а сообщение 800 с конфигурацией изображения может выводиться для каждого изображения в наборе изображений. Таким образом, сообщение 800 с конфигурацией изображения выводится чаще, чем сообщение 700 с конфигурацией набора изображений.

При относительно более частом выводе сообщения 800 с конфигурацией изображения используется больше данных, чем при относительно более редком выводе сообщения 800 с конфигурацией изображения. Однако при относительно частом выводе сообщения 800 с конфигурацией изображения декодер 110 получает обновленную информацию, которая позволяет декодеру 110 выполнять восстановление изображения с высокой степенью точности и/или надежности. Набор параметров конфигурации изображения, содержащийся в сообщении 800 с конфигурацией изображения, может отличаться, например, для разных изображений. Таким образом, существует компромисс между объемом данных, передаваемых на декодер 110, и точностью и/или надежностью, с которой декодер 110 может выполнять восстановление с использованием полученных данных. Для сообщения 700 с конфигурацией набора изображений и сообщения 800 с конфигурацией изображения оптимальный результат такого компромисса состоит в том, что отправка сообщения 800 с конфигурацией изображения происходит чаще, чем отправка сообщения 700 с конфигурацией набора изображений. Таким образом, отправка параметров конфигурации набора изображений с относительно низкой частотой и отправка параметров конфигурации изображения с относительно высокой частотой может позволить уменьшить объем информации, отправляемой в течение заданного периода времени, в то же время позволяя декодеру 110 выполнять восстановление с заданным уровнем точности и/или надежностью.

На фиг. 9 схематично показан пример сообщения 900 с конфигурацией. Сообщение 900 с конфигурацией может представлять собой часть данных 310 конфигурации. Сообщение 900 с конфигурацией может быть сгенерировано кодером 108.

Некоторые элементы, изображенные на фиг. 9, аналогичны элементам, показанным на фиг. 5. Числовые значения соответствующих ссылочных позиций увеличены на 400 и, таким образом, используются для аналогичных элементов.

Сообщение 900 с конфигурацией является более конкретным примером сообщения 500 с конфигурацией. Сообщение 900 с конфигурацией содержит часть-заголовок 910 и часть-полезную нагрузку 920. Часть-заголовок 910 содержит параметр 912 типа сообщения и параметр 914 размера полезной нагрузки. Часть-полезная нагрузка 920 содержит набор параметров конфигурации. Набор параметров конфигурации относится к обработке остаточных данных 216. В некоторых примерах сообщение 900 с конфигурацией имеет длину, равную целому количеству байтов.

Сообщение 900 с конфигурацией относится к типу сообщений с закодированными данными. Сообщение с закодированными данными может соответствовать заданному изображению в наборе изображений. Таким образом, для каждого изображения в наборе изображений может быть сгенерировано и/или выведено другое сообщение с закодированными данными.

Часть-полезная нагрузка 920 содержит набор параметров 950 конфигурации. Часть-полезная на-

грузка 920 также может содержать данные 960, основанные на остаточных данных 216 для заданного изображения. Данные 960, основанные на остаточных данных 216, могут содержать сами остаточные данные 216. В некоторых примерах данные 960, основанные на остаточных данных 216, содержат преобразованные остаточные данные. Преобразованные остаточные данные включают данные, которые являются выходными данными операции преобразования, выполненной в отношении остаточных данных 216. Операция преобразования может представлять собой операцию направленной декомпозиции, которая предполагает использование корреляции между остаточными элементами в остаточных данных 216.

В некоторых примерах данные 960, основанные на остаточных данных 216, содержат закодированные данные. Закодированные данные могут представлять собой закодированные остаточные данные или закодированные преобразованные остаточные данные. Закодированные данные могут быть получены с использованием одной или обеих из операции кодирования по длинам серий и операции кодирования по алгоритму Хаффмана (Huffman). Как описано выше, остаточные данные или преобразованные остаточные данные могут быть квантованы перед кодированием.

В некоторых примерах данные 960, основанные на остаточных данных 216, хранятся в одном или более элементах переменной длины, каждый из которых содержит целое количество байтов, как описано выше.

В этом примере набор параметров 950 конфигурации указывает, получены ли закодированные данные с использованием операции кодирования по алгоритму Хаффмана, операции кодирования по длинам серий или и того, и другого. В некоторых примерах набор параметров 950 конфигурации указывает, выводятся ли закодированные данные для заданного изображения. В некоторых примерах набор 950 параметров конфигурации относится к размеру закодированных данных. Например, параметр конфигурации может относиться к размеру данных, закодированных по длинам серий, и/или размеру данных, закодированных по алгоритму Хаффмана.

Сообщение 900 с закодированными данными выводится для множества изображений в наборе изображений. В некоторых примерах количество изображений в указанном множестве изображений равно количеству изображений в наборе изображений. Другими словами, сообщение 900 с закодированными данными может выводиться для каждого изображения в наборе изображений. В других примерах количество изображений во множестве изображений меньше количества изображений в наборе изображений. Количество изображений во множестве изображений может быть основано на показателе оптимизации. Показатель оптимизации может представлять собой показатель искажения в зависимости от скорости передачи.

