



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

## (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК  
H04N 19/105 (2022.05); H04N 19/593 (2022.05)

(21)(22) Заявка: 2021122378, 30.12.2019

(24) Дата начала отсчета срока действия патента:  
30.12.2019

Дата регистрации:  
16.12.2022

Приоритет(ы):  
(30) Конвенционный приоритет:  
31.12.2018 US 62/786,563

(45) Опубликовано: 16.12.2022 Бюл. № 35

(85) Дата начала рассмотрения заявки РСТ на  
национальной фазе: 02.08.2021

(86) Заявка РСТ:  
RU 2019/050261 (30.12.2019)

(87) Публикация заявки РСТ:  
WO 2020/141985 (09.07.2020)

Адрес для переписки:  
129090, Москва, ул. Б. Спасская, 25, стр. 3, ООО  
"Юридическая фирма Городиский и  
Партнеры"

(72) Автор(ы):  
РУФИЦКИЙ, Василий Алексеевич (CN),  
ЧЕН, Цзянле (US),  
МА, Сян (CN),  
ФИЛИППОВ, Алексей Константинович  
(CN)

(73) Патентообладатель(и):  
ХУАВЭЙ ТЕКНОЛОДЖИЗ КО., ЛТД.  
(CN)

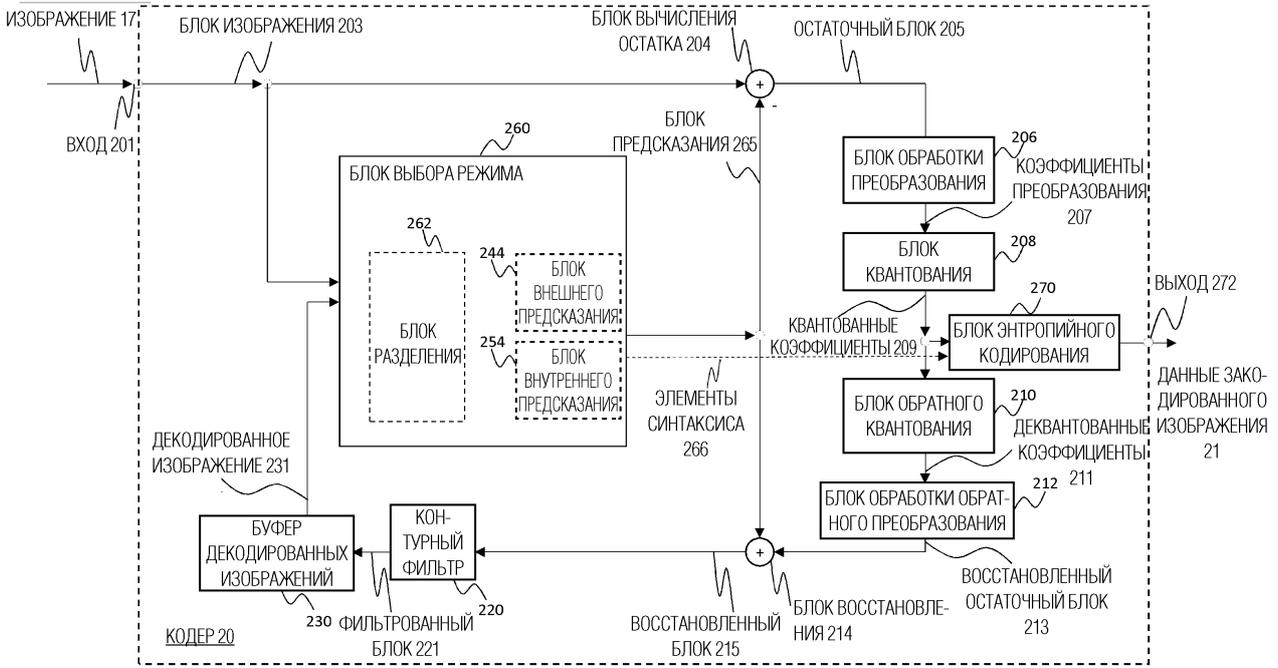
(56) Список документов, цитированных в отчете  
о поиске: WO 2016108188 A1, 07.07.2016. US  
20170221183 A1, 03.08.2017. US 20140086333 A1,  
27.03.2014. US 20120027316 A1, 02.02.2012. US  
20150100612 A1, 09.04.2015. WO 2016113458 A1,  
21.07.2016.

(54) СПОСОБ И УСТРОЙСТВО КРОСС-КОМПОНЕНТНОГО ЛИНЕЙНОГО МОДЕЛИРОВАНИЯ  
ДЛЯ ВНУТРЕННЕГО ПРЕДСКАЗАНИЯ

(57) Реферат:

Изобретение относится к средствам для кодирования видео. Технический результат заключается в повышении эффективности кодирования видео. Получают восстановленные выборки яркости. Определяют максимальные и минимальные значения выборок яркости на основе восстановленных выборок яркости. Получают разность максимального и минимального значений выборок яркости. Определяют положение старшего бита разности максимального и минимального значений

выборок яркости. Извлекают значение из справочной таблицы (LUT) с использованием набора битов в качестве индекса, причем набор битов следует за положением старшего бита разности максимального и минимального значений выборок яркости. Получают параметры  $\alpha$  и  $\beta$  линейной модели на основе извлеченного значения. Вычисляют предсказанное значение выборки цветности с использованием полученных параметров  $\alpha$  и  $\beta$  линейной модели. 8 н. и 16 з.п. ф-лы, 15 ил., 4 табл.



ФИГ. 2



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.  
*H04N 19/105* (2014.01)  
*H04N 19/593* (2014.01)

(12) **ABSTRACT OF INVENTION**

(52) CPC  
*H04N 19/105 (2022.05); H04N 19/593 (2022.05)*

(21)(22) Application: **2021122378, 30.12.2019**

(24) Effective date for property rights:  
**30.12.2019**

Registration date:  
**16.12.2022**

Priority:

(30) Convention priority:  
**31.12.2018 US 62/786,563**

(45) Date of publication: **16.12.2022** Bull. № 35

(85) Commencement of national phase: **02.08.2021**

(86) PCT application:  
**RU 2019/050261 (30.12.2019)**

(87) PCT publication:  
**WO 2020/141985 (09.07.2020)**

Mail address:  
**129090, Moskva, ul. B. Spasskaya, 25, str. 3, OOO  
"Yuridicheskaya firma Gorodisskij i Partnery"**

(72) Inventor(s):

**RUFITSKIY, Vasily Alexeevich (CN),  
CHEN, Jianle (US),  
MA, Xiang (CN),  
FILIPPOV, Alexey Konstantinovich (CN)**

(73) Proprietor(s):

**HUAWEI TECHNOLOGIES CO., LTD. (CN)**

(54) **METHOD AND DEVICE FOR CROSS-COMPONENT LINEAR MODELING FOR INTERNAL PREDICTION**

(57) Abstract:

FIELD: video encoding.

SUBSTANCE: invention relates to means for video encoding. Restored brightness samples are obtained. Maximum and minimum values of brightness samples are determined based on restored brightness samples. The difference between maximum and minimum values of brightness samples is obtained. A position of the most significant bit of the difference between maximum and minimum values of brightness samples is determined. A value is extracted from a lookup table

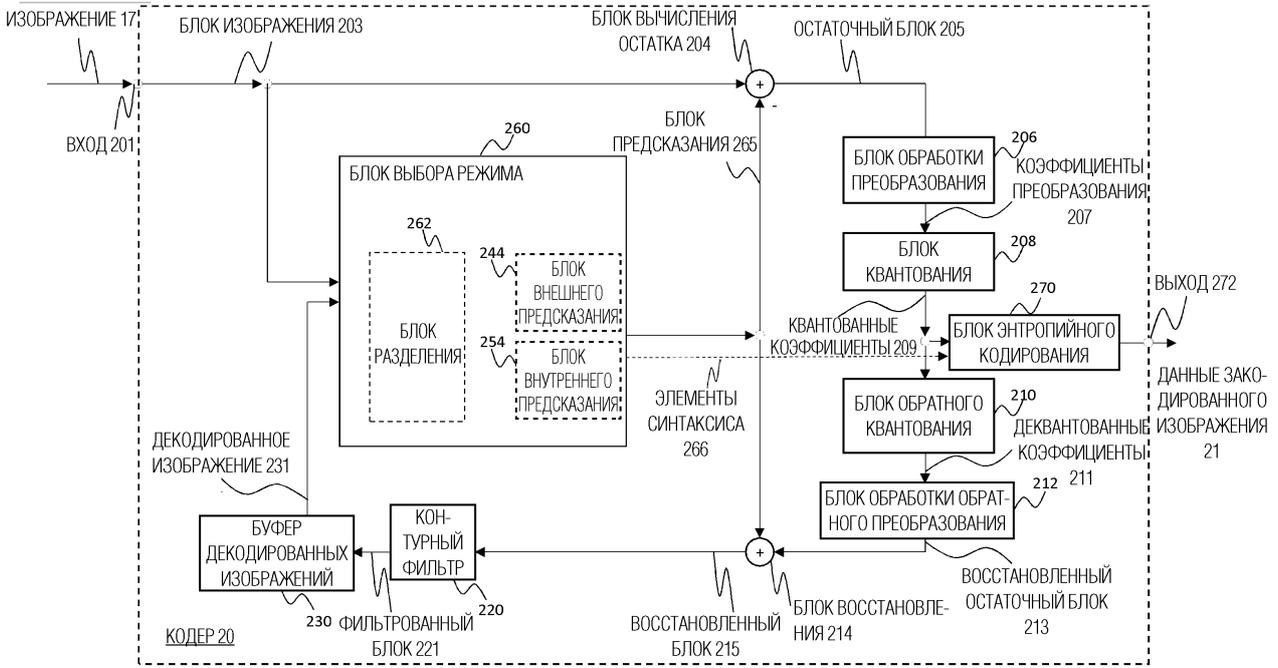
(LUT), using a set of bits as an index, wherein the set of bits follows the position of the most significant bit of the difference of maximum and minimum values of brightness samples. Parameters  $\alpha$  and  $\beta$  of a linear model are obtained based on the extracted value. A predicted value of the brightness sample is calculated using obtained parameters  $\alpha$  and  $\beta$  of the linear model.

EFFECT: increase in the efficiency of video encoding.

24 cl, 15 dwg, 4 tbl

RU 2 786 086 C1

RU 2 786 086 C1



ФИГ. 2

## ПЕРЕКРЕСТНАЯ ССЫЛКА НА СВЯЗАННЫЕ ЗАЯВКИ

Эта заявка испрашивает преимущество предварительной заявки США №. 62/786,563, поданной 31 декабря 2018 г. и озаглавленной «METHOD AND APPARATUS OF CROSS-COMPONENT LINEAR MODELING FOR INTRA PREDICTION», которая включена в  
5 настоящий документ посредством ссылки.

### Область техники

Варианты осуществления настоящей заявки (раскрытие) в целом относятся к области обработки изображений и, в частности, к внутреннему предсказанию с использованием кросс-компонентного линейного моделирования.

### 10 Уровень техники

Видеокодирование (кодирование и декодирование видео) используется в широком диапазоне применений цифрового видео, например, в широкоэмитальном цифровом телевидении, передаче видео через Интернет и мобильные сети, в диалоговых приложениях в реальном времени, таких как видеочат, видеоконференцсвязь, DVD и  
15 Blu-Ray диски, системы сбора и редактирования видеоконтента и видеокамеры приложений безопасности.

Количество видеоданных, необходимых для изображения даже относительно короткого видео, может быть существенным, что может привести к трудностям, когда данные должны передаваться в потоковом режиме или иным образом передаваться по  
20 сети связи с ограниченной пропускной способностью. Таким образом, видеоданные обычно сжимаются перед передачей по современным телекоммуникационным сетям. Размер видео также может быть проблемой, когда видео хранится на запоминающем устройстве, поскольку ресурсы памяти могут быть ограничены. Устройства сжатия видео часто используют программное обеспечение и/или оборудование в источнике  
25 для кодирования видеоданных перед передачей или хранением, тем самым уменьшая количество данных, необходимых для представления цифровых видеоизображений. Затем сжатые данные принимаются в месте назначения устройством распаковки видео, которое декодирует видеоданные. С ограниченными сетевыми ресурсами и постоянно растущими требованиями к более высокому качеству видео желательны улучшенные  
30 способы сжатия и распаковки, которые улучшают степень сжатия с минимальными потерями качества изображения или вообще без них.

### Сущность изобретения

Варианты осуществления настоящей заявки предоставляют устройства и способы для кодирования и декодирования согласно независимым пунктам формулы изобретения.

35 Вышеупомянутые и другие цели достигаются сущностью независимых пунктов формулы изобретения. Дополнительные формы реализации очевидны из зависимых пунктов формулы изобретения, описания и фигур.

Согласно первому аспекту изобретение относится к способу внутреннего предсказания выборки цветности блока путем применения кросс-компонентной  
40 линейной модели. Способ включает в себя: получение восстановленных выборок яркости; определение максимальных и минимальных значений выборок яркости на основе восстановленных выборок яркости; получение разности максимального и минимального значений выборок яркости; определение положения старшего бита разности максимального и минимального значений выборки яркости. Этот способ  
45 также включает в себя: извлечение значения из справочной таблицы (LUT) с использованием набора битов в качестве индекса, набора битов, следующих за положением старшего бита разности максимального и минимального значений выборки яркости; получение параметров  $\alpha$  и  $\beta$  линейной модели на основе извлеченного значения;

и вычисление предсказанного значения выборки цветности с использованием полученных параметров  $\alpha$  и  $\beta$  линейной модели.

Согласно первому аспекту изобретения индекс для LUT вычисляется элегантным способом, который извлекает несколько битов в двоичном представлении. В результате повышается эффективность извлечения значения из LUT.

В возможной форме реализации способа согласно первому аспекту как таковому способ получает параметры  $\alpha$  и  $\beta$  линейной модели путем умножения извлеченного значения на разность максимального и минимального значений восстановленных выборок цветности.

Поскольку эффективность извлечения значения из LUT увеличивается, величина множителя для получения параметров  $\alpha$  и  $\beta$  линейной модели сводится к минимуму.

В возможной форме реализации способа согласно первому аспекту как таковому, LUT включает в себя по меньшей мере два соседних значения, хранящихся в LUT, которые соответствуют различным шагам полученной разности, и значение этого шага увеличивается со значением разности или является константой. Индекс для LUT вычисляется элегантным способом, который извлекает несколько битов в двоичном представлении, и, соответственно, размер записи в LUT, соответствующей индексу, минимизируется. В результате размер LUT сводится к минимуму.

Устройство для внутреннего предсказания выборки цветности блока посредством применения кросс-компонентной линейной модели предоставляется согласно второму аспекту изобретения. Устройство согласно второму аспекту изобретения включает в себя блок получения, блок определения и блок вычисления. Блок получения, сконфигурированный для получения восстановленных выборок яркости. Блок определения, сконфигурированный для определения максимальных и минимальных значений выборок яркости на основе восстановленных выборок яркости. Блок получения, дополнительно сконфигурированный для получения разницы между максимальным и минимальным значениями выборки яркости. Блок определения, дополнительно сконфигурированный для определения положения старшего бита разности максимального и минимального значений выборки яркости. Блок вычисления, сконфигурированный для извлечения значения из справочной таблицы (LUT) с использованием набора битов в качестве индекса, набор битов, следующих за положением старшего бита разности значений максимальной и минимальной выборки яркости, получения параметров  $\alpha$  и  $\beta$  линейной модели на основе извлеченного значения; и вычисления предсказанного значения выборки цветности с использованием полученных параметров  $\alpha$  и  $\beta$  линейной модели.

Согласно второму аспекту изобретения устройство вычисляет индекс для LUT элегантным способом, который извлекает несколько битов в двоичном представлении. В результате повышается эффективность извлечения значения из LUT.

Согласно третьему аспекту изобретение относится к устройству для декодирования видеопотока, включающему в себя процессор и память. В памяти хранятся инструкции, которые заставляют процессор выполнять способ согласно первому аспекту или любому возможному варианту осуществления первого аспекта.

Согласно четвертому аспекту изобретение относится к устройству для кодирования видеопотока, включающему в себя процессор и память. В памяти хранятся инструкции, которые заставляют процессор выполнять способ согласно первому аспекту или любому возможному варианту осуществления первого аспекта.

Согласно пятому аспекту предлагается машиночитаемый носитель данных, на котором хранятся инструкции, которые при исполнении вызывают один или несколько

процессоров, сконфигурированных для кодирования видеоданных. Инструкции заставляют один или несколько процессоров выполнять способ согласно первому аспекту или любому возможному варианту осуществления первого аспекта.

5 Согласно шестому аспекту изобретение относится к компьютерной программе, содержащей программный код для выполнения способа согласно первому аспекту или любому возможному варианту осуществления первого аспекта при выполнении на компьютере.

10 Подробности одного или нескольких вариантов осуществления изложены на прилагаемых чертежах и в описании ниже. Другие особенности, объекты и преимущества будут очевидны из описания, чертежей и формулы изобретения.

#### КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Далее варианты осуществления изобретения описаны более подробно со ссылкой на сопровождающие фигуры и чертежи, на которых:

15 Фиг. 1А - блок-схема, показывающая пример системы кодирования видео, сконфигурированной для реализации вариантов осуществления изобретения;

Фиг. 1В является блок-схемой, показывающей другой пример системы кодирования видео, выполненной с возможностью реализации вариантов осуществления изобретения;

Фиг. 2 - блок-схема, показывающая пример видеокодера, сконфигурированного для реализации вариантов осуществления изобретения;

20 Фиг. 3 - блок-схема, показывающая примерную структуру видеodeкодера, сконфигурированного для реализации вариантов осуществления изобретения;

Фиг. 4 является блок-схемой, иллюстрирующей пример устройства кодирования или устройства декодирования;

25 Фиг. 5 является блок-схемой, иллюстрирующей другой пример устройства кодирования или устройства декодирования;

Фиг. 6 - чертеж, иллюстрирующий концепцию кросс-компонентной линейной модели для внутреннего предсказания цветности;

Фиг. 7 - чертеж, иллюстрирующий упрощенный способ в параметров линейной модели;

30 Фиг. 8 - чертеж, иллюстрирующий распределение вероятности возникновения разницы между максимальным и минимальным значениями опорных выборок яркости;

Фиг. 9 - блок-схема, иллюстрирующая примерный процесс генерирования справочной таблицы в соответствии с вариантом осуществления изобретения;

35 Фиг. 10 - чертеж с блок-схемами, иллюстрирующими вариант осуществления вывода индекса для примерной справочной таблицы;

Фиг. 11 - блок-схема, иллюстрирующая примерное внутреннее предсказание выборки цветности блока посредством применения кросс-компонентной линейной модели;

40 Фиг. 12 - блок-схема, показывающая примерную структуру устройства для внутреннего предсказания выборки цветности блока посредством применения кросс-компонентной линейной модели;

Фиг. 13 - блок-схема, показывающая примерную структуру системы 3100 доставки контента, которая реализует услугу доставки контента; а также

Фиг. 14 - блок-схема, показывающая структуру примера терминального устройства.

45 Далее одинаковые ссылочные позиции относятся к одинаковым или, по меньшей мере, функционально эквивалентным признакам, если явно не указано иное.

#### ПОДРОБНОЕ ОПИСАНИЕ ВАРИАНТОВ ОСУЩЕСТВЛЕНИЯ

В нижеследующем описании делается ссылка на сопроводительные фигуры, которые составляют часть раскрытия и показывают в качестве иллюстрации конкретные аспекты

вариантов осуществления изобретения или конкретные аспекты, в которых могут использоваться варианты осуществления настоящего изобретения. Понятно, что варианты осуществления изобретения могут использоваться в других аспектах и содержать структурные или логические изменения, не изображенные на фигурах.