В некоторых примерах сообщение 900 с закодированными данными генерируется и/или выводится выборочно. Будет ли сообщение с закодированными данными сгенерировано и/или выведено для данного изображения, может зависеть от того, необходимо ли восстанавливать входные данные 206 для заданного изображения с использованием остаточных данных 216 для заданного изображения. Необходимо ли восстанавливать входные данные 206 для заданного изображения, может быть указано с помощью параметра 851 использования остаточных данных в соответствующем сообщении 800 с конфигурацией изображения для заданного изображения. Объем передаваемой информации может быть уменьшен путем выборочного вывода сообщения 900 с закодированными данными по сравнению со случаем, когда сообщение 900 с закодированными данными выводится не выборочно.

Сообщения с конфигурацией, генерируемые кодером 108, могут относиться к другим типам сообщений, отличным от описанных выше. Еще один пример типа сообщения представляет собой сообщение с метаданными кодера. Сообщение с метаданными кодера содержит метаданные, относящиеся к одной или более характеристикам кодера. Еще один пример типа сообщения представляет собой сообщение управления декодером. Сообщение управления декодером содержит команды для управления декодером. Сообщение управления декодером может содержать параметр конфигурации, относящийся к сглаживанию переходов на изображении. Еще один пример типа сообщения представляет собой исследовательское сообщение. Исследовательское сообщение содержит определенную пользователем полезную нагрузку для исследовательских целей.

На фиг. 10 показан пример способа 1000 обработки данных конфигурации. Способ 1000 может выполняться устройством, содержащим кодер, такой как первое устройство 102, описанное выше.

На этапе 1010 для каждого изображения в наборе изображений получают представление изображения с относительно низким уровнем качества с использованием первого представления изображения с относительно высоким уровнем качества. Представление с относительно низким уровнем качества выводится для обработки декодером 110.

На этапе 1020 выводятся остаточные данные для одного или более выбранных изображений в наборе изображений. Остаточные данные могут быть использованы для восстановления первого представления с использованием второго представления изображения с относительно высоким уровнем качества, причем второе представление основано на представлении изображения с относительно низким уровнем качества. Остаточные данные получают на основании первого и второго представления.

В некоторых примерах одно или более выбираемых изображений включают все изображения в наборе изображений. В некоторых примерах одно или более выбираемых изображений выбирают на осно-

вании содержимого изображения. Например, если первое представление первого изображения содержит относительно большое количество визуальных деталей, может быть определено, что остаточные данные для первого изображения необходимо использовать и/или вывести для возможности восстановления первого представления первого изображения. И напротив, если первое представление второго изображения содержит относительно небольшое количество визуальных деталей, например, если второе изображение соответствует "черному экрану", может быть определено, что нет необходимости в использовании и/или выводе остаточных данных для второго изображения. Использование остаточных данных для восстановления первого представления второго изображения может существенно не улучшить отображаемое визуальное качество второго изображения из-за отсутствия в нем визуальных деталей. Выборочное использование и/или выборочный вывод остаточных данных может обеспечить уменьшение количества информации, передаваемой и/или обрабатываемой в течение заданного периода времени, по сравнению с невыборочным использованием и/или выводом остаточных данных.

На этапе 1030 данные конфигурации выводятся для обработки декодером 110. Данные конфигурации позволяют декодеру 110 определить, необходимо ли восстановить первое представление заданного изображения в наборе изображений с использованием остаточных данных для заданного изображения. Заданное изображение может содержаться в одном или более выбранных изображениях. Данные конфигурации могут выводиться для каждого изображения в наборе изображений. Данные конфигурации могут содержать сообщение 800 с конфигурацией изображения. Данные конфигурации могут, например, содержать параметр 851 использования остаточных данных. В некоторых примерах данные конфигурации содержат сообщение 900 с закодированными данными.

В некоторых примерах количество раз, которое необходимо вывести данные конфигурации для набора изображений, меньше количества изображений в наборе изображений. Данные конфигурации могут соответствовать множеству изображений в наборе изображений. Данные конфигурации могут содержать, например, сообщение 700 с конфигурацией набора изображений. В некоторых примерах количество раз, которое выводятся данные конфигурации для набора изображений, основано на показателе оптимизации.

На фиг. 11 показан пример способа 1100 для использования при обработке представлений изображения. Способ 1100 может выполняться декодером 110, описанным выше.

На этапе 1110 принимаются данные, которые могут быть использованы для получения представления изображения с относительно низким уровнем качества. Представление с относительно низким уровнем качества выводится кодером 108 с использованием первого представления изображения с относительно высоким уровнем качества и принимается декодером 110. В некоторых примерах изображение является одним из набора изображений. Набор изображений может представлять собой часть видеосигнала. В некоторых примерах принятые данные обрабатываются с получением представления с относительно низким уровнем качества. Например, принятые данные могут быть декодированы с получением представления с относительно низким уровнем качества.

На этапе 1120 определяется, были ли приняты заданные данные конфигурации. Заданные данные конфигурации соответствуют изображению.

Если на этапе 1120 определяется, что заданные данные конфигурации, соответствующие изображению, были приняты, на этапе 1130 выполняется восстановление первого представления изображения. Первое представление восстанавливается с использованием заданных данных конфигурации. Первое представление восстанавливается с использованием второго представления изображения с относительно высоким уровнем качества. В некоторых примерах представление с относительно низким уровнем качества обрабатывают с получением второго представления. Например, второе представление может быть получено путем повышения дискретизации представления с относительно низким уровнем качества. Первое представление может быть восстановлено путем объединения второго представления с остаточными данными.

На этапе 1135 выводится первое представление. Например, первое представление может быть выведено для отображения на устройстве отображения.

Если на этапе 1120 определяется, что заданные данные конфигурации, соответствующие изображению, не были приняты, восстановление первого представления изображения не выполняется. Вместо этого на этапе 1140 представление с относительно низким уровнем качества обрабатывается с получением второго представления. Второе представление может быть получено путем повышения дискретизации представления с относительно низким уровнем качества.

На этапе 1145 выводится второе представление. Например, второе представление может быть выведено для отображения на устройстве отображения.

В некоторых примерах, если определено, что заданные данные конфигурации, соответствующие изображению, не были приняты, второе представление не выводится. Например, представление с относительно низким уровнем качества может быть выведено вместо второго представления. В некоторых примерах выводятся данные, основанные на представлении с относительно низким уровнем качества. Данные, основанные на представлении с относительно низким уровнем качества, отличаются от первого представления и второго представления.

В том случае, если изображение является одним из набора изображений, заданные данные конфи-

гурации могут соответствовать множеству изображений в наборе изображений. Заданные данные конфигурации могут приниматься не с каждым изображением в наборе изображений. В одном примере заданные данные конфигурации принимаются только с первым изображением в наборе изображений. Заданные данные конфигурации могут быть применимыми к каждому последующему изображению в наборе изображений. Таким образом, если первое изображение в наборе изображений не принято декодером 110, заданные данные конфигурации для набора изображений также могут быть не приняты. Заданные данные конфигурации могут представлять собой данные конфигурации набора изображений. Заданные данные конфигурации могут содержать сообщение 700 с конфигурацией набора изображений.

В том случае, если набор изображений является частью потокового видео, декодер может вводить видеопоток частично в виде набора изображений. В таком случае, хотя первое представление не восстанавливается вследствие отсутствия принимаемых заданных данных конфигурации, представление с относительно высоким уровнем качества, а именно второе представление, все же может выводиться. Вывод второго представления для отображения обеспечивает снижение потенциальной потери визуального качества, вызванной отсутствием принимаемых заданных данных конфигурации, по сравнению со случаем, когда представление с относительно высоким уровнем качества не выводится. Вывод второго представления может считаться частью "минимального рабочего режима", в отличие от вывода первого представления, которое может считаться частью "полного рабочего режима". В том случае, если декодер 110 частично вводит видеопоток в виде набора изображений, для оставшейся части набора изображений может быть использован "минимальный рабочий режим". Затем, при приеме первого изображения в следующем наборе изображений также принимаются заданные данные конфигурации для следующего набора изображений. Таким образом, следующий набор изображений может быть обработан в соответствии с "полным рабочим режимом".

На фиг. 12 показана структурная схема примера устройства 1200.

В одном примере устройство 1200 содержит кодер. Еще в одном примере устройство 1200 содержит декодер.

Примеры устройства 1200 включают, среди прочих, переносной компьютер, персональную компьютерную систему, беспроводное устройство, базовую станцию, телефонное устройство, настольный компьютер, портативный компьютер, ноутбук, нетбук, суперкомпьютерную систему, карманный компьютер, рабочую станцию, сетевой компьютер, сервер приложений, устройство хранения данных, устройство бытовой электроники, такое как камера, видеокамера, мобильное устройство, игровая приставка, карманное игровое видеоприложение, или в более общем смысле вычислительное или электронное устройство любого типа.

В этом примере устройство 1200 содержит один или более процессоров 1201, выполненных с возможностью обработки информации и/или команд. Один или более процессоров 1201 могут включать центральный процессор (ЦП). Один или более процессоров 1201 подключены к шине 1202. Операции, выполняемые одним или более процессорами 1201, могут выполняться аппаратным обеспечением и/или программным обеспечением. Один или более процессоров 1201 могут включать множество процессоров, расположенных в одном месте, или множество процессоров, расположенных в разных местах.

В этом примере устройство 1200 содержит выполненное с возможностью использования на компьютере энергозависимое запоминающее устройство 1203, выполненное с возможностью хранения информации и/или команд для одного или более процессоров 1201. Выполненное с возможностью использования на компьютере энергозависимое запоминающее устройство 1203 подключено к шине 1202. Выполненное с возможностью использования на компьютере энергозависимое запоминающее устройство 1203 может содержать оперативное запоминающее устройство (ОЗУ).

В этом примере устройство 1200 содержит выполненное с возможностью использования на компьютере энергонезависимое запоминающее устройство 1204, выполненное с возможностью хранения информации и/или команд для одного или более процессоров 1201. Выполненное с возможностью использования на компьютере энергонезависимое запоминающее устройство 1204 подключено к шине 1202. Энергонезависимое запоминающее устройство 1204 может содержать постоянное запоминающее устройство (ПЗУ).

В этом примере устройство 1200 содержит один или более блоков 1205 хранения данных, выполненных с возможностью хранения информации и/или команд. Один или более блоков 1205 хранения данных подключены к шине 1202. Один или более блоков 1205 хранения данных могут, например, содержать магнитный или оптический диск и дисковый накопитель или полупроводниковый накопитель (solid-state drive, SSD).