5 Поэтому нижеследующее подробное описание не следует воспринимать в ограничивающем смысле, и объем настоящего изобретения определяется прилагаемой формулой изобретения.

Например, понятно, что раскрытие в связи с описанным способом также может быть верным для соответствующего устройства или системы, сконфигурированной для  
10 выполнения способа, и наоборот. Например, если описан один или множество конкретных этапов способа, соответствующее устройство может включать в себя один или множество блоков, например функциональных блоков, для выполнения описанного одного или множества этапов способа (например, один блок, выполняющий один или множество этапов или множество блоков, каждый из которых выполняет один или  
15 несколько из множества этапов), даже если такой один или несколько блоков явно не описаны или не проиллюстрированы на фигурах. С другой стороны, например, если конкретное устройство описано на основе одного или множества блоков, например функциональных блоков, соответствующий способ может включать в себя один этап для выполнения функциональных возможностей одного или множества блоков  
20 (например, один этап, выполняющий функциональные возможности одного или множества блоков или множество этапов, каждый из которых выполняет функциональные возможности одного или нескольких из множества блоков), даже если такой один или множество этапов явно не описаны или не проиллюстрированы на фигурах. Кроме того, понятно, что признаки различных примерных вариантов  
25 осуществления и/или аспектов, описанных в данном документе, могут быть объединены друг с другом, если конкретно не указано иное.

Кодирование видео обычно относится к обработке последовательности изображений, которые образуют видео или видеопоследовательность. Вместо термина «изображение» можно использовать термин «кадр» или «картинка» как синонимы в области  
30 кодирования видео. Кодирование видео (или кодирование в целом) состоит из двух частей: кодирования видео и декодирования видео. Кодирование видео выполняется на стороне источника, обычно содержит обработку (например, путем сжатия) исходных видеоизображений для уменьшения объема данных, необходимых для представления видеоизображений (для более эффективного хранения и/или передачи). Декодирование  
35 видео выполняется на стороне назначения (целевой стороне) и обычно содержит обратную обработку по сравнению с кодером для восстановления видеоизображений. Варианты осуществления, относящиеся к «кодированию» видеоизображений (или изображений в целом), следует понимать как относящиеся к «кодированию» или «декодированию» видеоизображений или соответствующих видеопоследовательностей.  
40 Комбинация части кодирования и части декодирования также называется CODEC (кодирование и декодирование).

В случае кодирования видео без потерь исходные видеоизображения могут быть восстановлены, т. е. восстановленные видеоизображения имеют то же качество, что и исходные видеоизображения (при условии отсутствия потерь передачи или других  
45 потерь данных во время хранения или передачи). В случае кодирования видео с потерями выполняется дополнительное сжатие, например, путем квантования, чтобы уменьшить объем данных, представляющих видеоизображения, которые не могут быть полностью восстановлены в декодере, т.е. качество восстановленных видеоизображений ниже или

хуже по сравнению с качеством исходных видеоизображений.

Несколько стандартов кодирования видео принадлежат к группе «гибридных видеокодеков с потерями» (т. е. сочетают пространственное и временное предсказание в области выборки и кодирование с двумерным преобразованием для применения квантования в области преобразования). Каждое изображение видеопоследовательности обычно разделяется на набор неперекрывающихся блоков, и кодирование обычно выполняется на уровне блоков. Другими словами, в кодере видео обычно обрабатывается, то есть кодируется на уровне блока (видеоблока), например, с использованием пространственного (внутри изображения) предсказания и/или временного (внешнего к изображению) предсказания для генерации блока предсказания, вычитая блок предсказания из текущего блока (блока, который в настоящее время обрабатывается/должен быть обработан) для получения остаточного блока, преобразования остаточного блока и квантования остаточного блока в области преобразования, чтобы уменьшить объем данных, которые должны быть переданы (сжатие), тогда как в декодере обратная обработка по сравнению с кодером применяется к закодированному или сжатому блоку, чтобы восстановить текущий блок для представления. Кроме того, кодер дублирует цикл обработки декодера, так что оба будут генерировать идентичные предсказания (например, внутреннее и внешнее предсказания) и/или восстановления для обработки, то есть кодирование последующих блоков.

В следующих вариантах осуществления система 10 кодирования видео, видеокодер 20 и видеодекодер 30 описаны на основе фиг. с 1 по 3.

Фиг. 1А является схематической блок-схемой, иллюстрирующей примерную систему 10 кодирования, например, систему 10 кодирования видео (или систему 10 короткого кодирования), которая может использовать методики этого настоящей заявки. Видеокодер 20 (или короткий кодер 20) и видеодекодер 30 (или короткий декодер 30) системы 10 кодирования видео представляют примеры устройств, которые могут быть сконфигурированы для выполнения методик в соответствии с различными примерами, описанными в настоящей заявке.

Как показано на фиг. 1А, система 10 кодирования содержит исходное устройство 12, сконфигурированное для предоставления данных 21 закодированного изображения, например, целевому устройству 14 для декодирования данных 21 закодированного изображения.

Исходное устройство 12 содержит кодер 20 и может дополнительно, т.е. необязательно, содержать источник 16 изображения, препроцессор (или блок предварительной обработки) 18, например, препроцессор 18 изображения, и интерфейс связи или блок 22 связи.

Источник 16 изображения может содержать или быть любым типом устройства захвата изображения, например камерой для захвата реального изображения, и/или устройством генерации изображения любого типа, например процессором компьютерной графики для создания изображения компьютерной анимации или любое другим устройством для получения и/или предоставления реального изображения любого типа, изображения, сгенерированного компьютером (например, содержимого экрана, изображения виртуальной реальности (VR)) и/или любой их комбинации (например, изображения дополненной реальности (AR)). Источником снимка может быть любой тип памяти или хранилища, в котором хранятся любые из вышеупомянутых изображений.

В отличие от препроцессора 18 и обработки, выполняемой блоком 18

предварительной обработки, изображение или данные 17 изображения также могут называться необработанным изображением или данными 17 необработанного изображения.

5 Препроцессор 18 выполнен с возможностью приема (необработанных) данных 17 изображения и выполнения предварительной обработки данных 17 изображения для получения предварительно обработанного изображения 19 или предварительно обработанных данных 19 изображения. Предварительная обработка, выполняемая препроцессором 18, может, например, содержать обрезку, преобразование цветового формата (например, из RGB в YCbCr), цветокоррекцию или уменьшение шума. Можно  
10 понять, что блок 18 предварительной обработки может быть необязательным компонентом.

Видеокодер 20 сконфигурирован для приема предварительно обработанных данных 19 изображения и предоставления данных 21 закодированного изображения (дополнительные подробности будут описаны ниже, например, на основе фиг. 2).  
15 Интерфейс 22 связи исходного устройства 12 может быть сконфигурирован для приема данных 21 закодированного изображения и передачи данных 21 закодированного изображения (или любой их дополнительной обработанной версии) по каналу 13 связи на другое устройство, например, на целевое устройство 14 или любое другое устройство, для хранения или прямой восстановления (восстановления).

20 Целевое устройство 14 содержит декодер 30 (например, видеodeкодер 30) и может дополнительно, т.е. необязательно, содержать интерфейс связи или блок 28 связи, постпроцессор 32 (или блок 32 постобработки) и устройство 34 отображения.

Интерфейс 28 связи целевого устройства 14 сконфигурирован для приема закодированных данных 21 изображения (или любой их дополнительной обработанной  
25 версии), например, непосредственно от исходного устройства 12 или из любого другого источника, например, устройства хранения, например, устройства хранения данных закодированного изображения, и предоставления данных 21 закодированного изображения в декодер 30.

Интерфейс 22 связи и интерфейс 28 связи могут быть выполнены с возможностью  
30 передачи или приема данных 21 закодированного изображения или закодированных данных 13 через прямую линию связи между исходным устройством 12 и целевым устройством 14, например прямое проводное или беспроводное соединение, или через сеть любого типа, например проводную или беспроводную сеть или любое их сочетание, или любую частную и общедоступную сеть, или любое их сочетание.

35 Интерфейс 22 связи может быть, например, выполнен с возможностью упаковки данных 21 закодированного изображения в соответствующий формат, например, в пакеты, и/или обработки данных закодированного изображения с использованием любого типа кодирования передачи или обработки для передачи по линии связи или сети связи.

40 Интерфейс 28 связи, образующий аналог интерфейса 22 связи, может быть, например, сконфигурирован для приема переданных данных и обработки данных передачи с использованием любого вида соответствующего декодирования или обработки передачи и/или распаковки для получения данных 21 закодированного изображения.

И интерфейс 22 связи, и интерфейс 28 связи могут быть сконфигурированы как  
45 однонаправленные интерфейсы связи, как показано стрелкой для канала 13 связи на фиг. 1А, указывающей от исходного устройства 12 к целевому устройству 14, или как интерфейсы двунаправленной связи, и могут быть сконфигурированы, например, для отправки и приема сообщений, например, для установки соединения, для подтверждения

и обмена любой другой информацией, относящейся к линии связи и/или передаче данных, например, к передаче данных закодированного изображения.

Декодер 30 выполнен с возможностью приема данных 21 закодированного изображения и предоставления данных 31 декодированного изображения или декодированного изображения 31 (дополнительные подробности будут описаны ниже, например, на основе Фиг.3 или Фиг.5.

Постпроцессор 32 целевого устройства 14 сконфигурирован для постобработки данных 31 декодированного изображения (также называемых данными восстановленного изображения), например, декодированного изображения 31, для получения данных 33 постобработанного изображения, например, постобработанного изображения 33. Постобработка, выполняемая блоком 32 постобработки, может содержать, например, преобразование цветового формата (например, из YCbCr в RGB), цветокоррекцию, обрезку или повторную выборку, или любую другую обработку, например, для подготовки данных 31 декодированного изображения для отображения, например, с помощью устройства 34 отображения.

Устройство 34 отображения целевого устройства 14 выполнено с возможностью приема данных 33 постобработанного изображения для отображения изображения, например, пользователю или зрителю. Устройство 34 отображения может быть или содержать любой вид дисплея для представления восстановленного изображения, например, интегрированный или внешний дисплей или монитор. Дисплеи могут, например, содержать жидкокристаллические дисплеи (LCD), дисплеи на органических светодиодах (OLED), плазменные дисплеи, проекторы, микро-светодиодные дисплеи, жидкие кристаллы на кремнии (LCoS), цифровой световой процессор (DLP) или любые другие дисплеи.

Хотя фиг. 1А изображает исходное устройство 12 и целевое устройство 14 как отдельные устройства, варианты осуществления устройств могут также содержать оба или обе функциональные возможности, исходное устройство 12 или соответствующие функциональные возможности и целевое устройство 14 или соответствующие функциональные возможности. В таких вариантах осуществления исходное устройство 12 или соответствующие функциональные возможности и целевое устройство 14 или соответствующие функциональные возможности могут быть реализованы с использованием одного и того же аппаратного и/или программного обеспечения, либо с помощью отдельных аппаратных средств и/или программного обеспечения, либо любой их комбинации.

Как будет очевидно для специалиста на основании описания, наличие и (точное) разделение функциональных возможностей различных блоков или функциональных возможностей в исходном устройстве 12 и/или целевом устройстве 14, как показано на Фиг.1А, может варьироваться в зависимости от фактического устройства и применения.

Кодер 20 (например, видеокодер 20) или декодер 30 (например, видеодекодер 30) или икодер 20, и декодер 30 могут быть реализованы через схему обработки, как показано на фиг. 1В, например, один или несколько микропроцессоров, цифровые сигнальные процессоры (DSP), специализированные интегральные схемы (ASIC), программируемые вентильные матрицы (FPGA), дискретную логику, аппаратное обеспечение, выделенное кодирование видео или любые их комбинации. Кодер 20 может быть реализован через схему 46 обработки для воплощения различных модулей, как обсуждалось в отношении кодера 20 на фиг. 2 и/или любой другой системы или подсистемы, описанной в данном документе. Декодер 30 может быть реализован через схему 46 обработки для воплощения различных модулей, как обсуждалось в отношении декодера 30 на фиг. 3 и/или любой

другой системы или подсистемы декодера, описанной в данном документе. Схема обработки может быть сконфигурирована для выполнения различных операций, как описано ниже. Как показано на Фиг.5, если методики частично реализованы в программном обеспечении, устройство может хранить инструкции для программного обеспечения на подходящем долговременном машиночитаемом носителе данных и может выполнять инструкции в аппаратных средствах, используя один или более процессоров для выполнения методик этого раскрытия. Любой из видеокодера 20 и видеодекодера 30 может быть интегрирован как часть объединенного кодера/декодера (CODEC) в одном устройстве, например, как показано на Фиг.1В.

Исходное устройство 12 и целевое устройство 14 могут содержать любое из широкого диапазона устройств, в том числе любые виды портативных или стационарных устройств, например ноутбуки или портативные компьютеры, мобильные телефоны, смартфоны, планшеты или планшетные компьютеры, камеры, настольные компьютеры, наборы-приставки, телевизоры, устройства отображения, цифровые медиаплееры, игровые приставки, устройства потоковой передачи видео (например, серверы служб контента или серверы доставки контента), устройство приемника вещания, устройство передатчика вещания и т.п. и может использовать никакой или любой тип операционной системы. В некоторых случаях исходное устройство 12 и целевое устройство 14 могут быть оборудованы для беспроводной связи. Таким образом, исходное устройство 12 и целевое устройство 14 могут быть устройствами беспроводной связи.

В некоторых случаях система 10 кодирования видео, проиллюстрированная на фиг. 1А, является просто примером, и способы настоящей заявки могут применяться к настройкам кодирования видео (например, кодированию видео или декодированию видео), которые не обязательно включают в себя передачу данных между устройствами кодирования и декодирования. В других примерах данные извлекаются из локальной памяти, передаются по сети и т.п. Устройство кодирования видео может кодировать и сохранять данные в памяти, и/или устройство декодирования видео может извлекать и декодировать данные из памяти. В некоторых примерах кодирование и декодирование выполняется устройствами, которые не обмениваются данными друг с другом, а просто кодируют данные в память и/или извлекают и декодируют данные из памяти.

Для удобства описания варианты осуществления изобретения описаны здесь, например, со ссылкой на высокоэффективное кодирование видео (HEVC) или на эталонное программное обеспечение универсального кодирования видео (VVC), стандарт кодирования видео следующего поколения, разработанный Группой сотрудничества по кодированию видео (JCT-VC) Группы экспертов ITU-T по кодированию видео (VCEG) и Группы экспертов ISO/IEC по движущимся изображениям (MPEG). Специалист в данной области техники поймет, что варианты осуществления изобретения не ограничиваются HEVC или VVC.

Кодер и способ кодирования

Фиг. 2 показывает схематическую блок-схему примерного видеокодера 20, который сконфигурирован для реализации способов настоящей заявки. В примере на фиг.2 видеокодер 20 содержит вход 201 (или входной интерфейс 201), блок 204 вычисления остатка, блок 206 обработки преобразования, блок 208 квантования, блок 210 обратного квантования и блок 212 обработки обратного преобразования, блок 214 восстановления, блок 220 контурного фильтра, буфер декодированных изображений (DPB) 230, блок 260 выбора режима, блок 270 энтропийного кодирования и выход 272 (или выходной интерфейс 272). Блок 260 выбора режима может включать в себя блок 244 внешнего предсказания, блок 254 внутреннего предсказания и блок 262 разделения. Блок 244

внешнего предсказания может включать в себя блок оценки движения и блок компенсации движения (не показаны). Видеокодер 20, показанный на фиг.2, также может называться гибридным видеокодером или видеокодером согласно гибридному видеокодеку.

5 Блок 204 вычисления остатка, блок 206 обработки преобразования, блок 208 квантования, блок 260 выбора режима могут упоминаться как формирующие прямой путь сигнала кодера 20, тогда как блок 210 обратного квантования, блок 212 обработки обратного преобразования, блок 214 восстановления, буфер 216, контурный фильтр 220, буфер 230 декодированного изображения (DPB), блок 244 внешнего предсказания  
10 и блок 254 внутреннего предсказания могут называться формирующими обратный путь сигнала видеокодера 20, при этом обратный путь сигнала видеокодера 20 соответствует пути сигнала декодера (см. видеокодер 30 на фиг. 3). Блок 210 обратного квантования, блок 212 обработки обратного преобразования, блок 214 восстановления, контурный фильтр 220, буфер 230 декодированного изображения (DPB), блок 244  
15 внешнего предсказания и блок 254 внутреннего предсказания также относятся к формированию «встроенного декодера» видеокодера 20.

Изображения и разделение изображений (изображения и блоки)

Кодер 20 может быть сконфигурирован для приема, например, через вход 201 изображения 17 (или данных 17 изображения), например, изображения  
20 последовательности изображений, образующих видео или видеопоследовательность. Принятое изображение или данные изображения также могут быть предварительно обработанным изображением 19 (или предварительно обработанными данными 19 изображения). Для простоты следующее описание относится к изображению 17. Изображение 17 также может называться текущим изображением или изображением,  
25 которое должно быть кодировано (в частности, при кодировании видео, чтобы отличать текущее изображение от других изображений, например, ранее кодированных и/или декодированных изображений той же видеопоследовательности, то есть видеопоследовательности, которая также содержит текущее изображение).

(Цифровое) изображение представляет собой или может рассматриваться как  
30 двумерный массив или матрица выборок со значениями интенсивности. Выборка в массиве также может называться пикселем (сокращенная форма элемента изображения) или пикселем. Количество выборок в горизонтальном и вертикальном направлении (или по оси) массива или изображения определяет размер и/или разрешение изображения. Для представления цвета обычно используются три цветовых компонента, то есть  
35 изображение может быть представлено или включать в себя три массива выборок. В формате RGB или цветовом пространстве изображение содержит соответствующий массив выборок красного, зеленого и синего цветов. Однако при кодировании видео каждый пиксель обычно представлен в формате яркости и цветности или цветовом пространстве, например, YCbCr, который содержит компонент яркости, обозначенный  
40 Y (иногда также вместо этого используется L), и два компонента цветности, обозначенные Cb и Cr. Компонент Y яркости (или сокращенно яркости) представляет яркость или интенсивность уровня серого (например, как в полутоновом изображении), в то время как два компонента цветности (или сокращенно цветности) Cb и Cr представляют компоненты цветности или информации о цвете. Соответственно,  
45 изображение в формате YCbCr содержит массив выборок яркости значений (Y) выборок яркости и два массива выборок цветности значений (Cb и Cr) цветности. Изображения в формате RGB могут быть конвертированы или преобразованы в формат YCbCr, и наоборот, этот процесс также известен как преобразование цвета или конверсия. Если

изображение является монохромным, оно может содержать только массив выборок яркости. Соответственно, изображение может быть, например, массивом выборок яркости в монохромном формате или массивом выборок яркости и двумя соответствующими массивами выборок цветности в цветовом формате 4:2:0, 4:2:2 и 4:4.

Варианты осуществления видеокодера 20 могут содержать блок разделения изображения (не показан на фиг.2), сконфигурированный для разделения изображения 17 на множество (обычно не перекрывающихся) блоков 203 изображения. Эти блоки также могут называться корневыми блоками, макроблоками (H.264/AVC) или блоками дерева кодирования (СТВ) или единицами дерева кодирования (СТУ) (H.265/HEVC и VVC). Блок разделения изображений может быть сконфигурирован для использования одного и того же размера блока для всех изображений видеопоследовательности и соответствующей сетки, определяющей размер блока, или для изменения размера блока между изображениями, или подмножествами, или группами изображений, и разделением каждого изображения на соответствующие блоки.