В этом примере устройство 1200 содержит одно или более устройств 1206 ввода/вывода (input/output, I/O), выполненных с возможностью передачи информации на один или более процессоров 1201 и/или от них. Одно или более устройств 1206 ввода/вывода подключены к шине 1202. Одно или более устройств 1206 ввода/вывода могут содержать по меньшей мере один сетевой интерфейс. по меньшей мере один сетевой интерфейс может обеспечивать для устройства 1200 обмен данными 110 одной или более сетями передачи данных. Примеры сетей передачи данных включают, среди прочих, Интернет и локальную сеть (LAN). Одно или более устройств 1206 ввода/вывода могут позволить пользова-

телю выполнять ввод данных в устройство 1200 с помощью одного или более устройств ввода (не показаны). Одно или более устройств ввода могут включать, например, пульт дистанционного управления, одну или более физических кнопок и т.д. Одно или более устройств 1206 ввода/вывода могут обеспечивать предоставление информации пользователю с помощью одного или более устройств вывода (не показаны). Одно или более устройств вывода могут, например, включать экран дисплея.

Также показаны различные другие объекты, относящиеся к устройству 1200. Например, если они присутствуют, операционная система 1207, модуль 1208 обработки сигналов данных, один или более дополнительных модулей 1209 и данные 1210 показаны как находящиеся в одном из выполненного с возможностью использования на компьютере энергозависимого запоминающего устройства 1203, выполненного с возможностью использования на компьютере энергонезависимого запоминающего устройства 1204 и одного или более блоков 1205 хранения данных, или в их комбинации. Модуль 1208 обработки сигнала данных может быть реализован с применением компьютерного программного кода, хранящегося в ячейках запоминающего устройства в выполненном с возможностью использования на компьютере энергонезависимом запоминающем устройстве 1204, в машиночитаемом носителе данных в одном или более блоках 1205 хранения данных и/или в других материальных машиночитаемых носителях данных. Примеры материальных машиночитаемых носителей данных включают, среди прочих, оптический носитель (например, CD-ROM, DVD-ROM или Blu-ray), карту памяти на электрически перепрограммируемом постоянном запоминающем устройстве с параллельным стиранием, гибкий диск или жесткий диск, либо любой другой носитель, выполненный с возможностью хранения машиночитаемых команд, такой как программное обеспечение, реализованное с помощью аппаратных средств, или микрокод по меньшей мере в одной микросхеме ПЗУ, ОЗУ или программируемом ПЗУ (НИЗУ), либо выполненный в виде специализированной заказной интегральной схемы (ASIC).

Таким образом, устройство 1200 может содержать модуль 1208 обработки сигнала данных, который может исполняться одним или более процессорами 1201. Модуль 1208 обработки сигнала данных может быть выполнен таким образом, чтобы содержать команды для реализации по меньшей мере некоторых операций, описанных в настоящем документе. В ходе функционирования один или более процессоров 1201 иницируют, запускают, исполняют, интерпретируют или иным образом выполняют команды в модуле 1208 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, расположенные на или в носителе, выполненные с возможностью применения примеров на практике. Носитель может представлять собой любой объект или устройство, на которых может содержаться программа.

Следует понимать, что устройство 1200 может содержать большее, меньшее количество компонентов и/или компоненты, отличные от изображенных на фиг. 12.

Устройство 1200 может быть расположено в одном местоположении или может быть распределено в множестве местоположений. Такие местоположения могут быть локальными или удаленными.

Описанные в настоящем документе способы могут быть реализованы в программном или аппаратном обеспечении, или могут быть реализованы с использованием комбинации программного и аппаратного обеспечения. Они могут включать конфигурирование устройства для выполнения и/или поддержки любого или всех из способов, описанных в настоящем документе.

Описанные в настоящем документе примеры относятся к синтаксису для использования данных конфигурации, относящихся к обработке остаточных данных, причем остаточные данные могут быть использованы при восстановлении изображения. Синтаксис обеспечивает структуру обработки, позволяющую кодеру получать и выводить данные конфигурации, и позволяющую декодеру принимать и обрабатывать данные конфигурации. Структура синтаксиса облегчает генерацию и обработку сообщений с конфигурацией, таких как сообщения, описанные выше. Декодер использует принятые данные конфигурации для восстановления представлений изображений, которые, например, представляют собой часть видеопотока, с высоким уровнем качества. Описанная в настоящем документе структура синтаксиса позволяет декодеру выполнять такое восстановление с относительно высокой степенью эффективности, надежности и точности.

Описанная в настоящем документе структура синтаксиса позволяет снизить размер служебных данных в передаваемых данных по сравнению с некоторыми известными системами. Описанная в настоящем документе структура синтаксиса может включать служебные данные размером не более 3 байтов для заданного изображения, если остаточные данные не нужно использовать для восстановления представления заданного изображения с высоким уровнем качества. Если остаточные данные необходимо использовать для восстановления представления с высоким уровнем качества, могут потребоваться служебные данные размером не более 43 байтов для заданного изображения. Это значительное преимущество по сравнению с некоторыми известными системами, в которых служебные данные для одного изображения могут составлять от 108 до 188 байтов. Таким образом, объем информации, генерируемой, хранимой, обрабатываемой и/или передаваемой в течение заданного периода времени, значительно со-

кращается благодаря использованию описанной в настоящем документе структуры синтаксиса.