В дополнительных вариантах осуществления видеокодер может быть сконфигурирован для приема непосредственно блока 203 изображения 17, например, одного, нескольких или всех блоков, образующих изображение 17. Блок 203 изображения может также называться блоком текущего изображения или блоком изображения, который должен быть кодирован.

Подобно изображению 17, блок 203 изображения снова является или может рассматриваться как двумерный массив или матрица выборок со значениями интенсивности (значениями выборок), хотя и меньшего размера, чем изображение 17. Другими словами, блок 203 может содержать, например, один массив выборок (например, массив яркости в случае монохромного изображения 17 или массив яркости или цветности в случае цветного изображения) или три массива выборок (например, яркость и два массива цветности в случае цветного изображения 17) или любое другое количество и/или тип массивов в зависимости от применяемого цветового формата. Количество выборок в горизонтальном и вертикальном направлении (или по оси) блока 203 определяет размер блока 203. Соответственно, блок может быть, например, массивом дискретных выборок  $M \times N$  ( $M$ -столбец на  $N$ -строка) или массивом  $M \times N$  коэффициентов преобразования.

Варианты осуществления видеокодера 20, как показано на фиг. 2, могут быть сконфигурированы для кодирования изображения 17 блок за блоком, например, кодирование и предсказание выполняются для каждого блока 203.

#### Вычисление остатка

Блок 204 вычисления остатка может быть сконфигурирован для вычисления остаточного блока 205 (также называемого остатком 205) на основе блока 203 изображения и блока 265 предсказания (дополнительные подробности о блоке 265 предсказания предоставлены позже), например, путем вычитания значений выборки блока 265 предсказания из значений выборки блока 203 изображения, выборка за выборкой (пиксель за пикселем), чтобы получить остаточный блок 205 в области выборки.

#### Преобразование

Блок 206 обработки преобразования может быть сконфигурирован для применения преобразования, например, дискретного косинусного преобразования (DCT) или дискретного синусоидального преобразования (DST), к значениям выборки остаточного блока 205 для получения коэффициентов 207 преобразования в области преобразования.

Коэффициенты 207 преобразования также могут называться остаточными коэффициентами преобразования и представлять остаточный блок 205 в области преобразования.

Блок 206 обработки преобразования может быть сконфигурирован для применения целочисленных приближений DCT/DST, таких как преобразования, указанные для H.265/HEVC. По сравнению с ортогональным преобразованием DCT, такие целочисленные приближения обычно масштабируются с некоторым коэффициентом. Чтобы сохранить норму остаточного блока, который обрабатывается прямым и обратным преобразованиями, в процессе преобразования применяются дополнительные коэффициенты масштабирования. Коэффициенты масштабирования обычно выбираются на основе некоторых ограничений, таких как коэффициенты масштабирования, равные степени двойки для операций сдвига, разрядность коэффициентов преобразования, компромисс между точностью и затратами на реализацию и т. д. Конкретные коэффициенты масштабирования, например, задаются для обратного преобразования, например, блоком 212 обработки обратного преобразования (и соответствующего обратного преобразования, например, блоком 312 обработки обратного преобразования в видеодекодере 30) и соответствующими коэффициентами масштабирования для прямого преобразования, например, блоком 206 обработки преобразования, могут быть заданы в кодере 20 соответственно.

Варианты осуществления видеокодера 20 (соответственно модуль 206 обработки преобразования) могут быть выполнены с возможностью вывода параметров преобразования, например, типа преобразования или преобразований, например, непосредственно или кодированы или сжаты через блок 270 энтропийного кодирования, так что, например, видеодекодер 30 может принимать и использовать параметры преобразования для декодирования.

#### Квантование

Блок 208 квантования может быть сконфигурирован для квантования коэффициентов 207 преобразования для получения квантованных коэффициентов 209, например, путем применения скалярного квантования или векторного квантования. Квантованные коэффициенты 209 также могут упоминаться как квантованные коэффициенты 209 преобразования или квантованные остаточные коэффициенты 209.

Процесс квантования может уменьшить битовую глубину, ассоциированную с некоторыми или всеми из коэффициентов 207 преобразования. Например,  $n$ -битовый коэффициент преобразования может быть округлен до  $m$ -битного коэффициента преобразования во время квантования, где  $n$  больше  $m$ . Степень квантования может быть изменена посредством регулировки параметра квантования (QP). Например, для скалярного квантования может применяться другое масштабирование для достижения более точного или более грубого квантования. Меньшие размеры шагов квантования соответствуют более точному квантованию, тогда как большие размеры шагов квантования соответствуют более грубому квантованию. Применимый размер шага квантования может быть указан параметром квантования (QP). Параметр квантования может, например, быть индексом для заранее определенного набора применимых размеров шага квантования. Например, малые параметры квантования могут соответствовать точному квантованию (малым размерам шагов квантования), а большие параметры квантования могут соответствовать грубому квантованию (большим размерам шагов квантования) или наоборот. Квантование может включать в себя деление на размер шага квантования, и соответствующее и/или обратное деквантование, например, посредством блока 210 обратного квантования, может включать в себя

умножение на размер шага квантования. Варианты осуществления согласно некоторым стандартам, например HEVC, могут быть сконфигурированы для использования параметра квантования для определения размера шага квантования. Обычно размер шага квантования может быть вычислен на основе параметра квантования с использованием приближения с фиксированной точкой уравнения, включающего в себя деление. Дополнительные коэффициенты масштабирования могут быть введены для квантования и деквантования, чтобы восстановить норму остаточного блока, который может быть изменен из-за масштабирования, используемого в приближении с фиксированной точкой уравнения для размера шага квантования и параметра квантования. В одной примерной реализации масштабирование обратного преобразования и деквантование могут быть скомбинированы. В качестве альтернативы могут использоваться настроенные таблицы квантования и сигнализироваться от кодера к декодеру, например, в потоке битов. Квантование - это операция с потерями, при которой потери возрастают с увеличением размеров шагов квантования.

Варианты осуществления видеокодера 20 (соответственно модуля 208 квантования) могут быть выполнены с возможностью вывода параметров квантования (QP), например, напрямую или кодированных через модуль 270 энтропийного кодирования, так что, например, видеодекодер 30 может принимать и применять параметры квантования для декодирования.

Обратное квантование

Блок 210 обратного квантования сконфигурирован для применения обратного квантования блока 208 квантования к квантованным коэффициентам для получения деквантованных коэффициентов 211, например, путем применения обратной схемы квантования, применяемой блоком 208 квантования, на основе или с использованием того же шага квантования в качестве блока 208 квантования. Деквантованные коэффициенты 211 также могут упоминаться как деквантованные остаточные коэффициенты 211 и соответствовать, хотя обычно не идентичны коэффициентам преобразования из-за потери при квантовании, коэффициентам 207 преобразования.

Обратное преобразование

Блок 212 обработки обратного преобразования сконфигурирован для применения обратного преобразования для преобразования, применяемого блоком 206 обработки преобразования, например, обратного дискретного косинусного преобразования (DCT) или обратного дискретного синусоидального преобразования (DST) или других обратных преобразований, для получения восстановленного остаточного блока 213 (или соответствующих деквантованных коэффициентов 213) в области выборки. Восстановленный остаточный блок 213 также может называться блоком 213 преобразования.

Восстановление

Блок 214 восстановления (например, сумматор или сумматор 214) сконфигурирован для добавления блока 213 преобразования (т.е. восстановленного остаточного блока 213) к блоку 265 предсказания, чтобы получить восстановленный блок 215 в области выборки, например, путем сложения, выборка за выборкой, выборочных значений восстановленного остаточного блока 213 и выборочных значений блока 265 предсказания.

Фильтрация

Блок 220 контурного фильтра (или сокращенно «контурный фильтр» 220) сконфигурирован для фильтрации восстановленного блока 215 для получения отфильтрованного блока 221 или, в общем, для фильтрации восстановленных выборок

для получения отфильтрованных выборок. Блок контурного фильтра, например, сконфигурирован для сглаживания переходов пикселей или иного улучшения качества видео. Блок 220 контурного фильтра может содержать один или несколько контурных фильтров, таких как деблочный фильтр, фильтр с адаптивным смещением (SAO) или один или несколько других фильтров, например, двусторонний фильтр, адаптивный контурный фильтр (ALF), повышения резкости, сглаживающие фильтры или коллаборативные фильтры, или любая их комбинация. Хотя блок 220 контурного фильтра показан на Фиг.2 как контурный фильтр, в других конфигурациях модуль 220 контурного фильтра может быть реализован как постконтурный фильтр.

Отфильтрованный блок 221 также может упоминаться как фильтрованный восстановленный блок 221.

Варианты осуществления видеокодера 20 (соответственно блока 220 контурного фильтра) могут быть сконфигурированы для вывода параметров контурного фильтра (таких как информация об адаптивном смещении выборки), например, непосредственно или закодированы через блок 270 энтропийного кодирования, так что, например, декодер 30 может принимать и применять одинаковые параметры контурного фильтра или соответствующие контурные фильтры для декодирования.

#### Буфер декодированного изображения

Буфер 230 декодированных изображений (DPB) может быть памятью, в которой хранятся опорные изображения или, в общем, данные опорных изображений для кодирования видеоданных с помощью видеокодера 20. DPB 230 может быть сформирован любым из множества запоминающих устройств, таких как динамическая память с произвольным доступом (DRAM), в том числе синхронная DRAM (SDRAM), магниторезистивная RAM (MRAM), резистивная RAM (RRAM) или другие типы запоминающих устройств. Буфер 230 декодированных изображений (DPB) может быть сконфигурирован для хранения одного или нескольких фильтрованных блоков 221. Буфер 230 декодированных изображений может быть дополнительно сконфигурирован для хранения других ранее отфильтрованных блоков, например, ранее восстановленных и отфильтрованных блоков 221, одного и того же текущего изображения или разных изображений, например, ранее восстановленных изображений, и может предоставлять полные ранее восстановленные, то есть декодированные, изображения (и соответствующие опорные блоки и выборки) и/или частично восстановленное текущее изображение (и соответствующие опорные блоки и выборки), например, для внешнего предсказания. Буфер декодированных изображений (DPB) 230 также может быть сконфигурирован для хранения одного или нескольких нефильтрованных восстановленных блоков 215 или, в общем, нефильтрованных восстановленных выборок, например, если восстановленный блок 215 не фильтруется блоком 220 контурной фильтрации, или любой другой дополнительно обработанной версии восстановленных блоков или выборок.

#### Выбор режима (разделение и предсказание)

Блок 260 выбора режима содержит блок 262 разделения, блок 244 внешнего предсказания и блок 254 внутреннего предсказания и сконфигурирован для приема или получения данных исходного изображения, например исходного блока 203 (текущий блок 203 текущего изображения 17), и данные восстановленного изображения, например отфильтрованные и/или нефильтрованные восстановленные выборки или блоки одного и того же (текущего) изображения и/или из одного или множества ранее декодированных изображений, например, из буфера 230 декодированных изображений или других буферов (например, строкового буфера, не показаны). Данные восстановленного

изображения используются в качестве данных опорного изображения для предсказания, например, внешнего предсказания или внутреннего предсказания, чтобы получить блок 265 предсказания или предсказатель 265.

5 Модуль 260 выбора режима может быть выполнен с возможностью определения или выбора разделения для текущего режима предсказания блока (в том числе отсутствие разделения) и режима предсказания (например, режима внутреннего или внешнего предсказания) и генерации соответствующего блока 265 предсказания, который используется для вычисления остаточного блока 205 и для восстановления восстановленного блока 215.

10 Варианты осуществления модуля 260 выбора режима могут быть сконфигурированы для выбора разделения и режима предсказания (например, из тех, которые поддерживаются или доступны для модуля 260 выбора режима), которые обеспечивают наилучшее совпадение или, другими словами, минимальный остаток (минимальный остаток означает лучшее сжатие для передачи или хранения), или минимальные  
15 накладные расходы на сигнализацию (минимальные накладные расходы на сигнализацию означают лучшее сжатие для передачи или хранения), или который учитывает или уравнивает оба. Блок 260 выбора режима может быть сконфигурирован для определения режима разделения и предсказания на основе оптимизации искажения скорости (RDO), то есть выбора режима предсказания, который  
20 обеспечивает минимальное искажение скорости. Такие термины, как «лучший», «минимальный», «оптимальный» и т. д. в этом контексте не обязательно относятся к общему «лучшему», «минимальному», «оптимальному» и т. д., но также могут относиться к выполнению прекращения или критерию выбора, такому как значение, превышающее или падающее ниже порогового значения, или другим ограничениям,  
25 потенциально ведущим к «суб-оптимальному выбору», но уменьшающие сложность и время обработки.

Другими словами, блок 262 разделения может быть сконфигурирован для разделения блока 203 на более мелкие разделы или подблоки (которые снова образуют блоки), например, итеративно с использованием четырехугольного разделения (QT), двоичного  
30 разделения (BT) или разделения на тройное дерево (TT) или любой их комбинации, и для выполнения, например, предсказания для каждого из разделов или подблоков блока, при этом выбор режима содержит выбор древовидной структуры разделенного блока 203, и режимы предсказания применяются к каждому из разделов или подблоков блока.

35 Далее более подробно поясняется разделение (например, посредством модуля 260 разделения) и обработка предсказания (посредством модуля 244 внешнего предсказания и модуля 254 внутреннего предсказания), выполняемые примерным видеокодером 20.

#### Разделение

Блок 262 разделения может разделять (или разбивать) текущий блок 203 на меньшие  
40 части, например, меньшие блоки квадратного или прямоугольного размера. Эти меньшие блоки (которые также могут называться субблоками) могут быть дополнительно разделены на еще меньшие разделы. Это также называется разделением дерева или иерархическим разделением дерева, в котором корневой блок, например, на уровне 0 корневого дерева (уровень 0 иерархии, глубина 0), может быть рекурсивно  
45 разделен, например разделен на два или более блоков следующего более низкого уровня дерева, например, узлы на уровне 1 дерева (уровень 1 иерархии, глубина 1), при этом эти блоки могут быть снова разделены на два или более блоков следующего более низкого уровня, например, уровень 2 дерева (уровень 2 иерархии, глубина 2) и т. д. до

тех пор, пока разделение не будет завершено, например, потому что критерий завершения выполнен, например, достигнута максимальная глубина дерева или минимальный размер блока. Блоки, которые не разделяются далее, также называются листовыми блоками или листовыми узлами дерева. Дерево, использующее разделение на два раздела, называется двоичным деревом (BT), дерево, использующее разделение на три раздела, называется тройным деревом (TT), а дерево, использующее разделение на четыре раздела, называется деревом квадратов (QT).

Как упоминалось ранее, используемый здесь термин «блок» может быть частью, в частности квадратной или прямоугольной частью изображения. Что касается, например, HEVC и VVC, блок может быть или соответствовать единице дерева кодирования (CTU), единице кодирования (CU), единице предсказания (PU) и единице преобразования (TU) и/или соответствующим блокам, например блоку дерева кодирования (CTB), блоку кодирования (CB), блоку преобразования (TB) или блоку предсказания (PB).

Например, единица дерева кодирования (CTU) может быть или содержать СТВ выборки яркости, два соответствующих СТВ выборки цветности изображения, которое имеет три массива выборок, или СТВ выборки монохромного изображения или изображения, которое кодируется с использованием трех отдельных цветовых плоскостей и синтаксических структур, используемых для кодирования выборок. Соответственно, блок дерева кодирования (CTB) может быть блоком NxN выборок для некоторого значения N, так что деление компонента на СТВ является разделением. Единица кодирования (CU) может быть или содержать блок кодирования выборок яркости, два соответствующих блока кодирования выборок цветности изображения, которое имеет три массива выборок, или блок кодирования выборок монохромного изображения или изображения, которое кодируется с использованием трех отдельных цветовых плоскостей и синтаксических структур, используемых для кодирования выборок. Соответственно, блок кодирования (CB) может быть блоком MxN выборок для некоторых значений M и N, так что деление СТВ на блоки кодирования является разделением.

В вариантах осуществления, например, согласно HEVC, единица дерева кодирования (CTU) может быть разделена на CU с использованием структуры дерева квадратов, обозначенной как дерево кодирования. Решение о том, кодировать ли область изображения с использованием внешнего (временного) или внутреннего (пространственного) предсказания, принимается на уровне CU. Каждая CU может быть дополнительно разделена на одну, две или четыре PU в соответствии с типом разделения PU. Внутри одной PU применяется тот же процесс предсказания, и соответствующая информация передается в декодер на основе PU. После получения остаточного блока путем применения процесса предсказания на основе типа разделения PU, CU может быть разделена на единицы преобразования (TU) в соответствии с другой структурой квадродерева, аналогичной дереву кодирования для CU.

В вариантах осуществления, например, согласно последнему разрабатываемому в настоящее время стандарту кодирования видео, который называется универсальным кодированием видео (VVC), для разделения блока кодирования используется разделение на квадраты и двоичное дерево (QTBT). В блочной структуре QTBT CU может иметь квадратную или прямоугольную форму. Например, единица дерева кодирования (CTU) сначала разделяется структурой дерева квадратов. Листовые узлы квадродерева дополнительно разделяются двоичным деревом или тройной (или тройной) древовидной структурой. Листовые узлы дерева разделения называются единицами кодирования (CU), и эта сегментация используется для обработки предсказания и преобразования

без какого-либо дальнейшего разделения. Это означает, что CU, PU и TU имеют одинаковый размер блока в структуре блока кодирования QTBT. Параллельно с блочной структурой QTBT было предложено использовать множественное разделение, например, тройное разделение дерева.

5 В одном примере блок 260 выбора режима видеокодера 20 может быть выполнен с возможностью выполнения любого сочетания способов разделения, описанных в данном документе.

Как описано выше, видеокодер 20 сконфигурирован для определения или выбора наилучшего или оптимального режима предсказания из набора (заранее определенных) режимов предсказания. Набор режимов предсказания может содержать, например, режимы внутреннего предсказания и/или режимы внешнего предсказания.

#### Внутреннее предсказание

10 Набор режимов внутреннего предсказания может содержать 35 различных режимов внутреннего предсказания, например, ненаправленные режимы, такие как режим DC (или средний) и планарный режим, или направленные режимы, например, как определено в HEVC, или может содержать 67 различных режимов внутреннего предсказания, например, ненаправленные режимы, такие как режим DC (или средний) и планарный режим, или направленные режимы, например, как определено для VVC.

15 Блок 254 внутреннего предсказания выполнен с возможностью использования восстановленных выборок соседних блоков одного и того же текущего изображения для генерации блока 265 внутреннего предсказания согласно режиму внутреннего предсказания набора режимов внутреннего предсказания.

20 Блок 254 внутреннего предсказания (или, в общем, блок 260 выбора режима) дополнительно выполнен с возможностью вывода параметров внутреннего предсказания (или, в общем, информации, указывающей выбранный режим внутреннего предсказания для блока) в блок 270 энтропийного кодирования в форме элементов 266 синтаксиса для включения в данные 21 закодированного изображения, чтобы, например, видеodeкодер 30 мог принимать и использовать параметры предсказания для декодирования.

#### 30 Внешнее предсказание

Набор (или возможные) режимов внешнего предсказания зависит от доступных опорных изображений (т. е. предыдущих, по меньшей мере, частично декодированных изображений, например, сохраненных в DPB 230) и других параметров внешнего предсказания, например, от того, все ли опорное изображение или только его часть, например, область окна поиска вокруг области текущего блока опорного изображения используется для поиска наиболее подходящего опорного блока и/или, например, применяется ли интерполяция пикселей, например, интерполяция половины/полупикселя и/или четверти пикселя, или нет.