Следует понимать, что любой признак, описанный в отношении одного варианта осуществления, может быть использован отдельно или в сочетании с другими описанными признаками, а также может быть использован в сочетании с одним или более признаками любого другого из вариантов осуществления, или любым сочетанием любого другого из вариантов осуществления. Кроме того, эквиваленты и изменения, не описанные выше, также могут быть использованы без отступления от объема настоящего изобретения, который определен в сопроводительной формуле изобретения.

## ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ обработки информации о конфигурации для обработки остаточных данных при декодировании изображений, включающий:

получение для заданного изображения в наборе изображений остаточных данных, которые могут быть использованы декодером для восстановления первого представления заданного изображения с первым уровнем качества в многоуровневой иерархии, содержащей множество уровней качества, с использованием второго представления заданного изображения, причем второе представление основано на представлении заданного изображения со вторым, более низким уровнем качества в многоуровневой иерархии, при этом остаточные данные получены из сравнения первого представления и второго представления;

генерацию данных конфигурации, относящихся к обработке остаточных данных, причем данные конфигурации содержат параметр временной обработки, который указывает, была ли выполнена временная обработка для заданного изображения, при этом указанная временная обработка связана с восстановлением, для заданного изображения, первого представления с использованием второго представления и остаточных данных, при этом выполнение временной обработки включает использование данных, основанных на множестве изображений в наборе изображений; и

вывод данных конфигурации для обработки декодером для обеспечения возможности восстановления декодером, для заданного изображения, первого представления с использованием второго представления и остаточных данных,

причем количество раз, которое параметр временной обработки выводится для набора изображений, меньше количества изображений в наборе изображений, и

при выводе параметра временной обработки, для каждого изображения в наборе изображений выводится дополнительный параметр временной обработки.

2. Способ по п.1, согласно которому дополнительный параметр временной обработки представляет собой параметр временного обновления.

3. Способ по п.2, согласно которому параметр временного обновления указывает, когда необходимо обновить временные данные, причем временные данные включают данные, которые могут быть использованы при временной обработке.

4. Способ по п.3, согласно которому параметр временного обновления указывает, необходимо ли очистить буфер, хранящий временные данные.

5. Способ по любому из пп.2-4, согласно которому длина в битах значения параметра временного обновления составляет один бит.

6. Способ по любому из пп.1-5, согласно которому параметр временной обработки выводится для набора изображений один раз.

7. Способ по любому из пп.1-6, согласно которому значение параметра временной обработки имеет длину в битах, равную одному биту.

8. Способ по любому из пп.1-7, согласно которому данные, основанные на множестве изображений, указывают степень временной корреляции между остаточными данными для заданного изображения и остаточными данными для одного или более дополнительных изображений в наборе изображений.

9. Способ по п.8, согласно которому данные содержат данные временной корреляции, основанные на разности между остаточными данными для заданного изображения и остаточными данными для одного или более дополнительных изображений в наборе изображений.

10. Способ по п. 8 или 9,

согласно которому каждое из изображений в наборе изображений соответствует разной временной выборке видео, а

одно или более дополнительных изображений соответствуют одной или более предыдущим временным выборкам в видео относительно временной выборки, соответствующей заданному изображению.

11. Способ по любому из пп.1-10, согласно которому параметр временной обработки сохраняется в заданном бите в заданном байте данных конфигурации.

12. Способ обработки информации о конфигурации для обработки остаточных данных при кодировании изображений, включающий:

прием данных конфигурации, относящихся к обработке остаточных данных, причем остаточные данные были получены для заданного изображения в наборе изображений на основании первого пред-



ставления изображения с первым уровнем качества в многоуровневой иерархии, имеющей множество уровней качества, и второго представления изображения, причем второе представление основано на представлении заданного изображения со вторым, более низким уровнем качества в многоуровневой иерархии, а остаточные данные могут быть использованы для восстановления первого представления с использованием второго представления, при этом остаточные данные получены из сравнения первого представления и второго представления; и

обработку данных конфигурации для восстановления, для заданного изображения, первого представления с использованием второго представления и остаточных данных,

причем данные конфигурации содержат параметр временной обработки, который указывает, была ли выполнена временная обработка для заданного изображения, причем указанная временная обработка связана с восстановлением первого представления, и

при этом выполнение временной обработки включает использование данных, основанных на множестве изображений в наборе изображений,

причем количество раз, которое параметр временной обработки принимается для набора изображений, меньше количества изображений в наборе изображений, и

при приеме параметра временной обработки для каждого изображения в наборе изображений принимается дополнительный параметр временной обработки.

13. Способ по п.12, согласно которому дополнительный параметр временной обработки представляет собой параметр временного обновления.

14. Способ по п.13, согласно которому параметр временного обновления указывает, когда необходимо обновить временные данные, причем временные данные включают данные, которые могут быть использованы при временной обработке.

15. Способ по п.14, согласно которому параметр временного обновления указывает, необходимо ли очистить буфер, хранящий временные данные.

16. Способ по любому из пп.13-15, согласно которому длина в битах значения параметра временного обновления составляет один бит.

17. Способ по любому из пп.12-16, согласно которому параметр временной обработки принимается для набора изображений один раз.