40 В дополнение к вышеупомянутым режимам предсказания могут применяться режим пропуска и/или прямой режим.

Блок 244 внешнего предсказания может включать в себя блок оценки движения (ME) и блок компенсации движения (MC) (оба не показаны на фиг.2). Блок оценки движения может быть сконфигурирован для приема или получения блока 203 изображения (блока 203 текущего изображения для текущего изображения 17) и декодированного изображения 231, или по меньшей мере одного или множества ранее восстановленных блоков, например, восстановленных блоков одного или множества других/различных ранее декодированных изображений 231 для оценки движения. Например, видеопоследовательность может содержать текущее изображение и ранее

декодированные изображения 231, или, другими словами, текущее изображение и ранее декодированные изображения 231 могут быть частью или формировать последовательность изображений, образующих видеопоследовательность.

5 Кодер 20 может, например, быть сконфигурирован для выбора опорного блока из множества опорных блоков одного и того же или разных изображений из множества других изображений и предоставления опорного изображения (или индекса опорного изображения) и/или смещения (пространственного смещение) между положением (координаты  $x$ ,  $y$ ) опорного блока и положением текущего блока в качестве параметров внешнего предсказания для блока оценки движения. Это смещение также называется

10 вектором движения (MV).

Блок компенсации движения сконфигурирован для получения, например, приема параметра внешнего предсказания и выполнения внешнего предсказания на основе или с использованием параметра внешнего предсказания для получения блока 265 внешнего предсказания. Компенсация движения, выполняемая блоком компенсации

15 движения, может включать в себя извлечение или генерацию блока предсказания на основе вектора движения/блока, определенного посредством оценки движения, возможно выполнение интерполяции с точностью до субпикселя. Интерполяционная фильтрация может генерировать дополнительные выборки пикселей из известных выборок пикселей, таким образом потенциально увеличивая количество кандидатных блоков предсказания,

20 которые могут использоваться для кодирования блока изображения. После приема вектора движения для PU текущего блока изображения блок компенсации движения может найти блок предсказания, на который указывает вектор движения, в одном из списков опорных изображений.

Блок компенсации движения также может генерировать элементы синтаксиса, связанные с блоками и слайсом видео, для использования видеodeкодером 30 при

25 декодировании блоков изображения слайса видео.

#### Энтропийное кодирование

Блок 270 энтропийного кодирования сконфигурирован для применения, например, алгоритма или схемы энтропийного кодирования (например, схемы кодирования с

30 переменной длиной (VLC), контекстно-адаптивной схемы VLC (CAVLC), схемы арифметического кодирования, бинаризации, контекстно-адаптивного двоичного арифметического кодирования (CABAC), основанного на синтаксисе контекстно-адаптивного двоичного арифметического кодирования (SBAC), энтропийного кодирования с интервалом вероятности (PIPE) или другого метода или способа

35 энтропийного кодирования) или обхода (без сжатия) квантованных коэффициентов 209, параметров внешнего предсказания, параметров внутреннего предсказания, параметров контурного фильтра и/или других элементов синтаксиса для получения данных 21 закодированного изображения, которые могут выводиться через выход 272, например, в форме закодированного потока 21 битов, так что, например, видеodeкодер

40 30 может принимать и использовать параметры для декодирования. Кодированный поток 21 битов может быть передан на видеodeкодер 30 или сохранен в памяти для последующей передачи или поиска видеodeкодером 30.

Другие структурные вариации видеodeкодера 20 могут использоваться для кодирования видеопотока. Например, кодер 20, не основанный на преобразовании, может квантовать

45 остаточный сигнал напрямую без блока 206 обработки преобразования для некоторых блоков или кадров. В другой реализации кодер 20 может иметь блок 208 квантования и блок 210 обратного квантования, объединенные в единый блок.

#### Декодер и способ декодирования

Фиг. 3 показывает пример видеодекодера 30, который сконфигурирован для реализации методик настоящей заявки. Видеодекодер 30 сконфигурирован для приема кодированных данных 21 изображения (например, кодированного потока 21 битов), например, закодированных кодером 20, для получения декодированного изображения 331. Кодированные данные изображения или поток битов содержат информацию для декодирования данных кодированного изображения, например данные, которые представляют блоки изображения кодированного слайса видео и связанные с ними элементы синтаксиса.

В примере на фиг. 3, декодер 30 содержит блок 304 энтропийного декодирования, блок 310 обратного квантования, блок 312 обработки обратного преобразования, блок 314 восстановления (например, сумматор 314), контурный фильтр 320, буфер декодированных изображений (DPB) 330, блок 344 внешнего предсказания и блок 354 внутреннего предсказания. Блок 344 внешнего предсказания может быть или включать в себя блок компенсации движения. Видеодекодер 30 может, в некоторых примерах, выполнять этап декодирования, в целом обратный этапу кодирования, описанному в отношении видеокодера 100 на Фиг.2.

Как пояснялось в отношении кодера 20, блока 210 обратного квантования, блока 212 обработки обратного преобразования, блока 214 восстановления, контурного фильтра 220, буфера 230 декодированных изображений (DPB), блок 344 внешнего предсказания и блок внутреннего предсказания 354 также упоминаются как образующие «встроенный декодер» видеокодера 20. Соответственно, блок 310 обратного квантования может быть идентичен по функциям блоку 110 обратного квантования, блок 312 обработки обратного преобразования может быть идентичен по функциям блоку 212 обработки обратного преобразования, блок 314 восстановления может быть идентичен по функциям блоку восстановления 214, контурный фильтр 320 может быть идентичен по функциям контурному фильтру 220, и буфер 330 декодированного изображения может быть идентичным по функции буферу 230 декодированного изображения. Следовательно, пояснения, предоставленные для соответствующих блоков и функций видеокодера 20, применяются соответственно к соответствующим модулям и функциям видеодекодера 30.

#### Энтропийное декодирование

Модуль 304 энтропийного декодирования выполнен с возможностью синтаксического анализа потока 21 битов или, в общем, данных 21 закодированного изображения) и выполнения, например, энтропийного декодирования для данных 21 закодированного изображения, чтобы получить, например, квантованные коэффициенты 309 и/или декодированные параметры кодирования (не показаны на Фиг.3), например, любые или все из параметров внешнего предсказания (например, индекс опорного изображения и вектор движения), параметр внешнего предсказания (например, индекс или режим внутреннего предсказания), параметры преобразования, параметры квантования, параметры контурного фильтра и/или другие элементы синтаксиса. Модуль 304 энтропийного декодирования может быть выполнен с возможностью применения алгоритмов или схем декодирования, соответствующих схемам кодирования, как описано в отношении модуля 270 энтропийного кодирования кодера 20. Блок 304 энтропийного декодирования может быть дополнительно сконфигурирован для предоставления параметров внешнего предсказания, параметра внутреннего предсказания и/или других элементов синтаксиса в блок 360 выбора режима и других параметров в другие блоки декодера 30. Видеодекодер 30 может принимать элементы синтаксиса на уровне слайса видео и/или уровне видеоблока.

### Обратное квантование

Блок 310 обратного квантования может быть сконфигурирован для приема параметров квантования (QP) (или, в общем, информации, относящейся к обратному квантованию) и квантованных коэффициентов из кодированных данных 21 изображения (например, путем анализа и/или декодирования, например, посредством блока 304 энтропийного декодирования) и применять на основе параметров квантования обратное квантование к декодированным квантованным коэффициентам 309 для получения деквантованных коэффициентов 311, которые также могут называться коэффициентами 311 преобразования. Процесс обратного квантования может включать в себя использование параметра квантования, определенного видеокодером 20 для каждого видеоблока в слайсе видео, для определения степени квантования и, аналогично, степени обратного квантования, которое должно применяться.

### Обратное преобразование

Блок 312 обработки обратного преобразования может быть сконфигурирован для приема деквантованных коэффициентов 311, также называемых коэффициентами 311 преобразования, и для применения преобразования к деквантованным коэффициентам 311, чтобы получить восстановленные остаточные блоки 213 в области выборки. Восстановленные остаточные блоки 213 также могут упоминаться как блоки 313 преобразования. Преобразование может быть обратным преобразованием, например, обратным DCT, обратным DST, обратным целочисленным преобразованием или концептуально аналогичным процессом обратного преобразования. Блок 312 обработки обратного преобразования может быть дополнительно сконфигурирован для приема параметров преобразования или соответствующей информации из данных 21 закодированного изображения (например, путем синтаксического анализа и/или декодирования, например, посредством блока 304 энтропийного декодирования), чтобы определять преобразование, которое должно применяться к деквантованным коэффициентам. 311.

### Восстановление

Блок 314 восстановления (например, сумматор или сумматор 314) может быть сконфигурирован для добавления восстановленного остаточного блока 313 к блоку 365 предсказания для получения восстановленного блока 315 в области выборки, например, путем добавления значений выборок восстановленного остаточного блока 313 к значениям выборки блока 365 предсказания.

### Фильтрация

Блок 320 контурного фильтра (либо в контуре кодирования, либо после контура кодирования) выполнен с возможностью фильтрации восстановленного блока 315 для получения отфильтрованного блока 321, например, для сглаживания переходов пикселей или иного улучшения качества видео. Блок 320 контурного фильтра может содержать один или несколько контурных фильтров, таких как фильтр деблокирования, фильтр с адаптивным смещением (SAO) или один или несколько других фильтров, например, двусторонний фильтр, адаптивный контурный фильтр (ALF), повышение резкости, сглаживающие фильтры или коллаборативные фильтры, или любая их комбинация. Хотя модуль 320 контурного фильтра показан на Фиг.3 как контурный фильтр, в других конфигурациях модуль 320 контурного фильтра может быть реализован как постконтурный фильтр.

### Буфер декодированного изображения

Декодированные видеоблоки 321 изображения затем сохраняются в буфере 330 декодированных изображений, который сохраняет декодированные изображения 331

в качестве опорных изображений для последующей компенсации движения для других изображений и/или для вывода, соответственно, отображения.

Декодер 30 выполнен с возможностью вывода декодированного изображения 311, например, через вывод 312 для представления или просмотра пользователю.

#### 5 Предсказание

Блок 344 внешнего предсказания может быть идентичен блоку 244 внешнего предсказания (в частности, блоку компенсации движения), а блок 354 внутреннего предсказания может быть идентичен блоку 254 внешнего предсказания по функциям и выполняет решения о разбиении или разделении и предсказание на основе параметров  
10 разделения и/или предсказания или соответствующей информации, принятой из данных 21 закодированного изображения (например, путем синтаксического анализа и/или декодирования, например, посредством блока 304 энтропийного декодирования). Блок 360 выбора режима может быть сконфигурирован для выполнения предсказания (внутреннего или внешнего предсказания) для каждого блока на основе восстановленных  
15 изображений, блоков или соответствующих выборок (фильтрованных или нефильтрованных) для получения блока 365 предсказания.

Когда слайс видео кодируется как слайс с внутренним кодированием (I), блок 354 внутреннего предсказания блока 360 выбора режима конфигурируется для генерации  
20 переданного в сигналах режима внутреннего предсказания и данных из ранее декодированных блоков текущего изображения. Когда видеоизображение кодируется как (то есть В или Р) слайс с внешним кодированием, блок 344 внешнего предсказания (например, блок компенсации движения) блока 360 выбора режима конфигурируется для создания блоков 365 предсказания для видеоблока слайса текущего видео на основе  
25 векторов движения и других элементов синтаксиса, принятых от блока 304 энтропийного декодирования. Для внешнего предсказания блоки предсказания могут быть созданы из одного из опорных изображений в одном из списков опорных изображений. Видеodeкодер 30 может создавать списки опорных кадров, Список 0 и Список 1, используя способы построения по умолчанию на основе опорных изображений,  
30 хранящихся в DRB 330.

Блок 360 выбора режима сконфигурирован для определения информации предсказания для видеоблока текущего слайса видео путем анализа векторов движения и других элементов синтаксиса и использует информацию предсказания для создания  
35 блоков предсказания для текущего декодируемого видеоблока. Например, модуль 360 выбора режима использует некоторые из принятых элементов синтаксиса для определения режима предсказания (например, внутреннее или внешнее предсказание), используемого для кодирования видеоблоков слайса видео, типа слайса внешнего предсказания (например, В-слайс, Р-слайс или GPB-слайс), информацию о построении для одного или нескольких списков опорных изображений для слайса, векторы движения  
40 для каждого интеркодированного видеоблока слайса, статус внешнего предсказания для каждого видеоблока с внешним предсказанием упомянутого слайса и другую информацию для декодирования видеоблоков в текущем слайсе видео.

Другие варианты видеodeкодера 30 могут использоваться для декодирования данных 21 закодированного изображения. Например, декодер 30 может формировать выходной  
45 видеопоток без блока 320 контурной фильтрации. Например, декодер 30, не основанный на преобразовании, может выполнить обратное квантование остаточного сигнала напрямую без блока 312 обработки обратного преобразования для некоторых блоков или кадров. В другой реализации видеodeкодер 30 может иметь блок 310 обратного

квантования и блок 312 обработки обратного преобразования, объединенные в один блок.

Следует понимать, что в кодере 20 и декодере 30 результат обработки текущего этапа может быть дополнительно обработан и затем выведен на следующий этап.

5 Например, после интерполяционной фильтрации, вывода вектора движения или петлевой фильтрации может выполняться дополнительная операция, такая как обрезка или сдвиг, над результатом обработки интерполяционной фильтрации, вывода вектора движения или петлевой фильтрации.

10 Следует отметить, что дополнительные операции могут применяться к производным векторам движения текущего блока (в том числе, но не ограничиваясь ими, векторы движения контрольной точки аффинного режима, векторы движения субблока в аффинном, планарном режимах, режимах ATMVP, временные векторы движения и т.д.). Например, значение вектора движения ограничено predetermined диапазоном в соответствии с его битом представления. Если представляющий бит вектора движения  
15 - bitDepth, то диапазон равен  $-2^{(bitDepth-1)} \sim 2^{(bitDepth-1)} - 1$ , где «^» означает возведение в степень. Например, если bitDepth установлено равным 16, диапазон составляет  $-32768 \sim 32767$ ; если bitDepth установлен равным 18, диапазон составляет  $-131072 \sim 131071$ . Например, значение производного вектора движения (например, MV четырех субблоков 4x4 в одном блоке 8x8) ограничено таким образом, что  
20 максимальная разница между целыми частями четырех MV субблока 4x4 не превышает N пикселей, например, не более 1 пикселя. Здесь представлены два способа ограничения вектора движения в соответствии с bitDepth.

Способ 1. Удалить старший бит переполнения (старший бит) с помощью потоковых операций

$$25 \quad ux = (mvx + 2^{bitDepth}) \% 2^{bitDepth} \quad (1)$$

$$mvx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (2)$$

$$uy = (mvy + 2^{bitDepth}) \% 2^{bitDepth} \quad (3)$$

$$30 \quad mvy = (uy \geq 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (4)$$

где mvx - горизонтальный компонент вектора движения блока или субблока изображения, mvy - вертикальный компонент вектора движения блока или субблока изображения, а ux и uy указывают промежуточное значение;

35 Например, если значение mvx равно -32769, после применения формул (1) и (2) результирующее значение будет 32767. В компьютерной системе десятичные числа хранятся в виде дополнения до двух. Дополнение до двух -32769 равно 1,0111,1111,1111,1111 (17 бит), затем старший бит отбрасывается, поэтому полученное дополнение до двух составляет 0111,1111,1111,1111 (десятичное число 32767), что то же самое, что выход, применяя формулы (1) и (2).

$$40 \quad ux = (mvrx + mvdx + 2^{bitDepth}) \% 2^{bitDepth} \quad (5)$$

$$mvx = (ux \geq 2^{bitDepth-1}) ? (ux - 2^{bitDepth}) : ux \quad (6)$$

$$uy = (mvry + mvdy + 2^{bitDepth}) \% 2^{bitDepth} \quad (7)$$

$$45 \quad mvy = (uy \geq 2^{bitDepth-1}) ? (uy - 2^{bitDepth}) : uy \quad (8)$$

Операции могут применяться во время суммирования mvr и mvd, как показано в формулах (5) - (8).

Способ 2: удалить старший бит переполнения, обрезав значение

$$vx = \text{Clip3}(-2^{bitDepth-1}, 2^{bitDepth-1} - 1, vx)$$

$$v_y = \text{Clip3}(-2^{\text{bitDepth}-1}, 2^{\text{bitDepth}-1} - 1, v_y)$$

где  $v_x$  - горизонтальный компонент вектора движения блока или субблока изображения,  $v_y$  - вертикальный компонент вектора движения блока или субблока изображения;  $x$ ,  $y$  и  $z$  соответственно соответствуют трем входным значениям процесса  
5 обрезки MV, а определение функции Clip3 выглядит следующим образом:

$$\text{Clip3}(x, y, z) = \begin{cases} x & ; & z < x \\ y & ; & z > y \\ z & ; & \text{otherwise} \end{cases}$$

(otherwise - иначе)

10 Фиг. 4 является схематической диаграммой устройства 400 кодирования видео согласно варианту осуществления раскрытия. Устройство 400 кодирования видео подходит для реализации раскрытых вариантов осуществления, как описано в данном документе. В варианте осуществления устройство 400 кодирования видео может быть  
15 декодером, таким как видеодекoder 30 по фиг. 1А или кодером, таким как видеокодер 20 по фиг. 1А.

Устройство 400 кодирования видео содержит входные порты 410 (или входные порты 410) и блоки 420 приемника (Rx) для приема данных; процессор, логический блок или  
20 центральный процессор (ЦП) 430 для обработки данных; блоки передатчика (Tx) 440 и выходные порты 450 (или выходные порты 450) для передачи данных; и память 460 для хранения данных. Устройство 400 кодирования видео может также содержать оптико-электрические (ОЕ) компоненты и электрические-оптические (ЕО) компоненты, подключенные к входным портам 410, блокам 420 приемника, блокам 440 передатчика  
и выходным портам 450 для выхода или входа оптических или электрических сигналов.

25 Процессор 430 реализован аппаратно и программно. Процессор 430 может быть реализован как одна или несколько микросхем ЦП, ядер (например, как многоядерный процессор), FPGA, ASIC и DSP. Процессор 430 поддерживает связь с входными портами 410, блоками 420 приемника, блоками 440 передатчика, выходными портами 450 и памятью 460. Процессор 430 содержит блок 470 кодирования. Блок 470 кодирования  
30 реализует раскрытые варианты осуществления, описанные выше. Например, блок 470 кодирования реализует, обрабатывает, подготавливает или предоставляет различные операции кодирования. Таким образом, включение в состав блока 470 кодирования обеспечивает существенное улучшение функциональных возможностей устройства 400 кодирования видео и обеспечивает преобразование устройства 400 кодирования видео  
35 в другое состояние. В качестве альтернативы блок 470 кодирования реализован как инструкции, хранящиеся в памяти 460 и выполняемые процессором 430.

Память 460 может содержать один или несколько дисков, ленточных накопителей и твердотельных накопителей и может использоваться в качестве запоминающего  
40 устройства для переполнения данных, для хранения программ, когда такие программы выбраны для выполнения, и для хранения инструкций и данных, которые считываются во время выполнения программы. Память 460 может быть, например, энергозависимой и/или энергонезависимой и может быть постоянным запоминающим устройством (ROM), оперативным запоминающим устройством (RAM), троичной памятью с адресацией по содержимому (TCAM) и/или статической памятью произвольного доступа (SRAM).