18. Способ по любому из пп.12-17, согласно которому значение параметра временной обработки имеет длину в битах, равную одному биту.

19. Способ по любому из пп.12-18, согласно которому данные, основанные на множестве изображений, указывают степень временной корреляции между остаточными данными для заданного изображения и остаточными данными для одного или более дополнительных изображений в наборе изображений.

20. Способ по п.19, согласно которому данные содержат данные временной корреляции, основанные на разности между остаточными данными для заданного изображения и остаточными данными для одного или более дополнительных изображений в наборе изображений.

21. Способ по п.19 или 20, согласно которому каждое из изображений в наборе изображений соответствует разной временной выборке видео, а

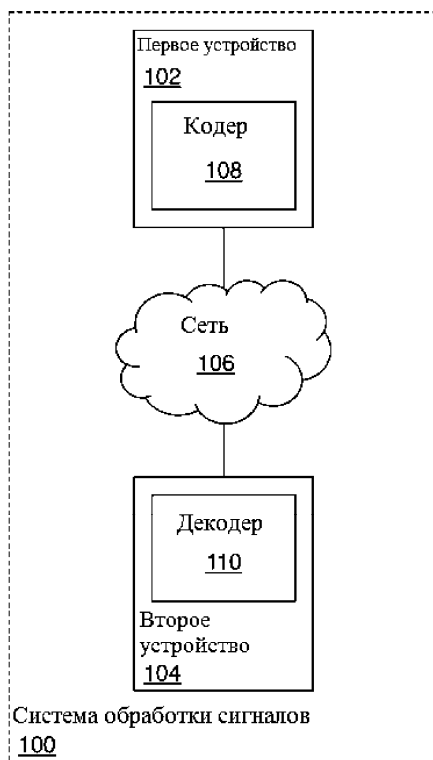
одно или более дополнительных изображений соответствуют одной или более предыдущим временным выборкам в видео относительно временной выборки, соответствующей заданному изображению.

22. Способ по любому из пп.12-21, согласно которому параметр временной обработки сохраняется в заданном бите в заданном байте данных конфигурации.

23. Устройство обработки информации о конфигурации для обработки остаточных данных, содержащее декодер изображений и выполненное с возможностью осуществления операций способа по любому из пп.1-11.

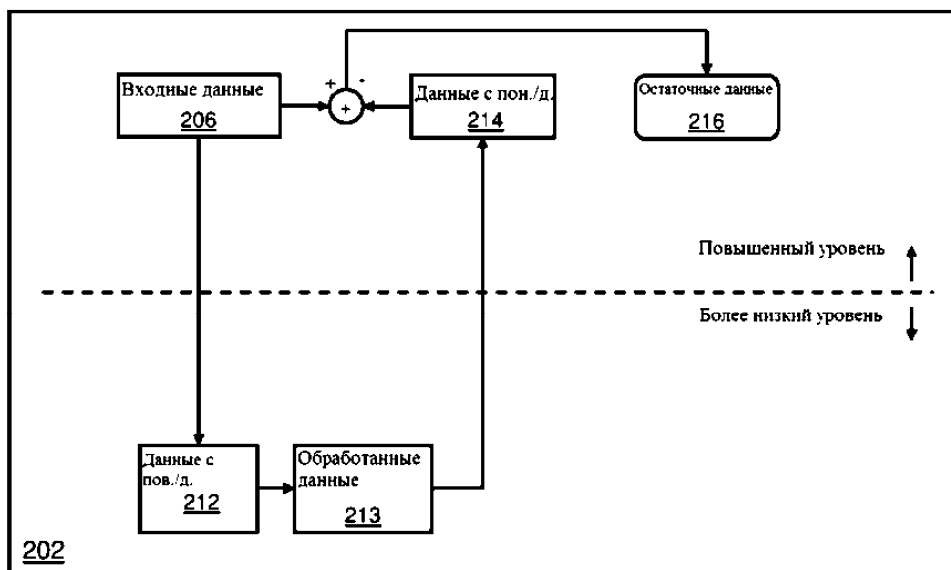
24. Устройство обработки информации о конфигурации для обработки остаточных данных, содержащее кодер изображений и выполненное с возможностью осуществления операций способа по любому из пп.12-22.

25. Машиночитаемый носитель информации, содержащий команды для кодера или декодера устройств по п.23 или 24 для осуществления операций способа по пп.1-11 или пп.12-22.