45 Фиг. 5 является упрощенной блок-схемой устройства 500, которое может использоваться в качестве одного или обоих из исходного устройства 12 и целевого устройства 14 на фиг. 1А согласно примерному варианту осуществления.

Процессор 502 в устройстве 500 может быть центральным процессором. В качестве

альтернативы, процессор 502 может быть устройством любого другого типа или множеством устройств, способных манипулировать или обрабатывать информацию, существующую в настоящее время или разработанную в будущем. Хотя раскрытые реализации могут быть реализованы с одним процессором, как показано, например процессором 502, преимущества в скорости и эффективности могут быть достигнуты с использованием более чем одного процессора.

Память 504 в устройстве 500 может быть постоянным запоминающим устройством (ROM) или устройством оперативной памяти (RAM) в некоторой реализации. В качестве памяти 504 может использоваться любой другой подходящий тип запоминающего устройства. Память 504 может включать в себя код и данные 506, к которым обращается процессор 502, используя шину 512. Память 504 может дополнительно включать в себя операционную систему 508 и прикладные программы 510, прикладные программы 510 включают в себя по меньшей мере одну программу, которая позволяет процессору 502 выполнять описанные здесь способы. Например, прикладные программы 510 могут включать в себя приложения с 1 по N, которые дополнительно включают в себя приложение кодирования видео, которое выполняет описанные здесь способы.

Устройство 500 также может включать в себя одно или несколько устройств вывода, таких как дисплей 518. Дисплей 518 может быть, в одном примере, дисплеем, чувствительным к касанию, который объединяет дисплей с элементом, чувствительным к касанию, способным считывать вводы касанием. Дисплей 518 может быть подключен к процессору 502 через шину 512.

Хотя здесь изображена как одна шина, шина 512 устройства 500 может состоять из нескольких шин. Кроме того, вторичное хранилище 514 может быть напрямую связано с другими компонентами устройства 500 или может быть доступно через сеть и может содержать один интегрированный блок, такой как карта памяти, или несколько блоков, например, несколько карт памяти. Таким образом, устройство 500 может быть реализовано в широком разнообразии конфигураций.

Внутреннее предсказание выборок цветности может быть выполнено с использованием выборок восстановленного блока яркости.

Во время разработки HEVC было предложено внутреннее предсказание цветности кросс-компонентной линейной модели (CCLM) [J. Ким, С.-В. Парк, Ж.-Й. Парк, Б.-М. Чон, Intra Chroma Prediction Using Inter Channel Correlation, document JCTVC-B021, Jul. 2010]. CCLM использует линейную корреляцию между выборкой цветности и выборкой яркости в соответствующей положению в блоке кодирования. Когда блок цветности кодируется с использованием CCLM, линейная модель выводится из восстановленных соседних выборок яркости и цветности с помощью линейной регрессии. Затем выборки цветности в текущем блоке могут быть предсказаны по восстановленным выборкам яркости в текущем блоке с выведенной линейной моделью (см. Фиг 6):

$$C(x, y) = \alpha \times L(x, y) + \beta ,$$

где  $C$  и  $L$  указывают значения цветности и яркости соответственно. Параметры  $\alpha$  и  $\beta$  определяются способом наименьших квадратов следующим образом:

$$\alpha = \frac{R(L, C)}{R(L, L)} ,$$

$$\beta = M(C) - \alpha \times M(L)$$

где  $M(A)$  представляет собой среднее значение  $A$ , а  $R(A, B)$  определяется следующим образом:

$$R(A, B) = M((A - M(A)) \times (B - M(B))) .$$

Если кодированное или декодированное изображение имеет формат, который задает различное количество выборок для компонентов яркости и цветности (например, формат YCbCr 4: 2: 0), выборки яркости подвергаются понижающей дискретизации перед моделированием и предсказанием.

Способ был принят для использования в VTM2.0. В частности, вывод параметров выполняется следующим образом:

$$\alpha = \frac{N \cdot \sum(L(n) \cdot C(n)) - \sum L(n) \cdot \sum C(n)}{N \cdot \sum(L(n) \cdot L(n)) - \sum L(n) \cdot \sum L(n)},$$

$$\beta = \frac{\sum C(n) - \alpha \cdot \sum L(n)}{N},$$

где  $L(n)$  представляет верхнюю и левую соседние восстановленные выборки яркости с пониженной дискретизацией,  $C(n)$  представляет верхнюю и левую соседние восстановленные выборки цветности.

В [G. Ларош, Ж. Таке, К. Жиске, П. Онно (Canon), «CE3: Cross-component linear model simplification (Test 5.1)», Input document to 12th JVET Meeting in Macao, China, Oct. 2018] был предложен другой способ получения  $\alpha$  и  $\beta$  (см. Рис. 7). В частности, параметры  $\alpha$  и  $\beta$  линейной модели получаются согласно следующим уравнениям:

$$\alpha = \frac{C(B) - C(A)}{L(B) - L(A)}$$

$$\beta = L(A) - \alpha C(A)$$

где  $B = \operatorname{argmax}(L(n))$  и  $A = \operatorname{argmin}(L(n))$  - положения максимального и минимального значений в выборках яркости.

Также было предложено реализовать операцию деления с помощью умножения на число, которое хранится в справочной таблице (LUT), которая указана в таблице 1. Эта замена возможна следующим способом:

$$\frac{v_0}{v_1} \approx v_0 \cdot \text{LUT}[v_1] \gg S$$

где  $S$  параметр сдвига, задающий точность.

В таблице 1 показано сопоставление диапазона индексов LUT (указанного в первой строке таблицы) со списками значений, хранящимися в LUT, причем каждый список соответствует диапазону его индексов. Можно заметить, что значения  $\text{LUT}[v_1]$  можно рассчитать следующим образом:

$$\text{LUT}[v_1] = \frac{1 \ll S}{v_1}.$$

Используя LUT, определенный в таблице 1 (или рассчитанный эквивалентным образом с использованием приведенного выше уравнения), вычисление  $\alpha$  выполняется следующим образом:

$$\alpha = \frac{C(B) - C(A)}{L(B) - L(A)} \approx (C(B) - C(A)) \cdot \text{LUT}[L(B) - L(A)] \gg S_0$$

Параметр сдвига  $S$  можно разложить на несколько частей, т. е.  $S = S_0 + S_1$ , потому что значение  $\alpha$  используется в дополнительном вычислении. Это разложение обеспечивает гибкость точности вычислений на различных этапах, и, таким образом, можно перераспределять битовую глубину операций умножения по шагам получения значения предсказанной выборки цветности. В конкретной реализации:

$$S_0 = 1,$$

$$S_1 = 15,$$

$$S = S_0 + S_1 = 16.$$

Таблица 1. Примерная таблица LUT для реализации операции деления с использованием умножения на число

5

Диапазоны индексов LUT:															
0	32	64	96	128	160	192	224	256	288	320	352	384	416	448	480
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
31	63	95	127	159	191	223	255	287	319	351	383	415	447	479	511
Значения, хранящиеся в LUT:															
65536	1985	1008	675	508	407	339	291	255	226	204	185	170	157	145	136
32768	1927	992	668	504	404	337	289	254	225	203	185	169	156	145	135
21845	1872	978	661	500	402	336	288	253	225	202	184	169	156	145	135
16384	1820	963	655	496	399	334	287	252	224	202	184	168	156	144	135
13107	1771	949	648	492	397	332	286	251	223	201	183	168	155	144	135
10922	1724	936	642	489	394	330	284	250	222	201	183	168	155	144	134
9362	1680	923	636	485	392	329	283	249	222	200	182	167	154	144	134
8192	1638	910	630	481	390	327	282	248	221	199	182	167	154	143	134
7281	1598	897	624	478	387	326	281	247	220	199	181	166	154	143	134
6553	1560	885	618	474	385	324	280	246	219	198	181	166	153	143	133
5957	1524	873	612	471	383	322	278	245	219	197	180	165	153	142	133
5461	1489	862	606	468	381	321	277	244	218	197	180	165	153	142	133
5041	1456	851	601	464	378	319	276	243	217	196	179	165	152	142	132
4681	1424	840	595	461	376	318	275	242	217	196	179	164	152	141	132
4369	1394	829	590	458	374	316	274	241	216	195	178	164	152	141	132
4096	1365	819	585	455	372	315	273	240	215	195	178	163	151	141	132
3855	1337	809	579	451	370	313	271	240	214	194	177	163	151	140	131
3640	1310	799	574	448	368	312	270	239	214	193	177	163	151	140	131
3449	1285	789	569	445	366	310	269	238	213	193	176	162	150	140	131
3276	1260	780	564	442	364	309	268	237	212	192	176	162	150	140	131
3120	1236	771	560	439	362	307	267	236	212	192	175	161	149	139	130
2978	1213	762	555	436	360	306	266	235	211	191	175	161	149	139	130
2849	1191	753	550	434	358	304	265	234	210	191	174	161	149	139	130
2730	1170	744	546	431	356	303	264	234	210	190	174	160	148	138	130
2621	1149	736	541	428	354	302	263	233	209	189	173	160	148	138	129
2520	1129	728	537	425	352	300	262	232	208	189	173	159	148	138	129
2427	1110	720	532	422	350	299	261	231	208	188	172	159	147	137	129
2340	1092	712	528	420	348	297	260	230	207	188	172	159	147	137	129
2259	1074	704	524	417	346	296	259	229	206	187	172	158	147	137	128
2184	1057	697	520	414	344	295	258	229	206	187	171	158	146	137	128
2114	1040	689	516	412	343	293	257	228	205	186	171	157	146	136	128
2048	1024	682	512	409	341	292	256	227	204	186	170	157	146	136	128

10

15

20

25

30

35

В этом случае коэффициент  $\alpha$  линейной модели имеет целочисленное представление с фиксированной точкой дробного значения, точность  $\alpha$  определяется значением  $S1 = S - S_0$ , которое используется для получения значения предсказанной выборки цветности:

40

$$pred_c(i, j) = (\alpha \cdot rec^{(i, j)}) \gg S_1 + \beta$$

Размер LUT довольно критичен для аппаратной реализации кодера или декодера. Самый простой способ решить эту проблему - регулярно выполнять субдискретизацию LUT, сохраняя только каждый N-й элемент (здесь N - коэффициент субдискретизации) исходной заданной LUT, т.е. в таблице 1.

45

После регулярной субдискретизации (подвыборке) с коэффициентом субдискретизации степени двойки, равным N, извлечение из LUT определяется по-другому:

$$m = \text{LUT}[(L(B) - L(A) \gg \log_2(N))]$$

вместо

$$m = \text{LUT}[L(B) - L(A)]$$

5 Можно заметить, что для естественных изображений вероятность  $L(B) - L(A)$  будет иметь небольшое значение, больше, чем вероятность того, что эта разница будет большой. Другими словами, вероятность появления значения из таблицы 1 уменьшается от левого столбца к правому столбцу, а внутри каждого столбца эта вероятность уменьшается от первого до последнего элемента, принадлежащего этому столбцу.

10 Примерная зависимость вероятности возникновения от значения  $L(B) - L(A)$  приведена на рисунке 8.

Следовательно, сохранение только каждого N-го элемента исходной таблицы LUT далеко не оптимально, поскольку соответствует равновероятному распределению его аргумента, что неверно.

15 Рассматривая это распределение, можно достичь лучшего компромисса между размером LUT и точностью вычислений, чем может обеспечить обычная субдискретизация.

Предлагается определить LUT с использованием нелинейной индексации, в частности, таким образом, чтобы две соседние записи LUT соответствовали различным шагам  $L(B) - L(A)$ , и значение этого шага увеличивается с индексом записи.

20 Одним из эффективных с вычислительной точки зрения решений является определение индекса в LUT с использованием нескольких наиболее значимых битов  $L(B) - L(A)$ . Положение старшего бита  $L(B) - L(A)$  определяет точность (т. е. шаг между двумя соседними записями LUT) на основе положения старших битов. Большие значения   
25 положения старшего бита соответствуют меньшей точности и большему значению шага.

Конкретный вариант осуществления показан на фиг.9 и фиг.10. На Фиг.9 показана блок-схема вычисления значений LUT, а на Фиг.10 показано, как получить индекс «idx»   
30 в LUT, соответствующий входному значению  $L(B) - L(A)$ .

Используя шаги, показанные на рисунке 9, можно получить значения, которые будут сохранены в LUT для их дальнейшего использования при моделировании SCLM. На   
рис. 9 переменная *ctr* перебирает все возможные значения  $L(B) - L(A) - 1$ . Можно   
35 заметить, что на самом деле LUT имеет поддиапазоны, каждый поддиапазон содержит записи «col<sub>max</sub>». Внутри поддиапазона значение «ctr» увеличивается с равным значением «шага». Для поддиапазона, имеющего индекс «lut\_shift+1», который больше 1, соответствующее значение «шага» увеличено вдвое по сравнению с поддиапазоном «lut\_shift». Первые два поддиапазона будут иметь шаг, равный единице из-за порога в   
40 операции «step=1<<max(0,lut\_shift)».

Как показано на фиг.9, блок-схема иллюстрирует примерный процесс создания справочной таблицы в соответствии с вариантом осуществления изобретения. На этапе   
902 устройство кодирования видео запускает процесс генерации справочной таблицы. Устройство кодирования видео может быть декодером, таким как видеодекoder 30 на   
фиг. 1А-1В, фиг. 3, или кодером, таким как видеокодер 20 на фиг. 1А-1В, фиг. 2, или   
45 устройство 400 кодирования видео по фиг. 4, или устройство 500 по фиг. 5.

На этапе 904 пусть *ctr*=1 и *lut\_shift*=1. На этапе 906 определяется, является ли индекс *lut\_shift*<*lut\_shift*<sub>max</sub>. Если *lut\_shift*<*lut\_shift*<sub>max</sub>, шаг рассчитывается как  $1 \ll \max(0, \text{lut\_shift})$ , и *col*=0 на этапе 908; иначе, процесс генерации заканчивается на этапе 922.

На этапе 910 начальное смещение обеспечивается этапом « $\text{ctr}=\text{ctr}+(\text{step}\gg 1)$ ». Тогда будет ли  $\text{col}<\text{col}_{\text{max}}$  определяется на этапе 912. Если  $\text{col}<\text{col}_{\text{max}}$ ,  $\text{LUT}[\text{ctr}]=1<<\text{S}/\text{ctr}$  на этапе 914, здесь, предварительно вычисленные значения LUT, которые определены на этапах 912-918, генерируют одну строку LUT; иначе, пусть индекс  $\text{lut\_shift}=\text{lut\_shift}+1$  на этапе 920. На этапе 916 значения  $\text{ctr}$ , соответствующие началу каждого поддиапазона, устанавливаются как  $\text{ctrl}+\text{step}$ . На этапе 918 процесс генерации переходит к следующему столбцу, затем процесс возвращается к этапу 912.

Примерная LUT, сгенерированная с использованием блок-схемы, показанной на фиг.9, приведена в таблице 2. Строка этой таблицы соответствует поддиапазону с индексом « $\text{lut\_shift}$ ». Эта таблица была получена с « $\text{lut\_shift}_{\text{max}}$ », равным 6, и « $\text{col}_{\text{max}}$ », равным 3, в результате чего в LUT 48 записей. Значения  $\text{ctr}$ , соответствующие началу каждого поддиапазона (нулевое значение « $\text{col}$ »), следующие:

0, 8, 17, 35, 71, 143, 287.

Эти значения не всегда являются степенью двойки, потому что субдискретизация в каждом поддиапазоне выполняется относительно середины поддиапазона. Соответствующее начальное смещение обеспечивается шагом « $\text{ctr}=\text{ctr}+(\text{step}\gg 1)$ », показанным на рис. 9. Значение «шаг» для значений « $\text{lut\_shift}$ » не больше 0 устанавливается равным 1.

Таблица 2. Примерная LUT, сгенерированная с использованием блок-схемы, показанной на рис. 9

lut_shift	col							
	0	1	2	3	4	5	6	7
-1	65536	32768	21845	16384	13107	10922	9362	8192
0	7281	6553	5957	5461	5041	4681	4369	4096
1	3640	3276	2978	2730	2520	2340	2184	2048
2	1820	1638	1489	1365	1260	1170	1092	1024
3	910	819	744	682	630	585	546	512
4	455	409	372	341	315	292	273	256
5	227	204	186	170	157	146	136	128

На фиг. 10А двоичное представление 1001 входного значения (например, разность  $L(B) - L(A)$ ) обрабатывается для определения положения соответствующей записи в LUT, показанной в таблице 2. Старший ненулевой бит входного значения помечается как 1002. Положение этого бита определяет значение « $\text{msb}$ ». Фактически, значение  $\text{msb}$  - это  $\log_2()$  входного значения 1001. Вычитание 1 из « $\text{msb}$ » дает значение « $\text{lut\_shift}$ », которое выбирает строку в таблице 2. В противном случае шаг рассчитывается как « $1<<\text{lut\_shift}$ ».

Выбор столбца выполняется путем взятия битов « $\text{col}_{\text{max}}$ » рядом с 1002. Значение столбца « $\text{col}$ » в таблице 2 получается следующим образом:

значение « $\text{high\_bits}$ » получается путем выбора битов « $\text{col}_{\text{max}}+1$ », следующих за старшим битом 1020;

$\text{col}$  устанавливается равным значению, хранящемуся в « $\text{high\_bits}$ », уменьшенному на 1.

Для случая, когда шаг выравнивания « $\text{ctr}=\text{ctr}+(\text{step}\gg 1)$ » не выполняется на Фиг.9, вывод значения « $\text{lut\_shift}$ » такой же, а вывод значения « $\text{col}$ » еще проще (Фиг. 10В): значение « $\text{col}$ » получается путем выбора битов « $\text{col}_{\text{max}}$ », следующих за старшим битом 1020. Этот способ выведения индекса соответствует таблице 3. Значения « $\text{ctr}$ » (фиг. 9), соответствующие началу каждого поддиапазона (нулевое значение « $\text{col}$ »), следующие:

0, 8, 16, 32, 64, 128, 256

Таблица 3. Другая примерная LUT, сгенерированная с использованием блок-схемы, показанной на фиг. 9, в случае, когда «ctr=ctr+(step>>1)» пропущено

lut_shift	col							
	0	1	2	3	4	5	6	7
-1	65536	32768	21845	16384	13107	10922	9362	8192
0	7281	6553	5957	5461	5041	4681	4369	4096
1	3855	3449	3120	2849	2621	2427	2259	2114
2	1985	1771	1598	1456	1337	1236	1149	1074
3	1008	897	809	736	675	624	579	541
4	508	451	407	370	339	313	291	271
5	255	226	204	185	170	157	145	136

При выведении значения «col» msb может быть меньше или равным «col<sub>max</sub>». В этом случае значение «col» устанавливается равным младшим значащим битам «col<sub>max</sub>»

входной разности 1001.

В практических реализациях LUT-индекс является одномерным. Обычно понимается, что LUT, показанные в таблицах 2 и 3, можно адресовать линейно, используя набор индексов, равный:

$$\max(\text{lut\_shift}, 0) \cdot (1 \ll \text{col}_{\max}) + \text{col}.$$

Обе LUT, показанные в Таблице 2 и Таблице 3, хранят значения, которые сильно различаются по своей величине. Следовательно, разумно иметь одинаковое количество битов для всех значений, хранящихся в LUT. Значение, извлекаемое из LUT, может быть дополнительно сдвинуто влево в соответствии со значением lut\_shift. Единственное исключение из этого правила - первые четыре значения, которые имеют отличную точность с последними четырьмя значениями в этой строке. Однако эту проблему можно решить с помощью дополнительной таблицы LUT, в которой хранятся эти дополнительные сдвиги для первых четырех значений. В этом варианте осуществления значение множителя восстанавливается из значения, извлеченного из LUT, следующим образом:

$$m = \text{LUT}[\text{idx}] \ll m_s$$

где  $m_s = 6 - (\text{lut\_shift} + 1) + \delta$ . Значение  $\delta$  устанавливается равным 3, 2, 1, 1, соответственно, для значений «idx», меньших или равных 4. Справочная таблица для этого варианта приведена в таблице 4.

Таблица 4. Другая примерная LUT, сгенерированная с использованием блок-схемы, показанной на фиг.9, в случае равной точности в пределах диапазонов.

lut_shift	col							
	0	1	2	3	4	5	6	7
-1	128	128	171	128	205	171	146	128
0	228	205	186	171	158	146	137	128
1	241	216	195	178	164	152	141	132
2	248	221	200	182	167	155	144	134
3	252	224	202	184	169	156	145	135
4	254	226	204	185	170	157	146	136
5	128	128	171	128	205	171	146	128

Можно заметить, что последние строки таблицы 4 очень похожи друг на друга. Следовательно, можно уменьшить размер LUT, сохранив только одну строку для некоторых наборов поддиапазонов. В конкретных вариантах осуществления, когда

значение «lut\_shift» больше некоторого порога, значение lut\_shift устанавливается равным пороговому значению, а значение  $\delta$  уменьшается на разницу между начальным значением «lut\_shift» и значением порогового значения.

5 Фиг. 11 - блок-схема, иллюстрирующая примерное внутреннее предсказание выборки цветности блока с применением кросс-компонентной линейной модели. На этапе 1102 устройство кодирования видео получает восстановленные выборки яркости. Устройство кодирования видео может быть декодером, таким как видеodeкодер 30 на фиг. 1A-1B, фиг. 3, или кодером, таким как видеodeкодер 20 на фиг. 1A-1B, фиг. 2, или устройство 400 кодирования видео по фиг. 4, или устройство 500 по фиг. 5.

10 На этапе 1104 устройство кодирования видео определяет положения максимального и минимального значений восстановленной выборки в восстановленных выборках яркости. Например, восстановленные выборки яркости являются соседними восстановленными выборками яркости, соответствующими выборке цветности.

15 На этапе 1106 устройство кодирования видео получает значение разности максимума и минимума восстановленных выборок яркости.

На этапе 1108 устройство кодирования видео вычисляет индекс для LUT, чтобы получить значение множителя, которое соответствует значению разности максимума и минимума восстановленных выборок яркости.

20 Например, устройство кодирования видео определяет положение старшего бита разности максимального и минимального значений выборки яркости и использует набор битов, следующих за положением старшего бита разности максимального и минимального значений выборок яркости в качестве индекса для LUT, чтобы извлечь значение. Положение старшего бита разности максимального и минимального значений выборок яркости может быть получено как логарифм упомянутой разности по  
25 основанию два. Устройство кодирования видео определяет набор битов, следующих за положением старшего бита разности. Как возможный результат, набор битов включает в себя четыре бита.

LUT генерируется с шагом выравнивания или без него. LUT может включать в себя по меньшей мере два соседних значения, хранящихся в LUT, которые соответствуют  
30 различным шагам полученной разности, и значение этого шага увеличивается со значением разности или является константой.

Как раскрыто в примерных таблицах 1-4, LUT может включать в себя поддиапазоны значений. Шаг значения разности максимума и минимума восстановленных выборок яркости постоянен в пределах поддиапазона; и шаг для разных поддиапазонов разный.  
35 Например, шаг значения разности максимума и минимума восстановленных выборок яркости увеличивается с увеличением индекса поддиапазона. Например, шаг значения разности максимума и минимума восстановленных выборок яркости может быть степенью двойки индекса поддиапазона.

Пусть LUT включает в себя по меньшей мере три значения: первое значение, второе  
40 значение и третье значение. Среди трех значений первое значение и второе значение являются двумя соседними значениями, и второе значение и третье значение - двумя соседними значениями. Шаг (то есть точность или разница) между первым значением и вторым значением может быть равен или может отличаться от шага между вторым значением и третьим значением. Если первое значение индексируется первым набором битов, а второе значение индексируется вторым набором битов, первое значение меньше  
45 второго значения, когда значение первого набора битов больше, чем значение второго набора битов; или первое значение больше второго значения, когда значение первого набора битов меньше значения второго набора битов.

LUT разделена на поддиапазоны. Индекс поддиапазона определяется с использованием положения старшего ненулевого бита разности максимума и минимума восстановленных выборок яркости. Например, размер поддиапазона равен 8, а количество поддиапазонов - 6. В качестве другого примера, разные соседние поддиапазоны имеют разное увеличение значения с одним и тем же шагом.

LUT может включать в себя нелинейные индексы. Две соседние записи LUT соответствуют различным шагам  $L(B)-L(A)$ , где  $L(B)$  представляет максимальное значение восстановленных выборок яркости,  $L(A)$  представляет минимальное значение восстановленных выборок яркости. Значение шага записи может увеличиваться с увеличением индекса записи.

Когда LUT использует несколько старших битов  $L(B)-L(A)$ , положение старшего бита  $L(B)-L(A)$  определяет точность (то есть шаг между двумя соседними записями LUT) на основе положения старших битов. Большие значения положения старшего бита могут соответствовать более низкой точности и большему значению шага.

На этапе 1110 устройство кодирования видео получает параметры  $\alpha$  и  $\beta$  линейной модели путем умножения извлеченного значения на разность максимального и минимального значений восстановленных выборок цветности.

На этапе 1112 устройство кодирования видео вычисляет предсказанное значение выборки цветности с использованием полученных параметров  $\alpha$  и  $\beta$  линейной модели.

Фиг. 12 является блок-схемой, показывающей примерную структуру устройства 1200 для внутреннего предсказания выборки цветности блока путем применения кросс-компонентной линейной модели. Устройство 1200 сконфигурировано для выполнения вышеуказанных способов и может включать в себя:

блок 1210 получения, сконфигурированный для получения восстановленных выборок яркости;

блок 1220 определения, сконфигурированный для определения максимальных и минимальных значений выборок яркости на основе восстановленных выборок яркости;

блок 1210 получения, дополнительно сконфигурированный для получения разности максимального и минимального значений выборок яркости.

блок 1220 определения, дополнительно сконфигурированный для определения положения старшего бита разности максимального и минимального значений выборок яркости. Например, положение старшего бита разности максимального и минимального значений выборки яркости является логарифмом упомянутой разности по основанию два. В качестве реализации старший бит - это первый ненулевой бит.

Устройство 1200 дополнительно включает в себя блок вычисления 1230, сконфигурированный для извлечения значения из справочной таблицы (LUT) с использованием набора битов в качестве индекса, причем набор битов следует за положением старшего бита разности максимальных и минимальных значений выборок яркости, получения параметров  $\alpha$  и  $\beta$  линейной модели на основе выбранного значения; и вычисления предсказанного значения выборки цветности с использованием полученных параметров  $\alpha$  и  $\beta$  линейной модели. Например, набор битов содержит четыре бита.

Блок вычисления может получить параметры  $\alpha$  и  $\beta$  линейной модели на основе выбранного значения и разности максимального и минимального значений восстановленных выборок цветности. Например, блок вычисления получает параметры  $\alpha$  и  $\beta$  линейной модели путем умножения извлеченного значения на разность максимального и минимального значений восстановленных выборок цветности.

Преимущество вариантов осуществления изобретения

1. Индекс для LUT вычисляется элегантным способом, который извлекает несколько битов в двоичном представлении. В результате повышается эффективность извлечения значения из LUT.

2. Поскольку эффективность извлечения значения из LUT увеличивается, величина множителя для получения параметров  $\alpha$  и  $\beta$  линейной модели сводится к минимуму.

3. Размер LUT минимизирован. Кривая функции деления ( $f(x) = 1/x$ , известная как гипербола) в вариантах осуществления настоящего изобретения была аппроксимирована следующим образом:

1) Размер таблицы LUT может быть 16

(минимальное количество записей для аппроксимации  $1/x$  кривой, производная которой изменяется от 0 до бесконечности)

2) Элементы LUT имеют нелинейную зависимость от индекса записи (для аппроксимации кривой  $1/x$ )

3) Множители (элементы LUT) - это 3-битные целые числа без знака (0... 7).

(минимальная точность для аппроксимации  $1/x$  кривой, производная которой изменяется от 0 до бесконечности)

Ниже приводится объяснение применений способа кодирования, а также способа декодирования, как показано в вышеупомянутых вариантах осуществления, и системы, использующей их.

Фиг. 13 - это блок-схема, показывающая систему 3100 поставки контента для реализации услуги распространения контента. Эта система 3100 предоставления контента включает в себя устройство 3102 захвата, терминальное устройство 3106 и, необязательно, включает в себя дисплей 3126. Устройство 3102 захвата обменивается данными с терминальным устройством 3106 по линии 3104 связи. Линия связи может включать в себя канал 13 связи, описанный выше. Линия связи 3104 включает в себя, помимо прочего, WIFI, Ethernet, кабель, беспроводную связь (3G/4G/5G), USB или любую их комбинацию и т.п.

Устройство 3102 захвата генерирует данные и может кодировать данные способом кодирования, как показано в приведенных выше вариантах осуществления. В качестве альтернативы устройство 3102 захвата может распространять данные на сервер потоковой передачи (не показан на фигурах), и сервер кодирует данные и передает закодированные данные на терминальное устройство 3106. Устройство 3102 захвата включает в себя, помимо прочего, камеру, смартфон или планшет, компьютер или портативный компьютер, систему видеоконференцсвязи, PDA, установленное на транспортном средстве устройство или комбинацию любого из них или тому подобное. Например, устройство 3102 захвата может включать в себя исходное устройство 12, как описано выше. Когда данные включают в себя видео, видеокодер 20, включенный в устройство 3102 захвата, может фактически выполнять обработку кодирования видео. Когда данные включают в себя аудио (то есть голос), аудиокодер, включенный в устройство 3102 захвата, может фактически выполнять обработку кодирования аудио. Для некоторых практических сценариев устройство 3102 захвата распределяет закодированные видео- и аудиоданные, мультиплексируя их вместе. Для других практических сценариев, например, в системе видеоконференцсвязи, закодированные аудиоданные и закодированные видеоданные не мультиплексируются. Устройство 3102 захвата отдельно распределяет закодированные аудиоданные и закодированные видеоданные на терминальное устройство 3106.

В системе 3100 доставки контента терминальное устройство 310 принимает и воспроизводит закодированные данные. Терминальное устройство 3106 может быть

устройством с возможностью приема и восстановления данных, таким как смартфон или планшет 3108, компьютер или ноутбук 3110, сетевой видеореги­стратор (NVR)/цифровой видеоманитон (DVR) 3112, телевизор 3114, телеприставка (STB) 3116, система 3118 видеоконференцсвязи, система 3120 видеонаблюдения, персональный цифровой помощник (PDA) 3122, устройство 3124, установленное на транспортном средстве, или их комбинацию, или тому подобное, способное декодировать вышеупомянутые закодированные данные. Например, терминальное устройство 3106 может включать в себя целевое устройство 14, как описано выше. Когда кодированные данные включают в себя видео, видеодекодеру 30, включенному в терминальное устройство, отдается приоритет для выполнения декодирования видео. Когда кодированные данные включают в себя аудио, аудиодекодер, включенный в терминальное устройство, получает приоритет для выполнения обработки декодирования звука.

Для терминального устройства с дисплеем, например, смартфона или планшета 3108, компьютера или ноутбука 3110, сетевого видеореги­стратора (NVR)/цифрового видеоманитона (DVR) 3112, телевизора 3114, персонального цифрового помощника (PDA) 3122, устройства 3124, установленного на транспортном средстве, терминальное устройство может передавать декодированные данные на свой дисплей. Для терминального устройства, не оснащенного дисплеем, например STB 3116, системы 3118 видеоконференцсвязи или системы 3120 видеонаблюдения, в нем устанавливается контакт с внешним дисплеем 3126 для приема и отображения декодированных данных.

Когда каждое устройство в этой системе выполняет кодирование или декодирование, может использоваться устройство кодирования изображения или устройство декодирования изображения, как показано в вышеупомянутых вариантах осуществления.

Фиг. 14 - схема, показывающая структуру примера терминального устройства 3106. После того, как терминальное устройство 3106 принимает поток от устройства 3102 захвата, блок 3202 обработки протокола анализирует протокол передачи потока. Протокол включает в себя, помимо прочего, протокол потоковой передачи в реальном времени (RTSP), протокол передачи гипертекста (HTTP), протокол потоковой передачи в реальном времени HTTP (HLS), MPEG-DASH, транспортный протокол в реальном времени (RTP), протокол обмена сообщениями в реальном времени (RTMP), или любую их комбинацию и т.п.

После того, как блок 3202 обработки протокола обрабатывает поток, генерируется файл потока. Файл выводится в блок 3204 демультимплексирования. Блок 3204 демультимплексирования может разделять мультимплексированные данные на кодированные аудиоданные и кодированные видеоданные. Как описано выше, для некоторых практических сценариев, например, в системе видеоконференцсвязи, кодированные аудиоданные и кодированные видеоданные не мультимплексируются. В этой ситуации кодированные данные передаются на видеодекодер 3206 и аудиодекодер 3208 без использования модуля 3204 демультимплексирования.

Посредством обработки демультимплексирования генерируются элементарный видеопоток (ES), аудио ES и, необязательно, субтитры. Видеодекодер 3206, который включает в себя видеодекодер 30, как объяснено в вышеупомянутых вариантах осуществления, декодирует видео ES с помощью способа декодирования, как показано в вышеупомянутых вариантах осуществления, для генерации видеокадра и подает эти данные в блок 3212 синхронизации. Аудиодекодер 3208 декодирует аудио ES для генерации аудиокадра и подает эти данные в блок 3212 синхронизации. В качестве альтернативы видеокадры могут сохраняться в буфере (не показан на фиг. 9) перед

подачей его в блок 3212 синхронизации. Точно так же аудиокадр может сохраняться в буфере (не показан на фиг. Y) перед подачей его в блок 3212 синхронизации.

Блок 3212 синхронизации синхронизирует видеокадр и аудиокадр и подает видео/аудио на видео/аудио дисплей 3214. Например, блок 3212 синхронизации синхронизирует представление видео и аудио информации. Информация может закодироваться в синтаксисе с использованием меток времени, касающихся представления кодированных аудио- и визуальных данных, и меток времени, касающихся доставки самого потока данных.

Если субтитры включены в поток, декодер 3210 субтитров декодирует субтитры и синхронизирует их с видеокадром и аудиокадром и передает видео/аудио/субтитры на дисплей 3216 видео/аудио/субтитров.

Настоящее изобретение не ограничивается вышеупомянутой системой, и либо устройство кодирования изображения, либо устройство декодирования изображения в вышеупомянутых вариантах осуществления может быть включено в другую систему, например, автомобильную систему.

Хотя варианты осуществления изобретения были в основном описаны на основе кодирования видео, следует отметить, что варианты осуществления системы 10 кодирования, кодера 20 и декодера 30 (и, соответственно, системы 10) и другие варианты осуществления, описанные в данном документе, также могут быть сконфигурированы для обработки или кодирования неподвижного изображения, то есть обработки или кодирования отдельного изображения независимо от любого предыдущего или последующего изображения, как при кодировании видео. В общем, только блоки 244 (кодер) и 344 (декодер) внешнего предсказания могут быть недоступны в случае, если кодирование обработки изображения ограничено одним изображением 17. Все другие функции (также называемые инструментами или технологиями) видеокодера 20 и видеodeкодера 30 могут в равной степени использоваться для обработки статических изображений, например вычисление остатка 204/304, преобразование 206, квантование 208, обратное квантование 210/310, (обратное) преобразование 212/312, разделение 262/362, внутреннее предсказание 254/354 и/или контурная фильтрация 220, 320, энтропийное кодирование 270 и энтропийное декодирование 304.

Варианты осуществления, например, кодера 20 и декодера 30, и функции, описанные в данном документе, например, со ссылкой на кодер 20 и декодер 30, могут быть реализованы в аппаратных средствах, программном обеспечении, встроенном программном обеспечении или любой их комбинации. Если реализованы в программном обеспечении, функции могут храниться на машиночитаемом носителе или передаваться через среду связи в виде одной или нескольких инструкций или кода и выполняться аппаратным процессором. Машиночитаемый носитель может включать в себя машиночитаемый носитель данных, который соответствует материальному носителю, например, носителю данных или носителю связи, включая любой носитель, который облегчает передачу компьютерной программы из одного места в другое, например, в соответствии с протоколом связи. Таким образом, машиночитаемые носители обычно могут соответствовать (1) материальным машиночитаемым носителям данных, которые являются энергонезависимыми, или (2) средам связи, таким как сигнал или несущая волна. Носители данных могут быть любыми доступными носителями, к которым может получить доступ один или несколько компьютеров или один или несколько процессоров для извлечения инструкций, кода и/или структур данных для реализации способов, описанных в этом раскрытии. Компьютерный программный продукт может включать в себя машиночитаемый носитель.

В качестве примера, но не ограничения, такие машиночитаемые носители данных могут содержать RAM, ROM, EEPROM, CD-ROM или другое запоминающее устройство на оптическом диске, запоминающее устройство на магнитном диске или другие магнитные запоминающие устройства, флэш-память или любой другой носитель, который может использоваться для хранения желаемого программного кода в форме инструкций или структур данных и может быть доступен для компьютера. Кроме того, любое соединение правильно называть машиночитаемым носителем. Например, если инструкции передаются с веб-сайта, сервера или другого удаленного источника с использованием коаксиального кабеля, оптоволоконного кабеля, витой пары, цифровой абонентской линии (DSL) или беспроводных технологий, таких как инфракрасный порт, радио и микроволновая печь, тогда коаксиальный кабель, оптоволоконный кабель, витая пара, DSL или беспроводные технологии, такие как инфракрасный порт, радио и микроволны включены в определение среды. Однако следует понимать, что машиночитаемые носители данных и носители данных не включают в себя соединения, несущие волны, сигналы или другие временные носители, а вместо этого направлены на энергонезависимые материальные носители данных. Диск и диск, используемые здесь, включают в себя компакт-диск (CD), лазерный диск, оптический диск, цифровой универсальный диск (DVD), дискету и диск Blu-ray, где диски обычно воспроизводят данные магнитным способом, и диски воспроизводят данные оптически с помощью лазеров. Комбинации вышеперечисленного также должны быть включены в объем машиночитаемых носителей.

Инструкции могут выполняться одним или несколькими процессорами, такими как один или несколько процессоров цифровых сигналов (DSP), микропроцессоры общего назначения, специализированные интегральные схемы (ASIC), программируемые логические матрицы (FPGA) или другие эквивалентные интегральные или дискретные логические схемы. Соответственно, используемый здесь термин «процессор» может относиться к любой из вышеупомянутой структуры или любой другой структуре, подходящей для реализации описанных здесь методик. Кроме того, в некоторых аспектах описанные здесь функциональные возможности могут быть предоставлены в рамках выделенных аппаратных и/или программных модулей, сконфигурированных для кодирования и декодирования или включенных в комбинированный кодек. Кроме того, способы могут быть полностью реализованы в одной или нескольких схемах или логических элементах.