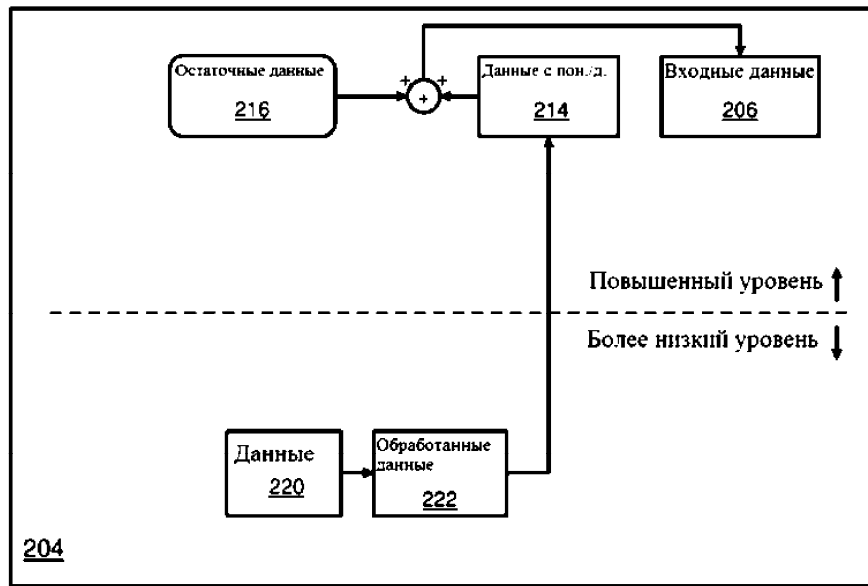
Фиг. 1

200



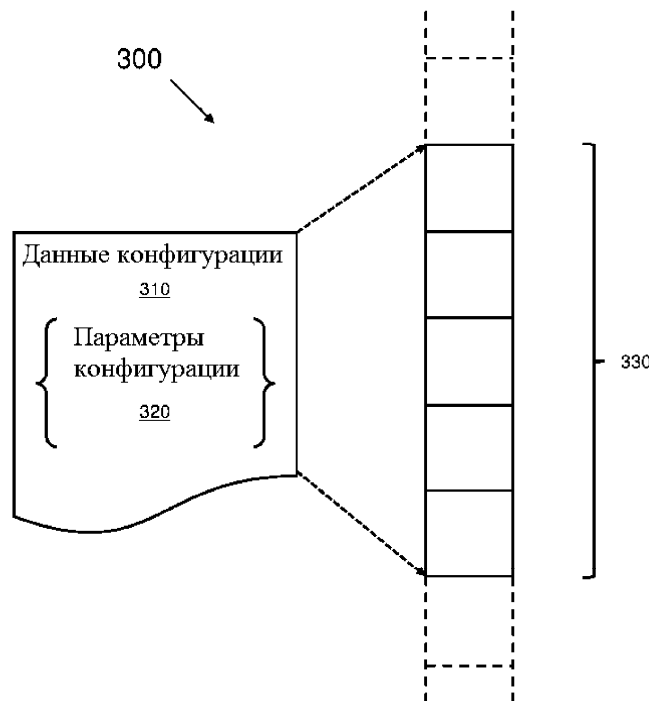
Фиг. 2А

200



Фиг. 2В

300

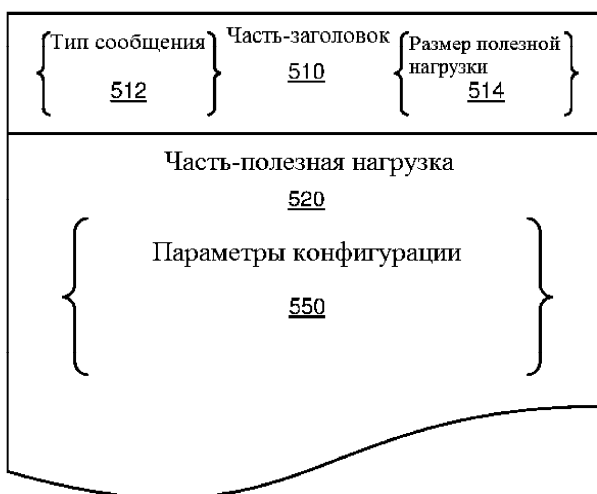


Фиг. 3



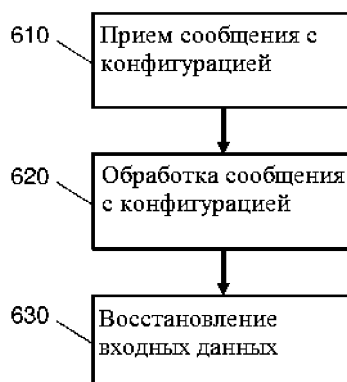
Фиг. 4

500