Способы этого раскрытия могут быть реализованы в большом количестве устройств или аппаратур, в том числе беспроводной телефон, интегральную схему (IC) или набор IC (например, набор микросхем). В этом раскрытии описаны различные компоненты, модули или блоки, чтобы подчеркнуть функциональные аспекты устройств, сконфигурированных для выполнения раскрытых технологий, но не обязательно требующих реализации различными аппаратными блоками. Скорее, как описано выше, различные блоки могут быть объединены в аппаратный блок кодека или предоставлены набором взаимодействующих аппаратных блоков, в том числе один или несколько процессоров, как описано выше, в сочетании с подходящим программным обеспечением и/или встроенным программным обеспечением.

(57) Формула изобретения

1. Способ внутреннего предсказания выборки цветности блока путем применения кросс-компонентной линейной модели, содержащий:  
получение восстановленных выборок яркости;

определение максимальных и минимальных значений выборок яркости на основе восстановленных выборок яркости;

получение разности максимального и минимального значений выборок яркости;

определение положения старшего бита разности максимального и минимального значений выборок яркости;

извлечение значения из справочной таблицы (LUT) с использованием набора битов в качестве индекса, причем набор битов следует за положением старшего бита разности максимального и минимального значений выборок яркости;

получение параметров  $\alpha$  и  $\beta$  линейной модели на основе извлеченного значения; и вычисление предсказанного значения выборки цветности с использованием полученных параметров  $\alpha$  и  $\beta$  линейной модели.

2. Способ по п. 1, в котором положение старшего бита разности максимального и минимального значений выборок яркости получают как логарифм упомянутой разности по основанию два.

3. Способ по п. 1 или 2, в котором определение набора битов, следующих за положением старшего бита разности, приводит к набору битов, содержащему четыре бита.

4. Способ по любому из пп. 1-3, в котором старший бит является первым ненулевым битом.

5. Способ по любому из пп. 1-4, причем способ содержит: получение параметров  $\alpha$  и  $\beta$  линейной модели на основе извлеченного значения и разности максимального и минимального значений восстановленных выборок цветности.

6. Способ по п. 5, при этом способ содержит:

получение параметров  $\alpha$  и  $\beta$  линейной модели путем умножения извлеченного значения на разность максимального и минимального значений восстановленных выборок цветности.

7. Способ по любому из пп. 1-6, в котором LUT содержит по меньшей мере три значения: первое значение, второе значение и третье значение;

среди трех значений первое значение и второе значение являются двумя соседними значениями, и второе значение и третье значение являются двумя соседними значениями.

8. Способ по п. 7, в котором шаг между первым значением и вторым значением равен шагу между вторым значением и третьим значением.

9. Способ по п. 7, в котором шаг между первым значением и вторым значением отличается от шага между вторым значением и третьим значением.

10. Способ по любому из пп. 7-9, в котором первое значение индексируется первым набором битов, а второе значение индексируется вторым набором битов;

когда значение первого набора битов больше, чем значение второго набора битов, первое значение меньше второго значения; или

когда значение первого набора битов меньше значения второго набора битов, первое значение больше второго значения.

11. Способ по любому из пп. 1-10, в котором LUT содержит поддиапазоны значений, а шаг любых двух соседних значений является постоянным в пределах одного поддиапазона.

12. Устройство для внутреннего предсказания выборки цветности блока путем применения кросс-компонентной линейной модели, при этом устройство является кодером или декодером, и устройство содержит:

блок получения, сконфигурированный для получения восстановленных выборок яркости;

блок определения, сконфигурированный для определения максимальных и минимальных значений выборок яркости на основе восстановленных выборок яркости;  
 блок получения, дополнительно сконфигурированный для получения разности максимального и минимального значений выборок яркости;

5 блок определения, дополнительно сконфигурированный для определения положения старшего бита разности максимального и минимального значений выборок яркости;  
 и

10 блок вычисления, сконфигурированный для извлечения значения из справочной таблицы (LUT) с использованием набора битов в качестве индекса, причем набор битов следует за положением старшего бита разности максимального и минимального значений выборок яркости, получения параметров  $\alpha$  и  $\beta$  линейной модели на основе извлеченного значения; и вычисления предсказанного значения выборки цветности с использованием полученных параметров  $\alpha$  и  $\beta$  линейной модели.

15 13. Устройство по п. 12, в котором положение старшего бита разности максимального и минимального значений выборки яркости является логарифмом упомянутой разности по основанию два.

14. Устройство по п. 12 или 13, в котором набор битов содержит четыре бита.

15. Устройство по любому из пп. 12-14, в котором старший бит является первым ненулевым битом.

20 16. Устройство по любому из пп. 12-15, в котором блок вычисления сконфигурирован для: получения параметров  $\alpha$  и  $\beta$  линейной модели на основе извлеченного значения и разности максимального и минимального значений восстановленных выборок цветности.

25 17. Устройство по п. 16, в котором блок вычисления выполнен с возможностью: получить параметры  $\alpha$  и  $\beta$  линейной модели, умножив извлеченное значение на разность максимального и минимального значений восстановленных выборок цветности.

18. Устройство по любому из пп. 12-17, причем LUT содержит поддиапазоны значений, и шаг любых двух соседних значений является постоянным в одном поддиапазоне.

19. Кодер, содержащий схему обработки для выполнения способа по любому из пп. 1-11.

30 20. Декодер, содержащий схему обработки для выполнения способа по любому из пп. 1-11.

21. Машиночитаемый носитель данных, содержащий программный код для выполнения способа по любому из пп. 1-11.

22. Декодер, содержащий:

35 один или несколько процессоров; и

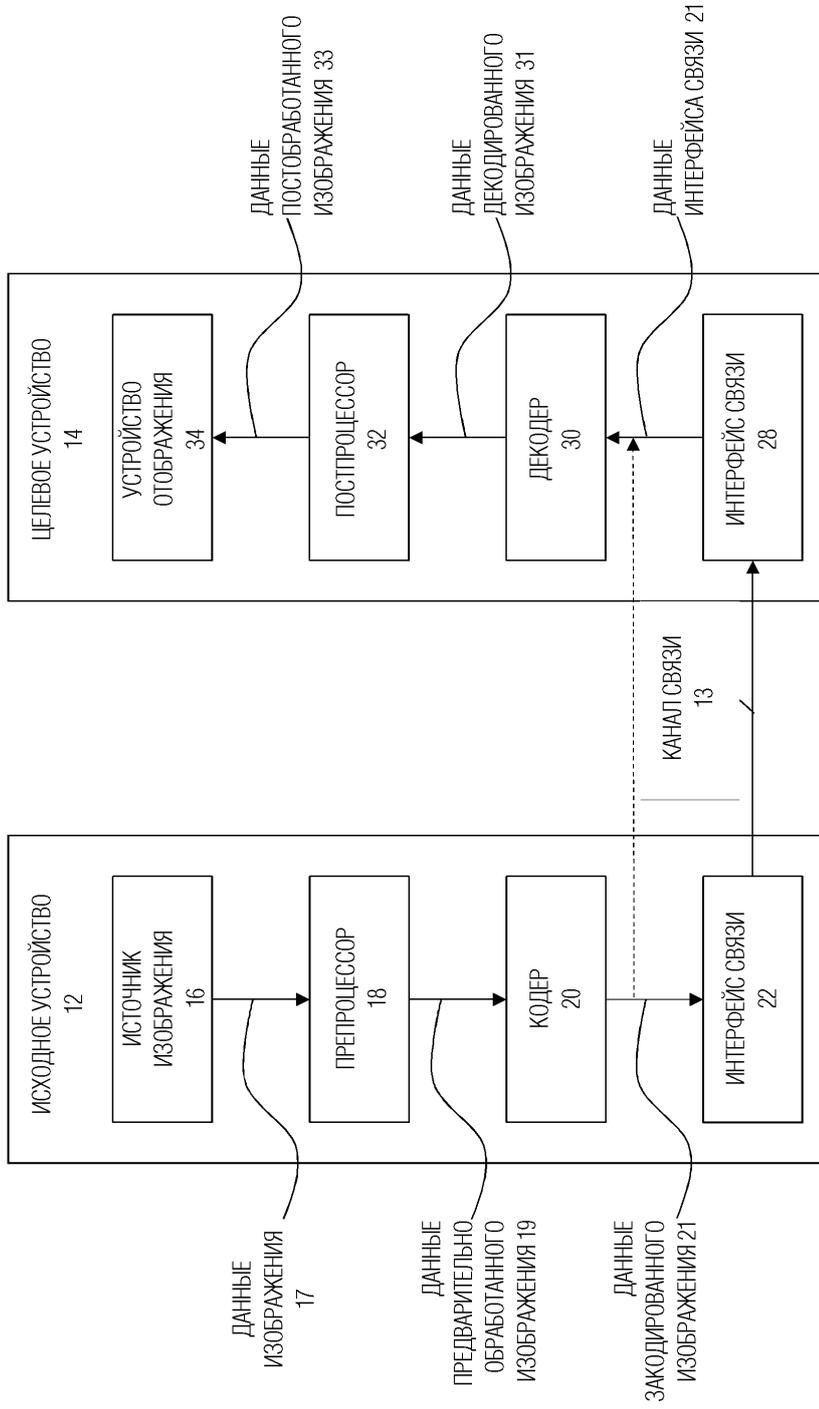
энергонезависимый машиночитаемый носитель данных, подсоединенный к процессорам и хранящий программу для выполнения процессорами, причем программа, когда выполняется процессорами, конфигурирует декодер для выполнения способа по любому из пп. 1-11.

40 23. Кодер, содержащий:

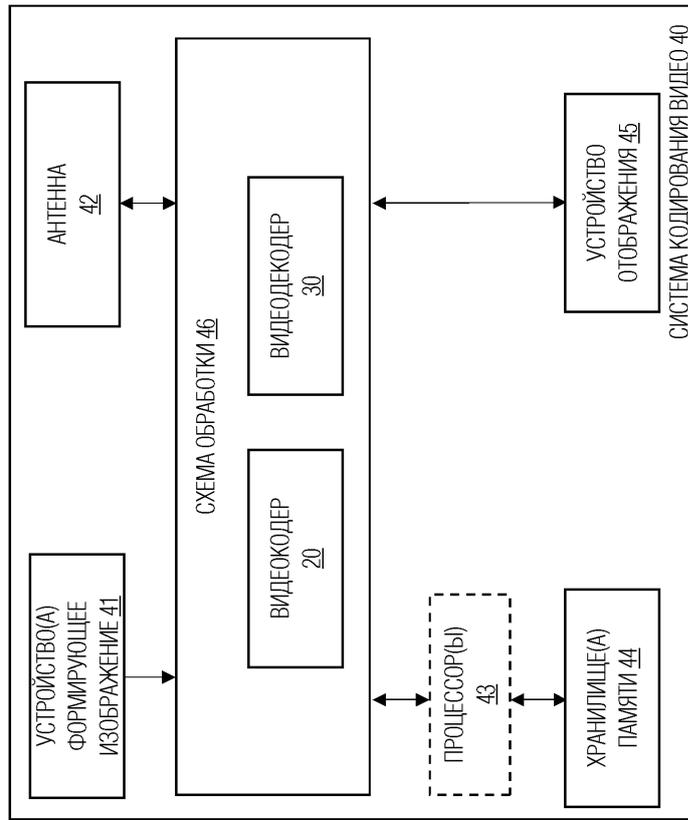
один или несколько процессоров; и

45 энергонезависимый машиночитаемый носитель данных, подсоединенный к процессорам и хранящий программу для выполнения процессорами, причем программа, когда выполняется процессорами, конфигурирует кодер для выполнения способа по любому из пп. 1-11.

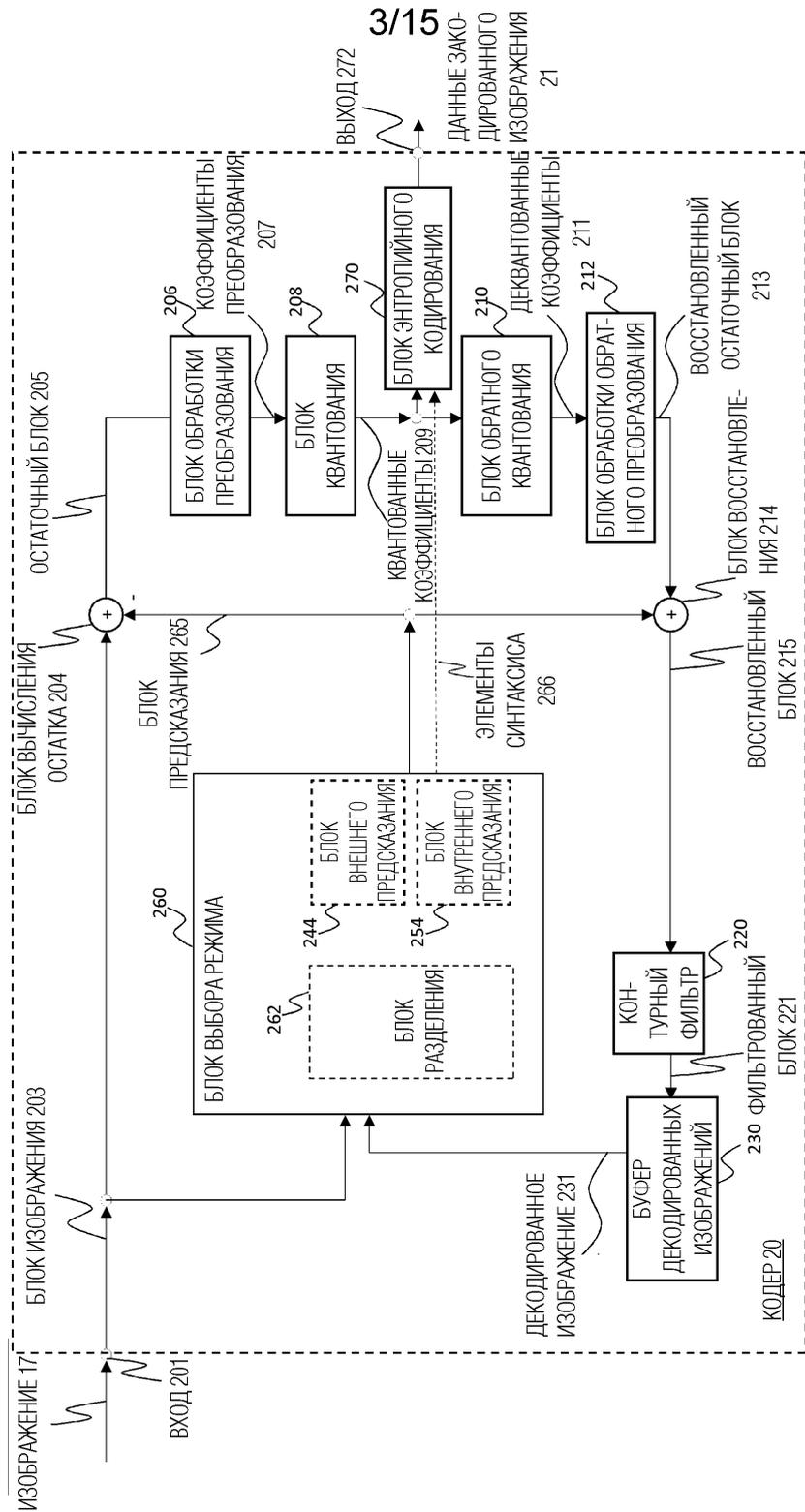
24. Энергонезависимый носитель записи, который включает в себя кодированный с предсказанием битовый поток, декодированный устройством, причем битовый поток сгенерирован согласно любому из пп. 1-11.