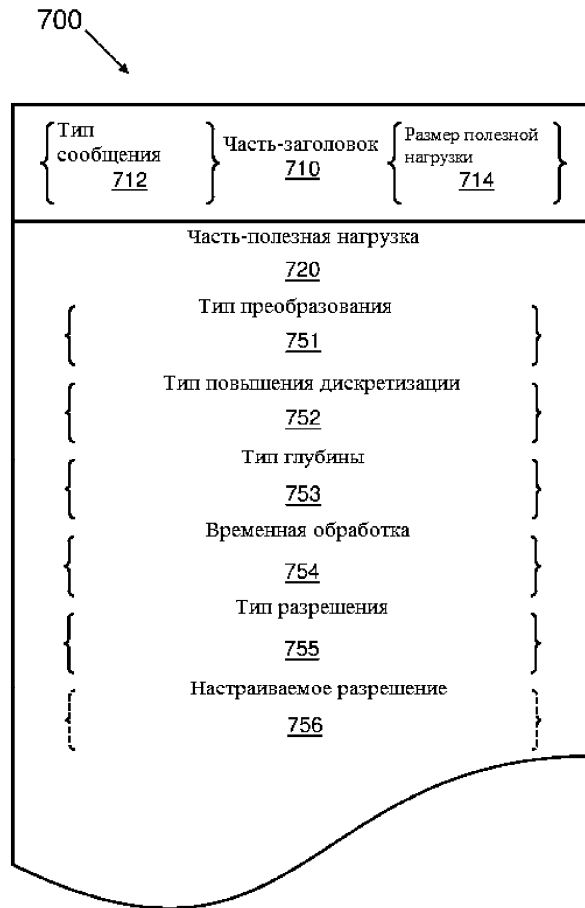


Фиг. 5

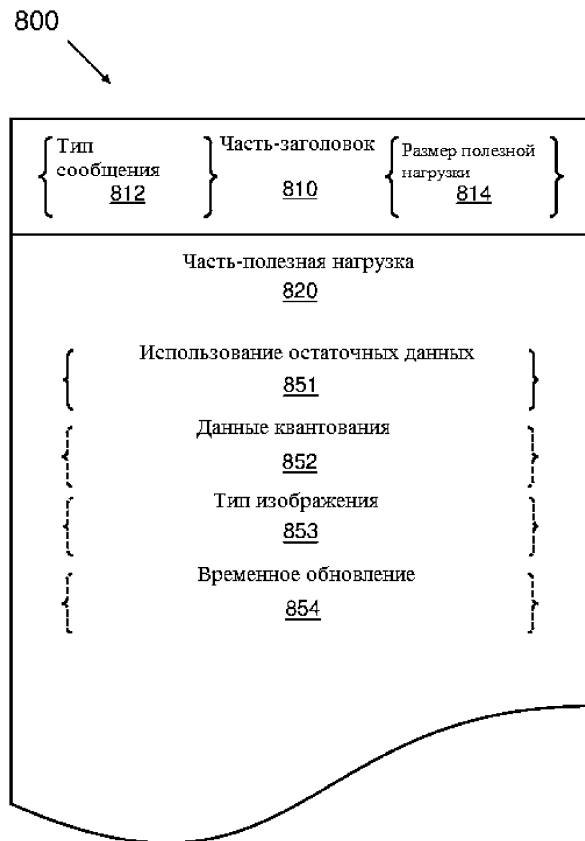
600



Фиг. 6



Фиг. 7



Фиг. 8

900

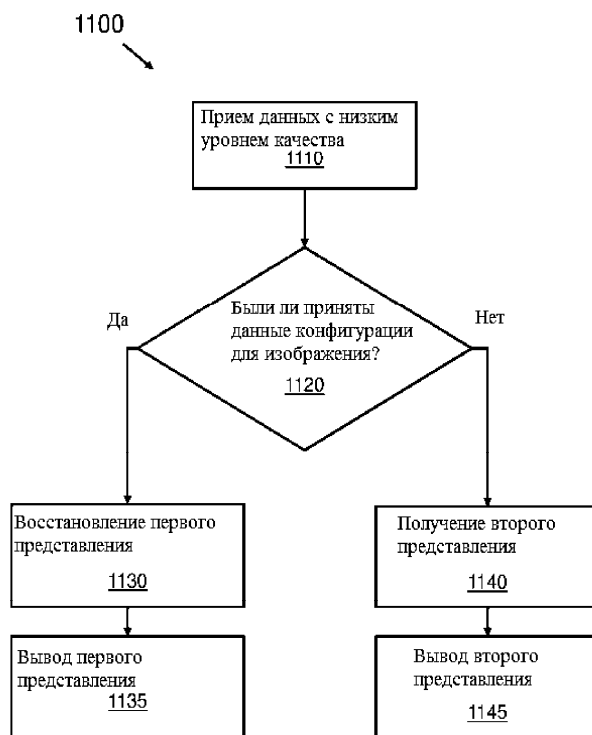


Фиг. 9

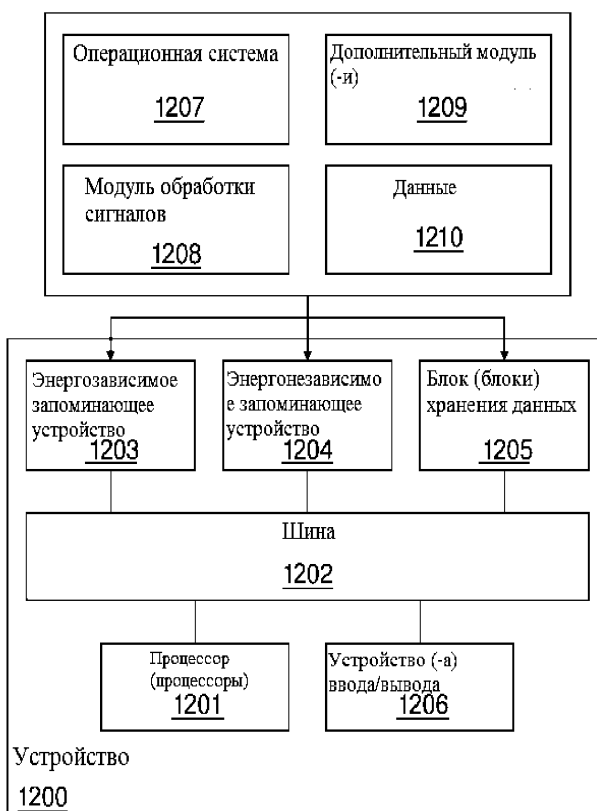
1000



Фиг. 10



Фиг. 11



Фиг. 12