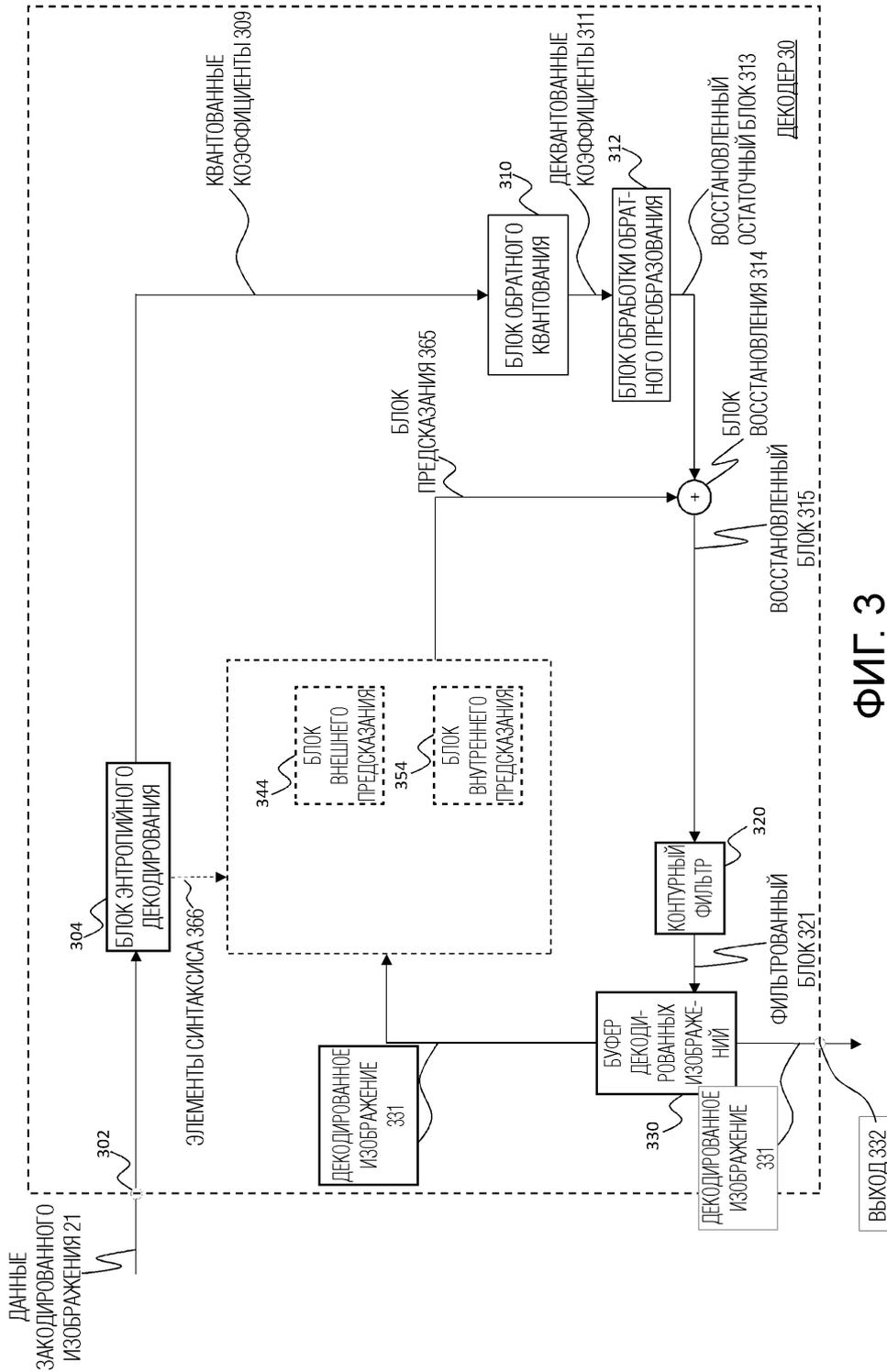
ФИГ. 1А



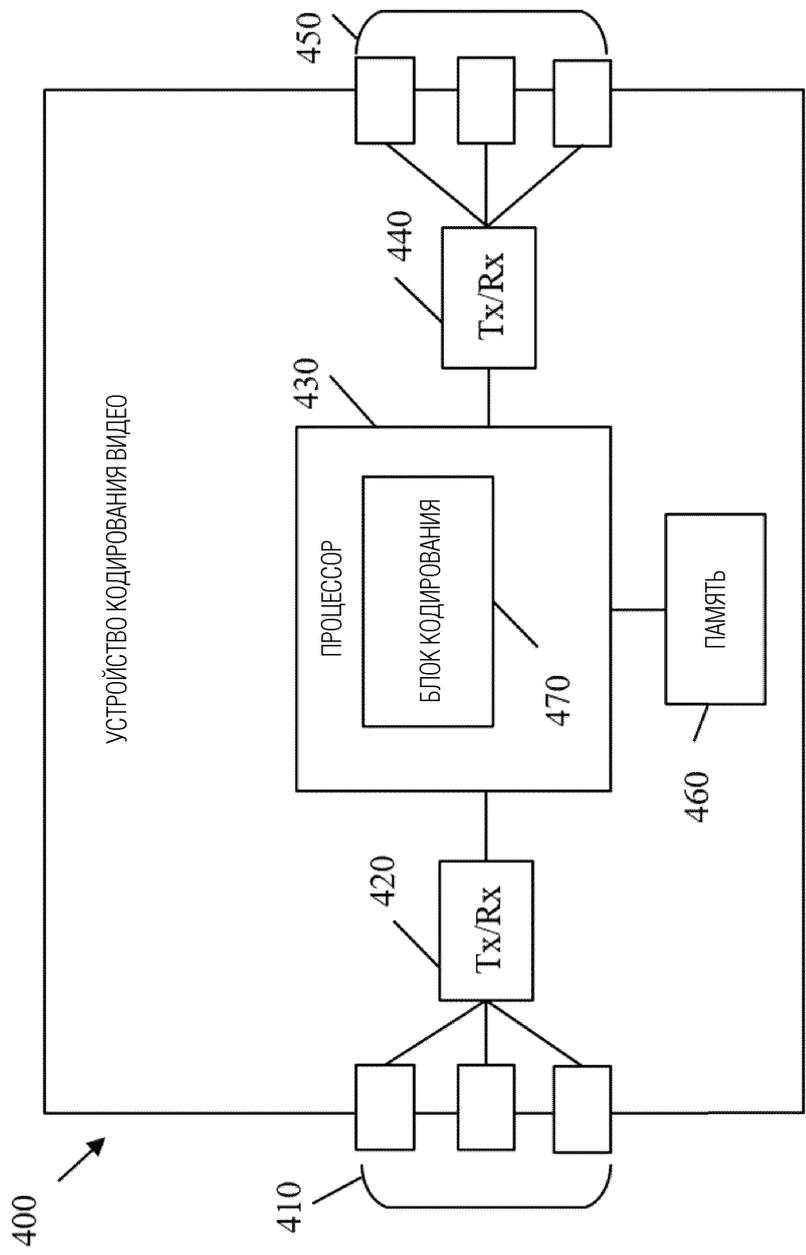
ФИГ. 1В



ФИГ. 2



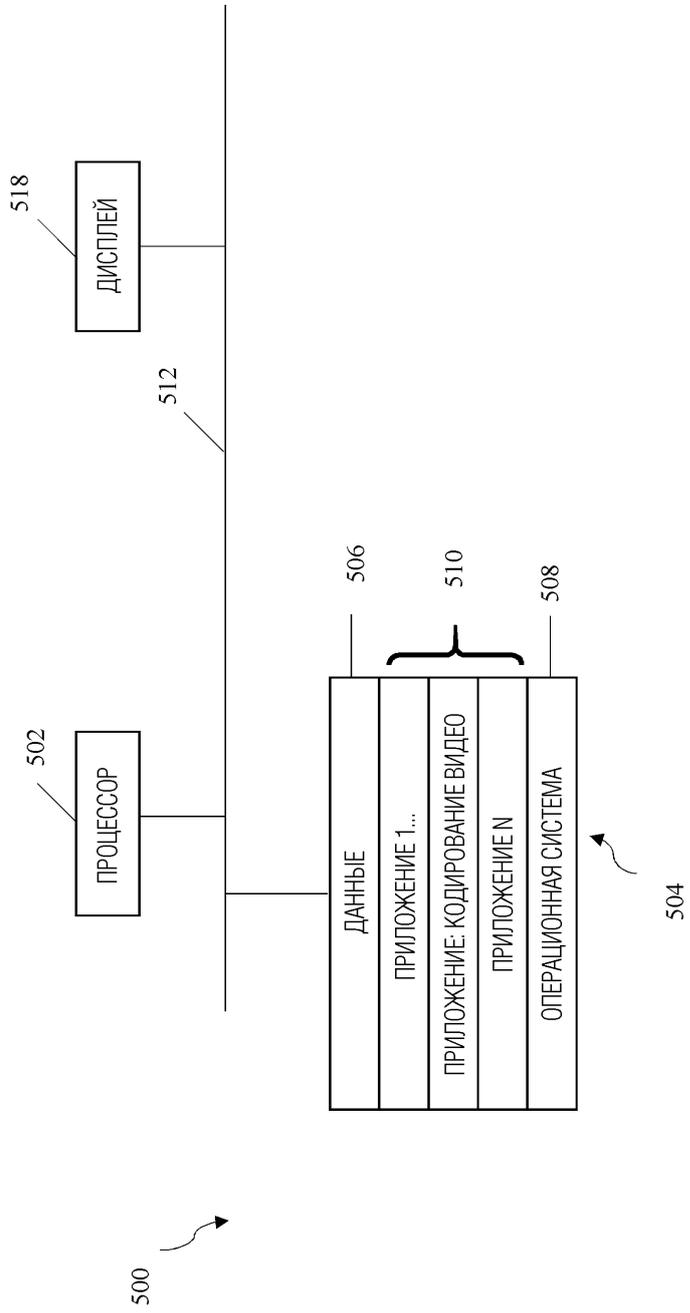
ФИГ. 3



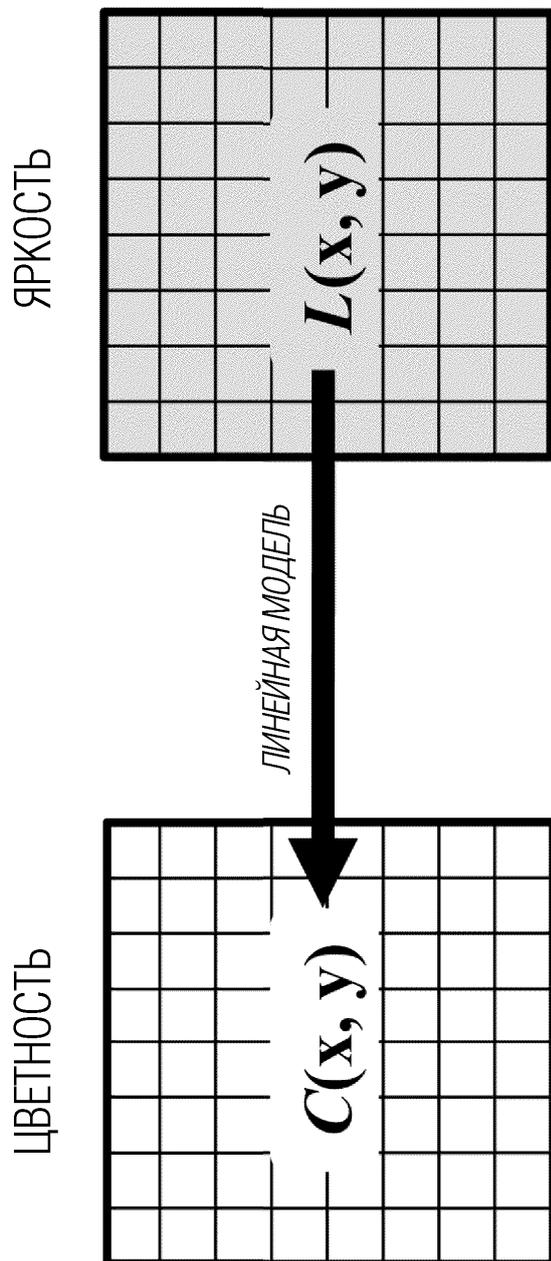
ВОСХОДЯЩИЕ ПОРТЫ

ФИГ. 4

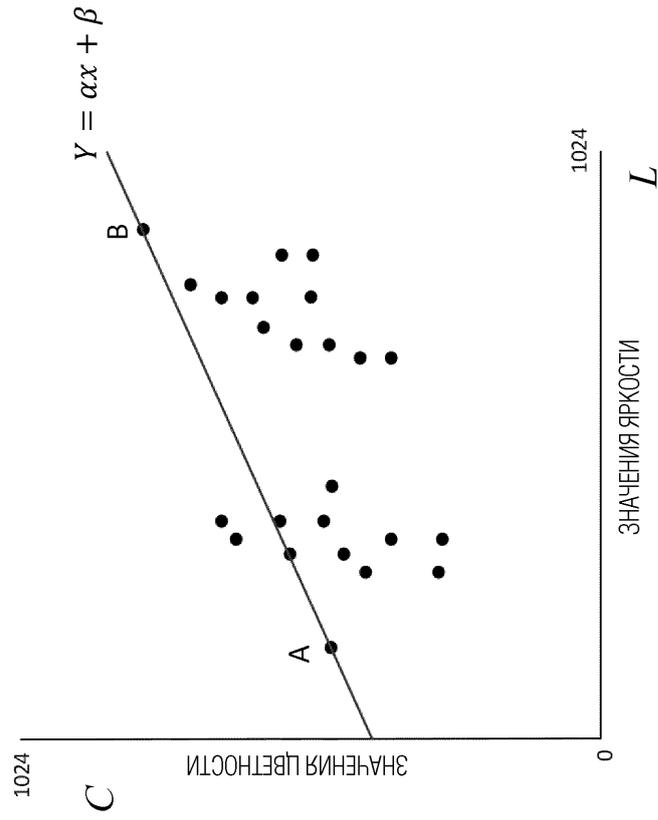
НИСХОДЯЩИЕ ПОРТЫ



ФИГ. 5

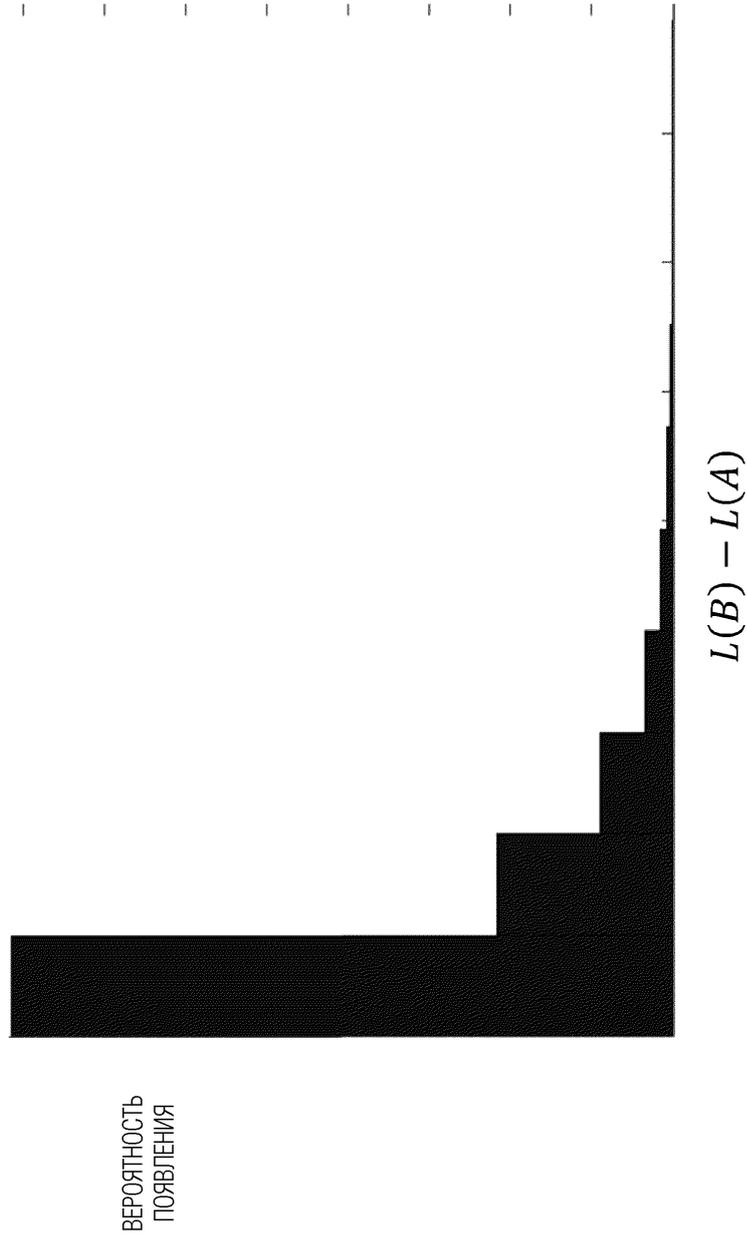


ФИГ. 6



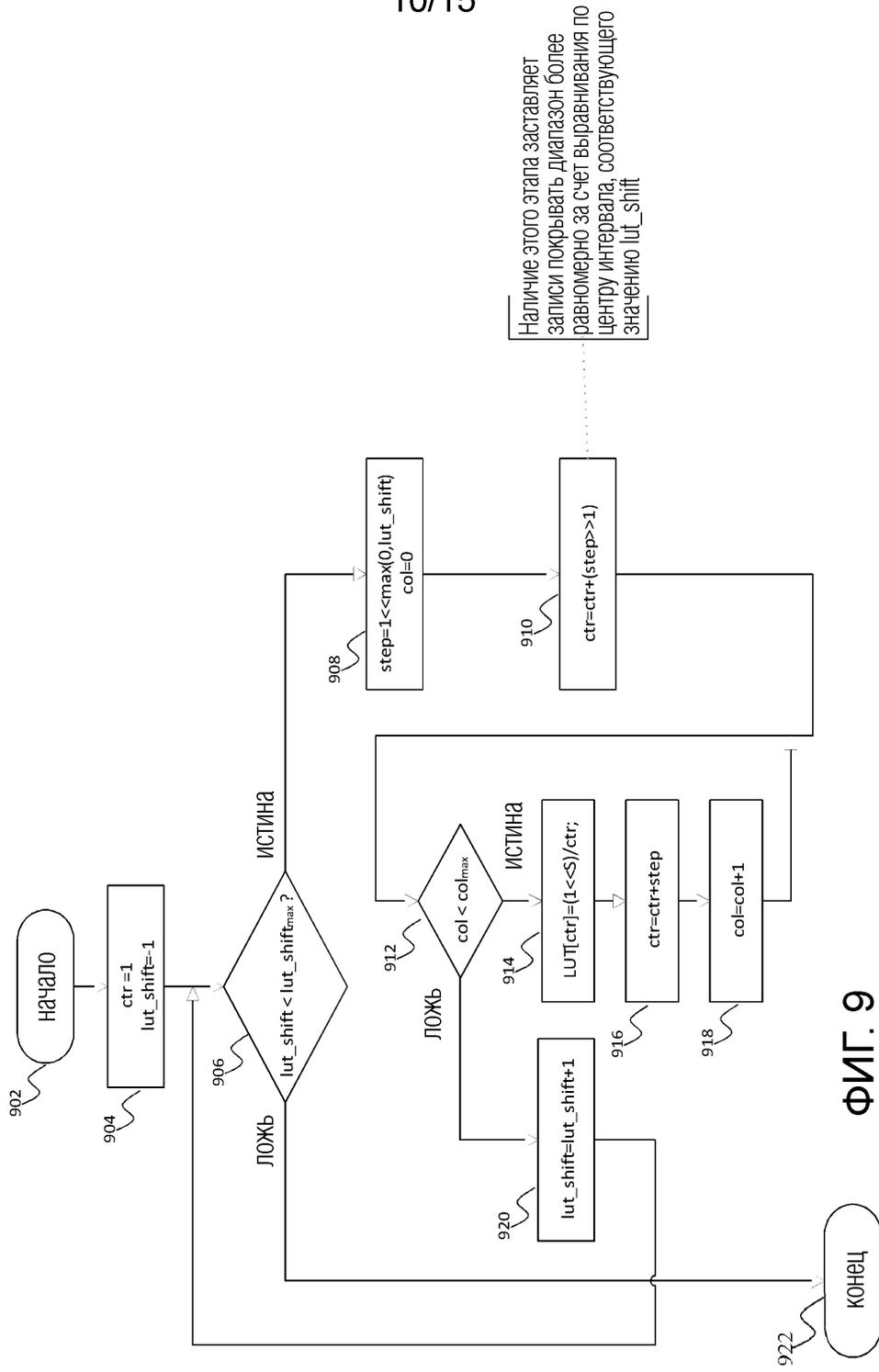
ФИГ. 7

9/15

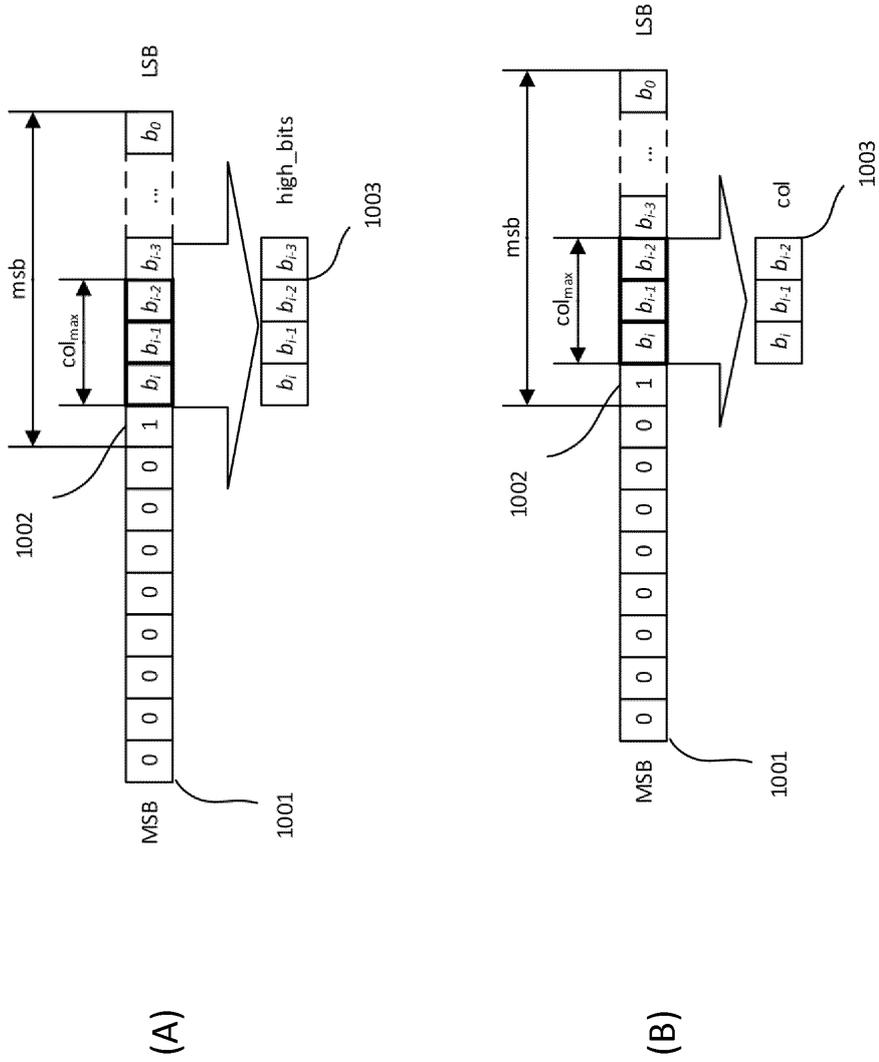


ФИГ. 8

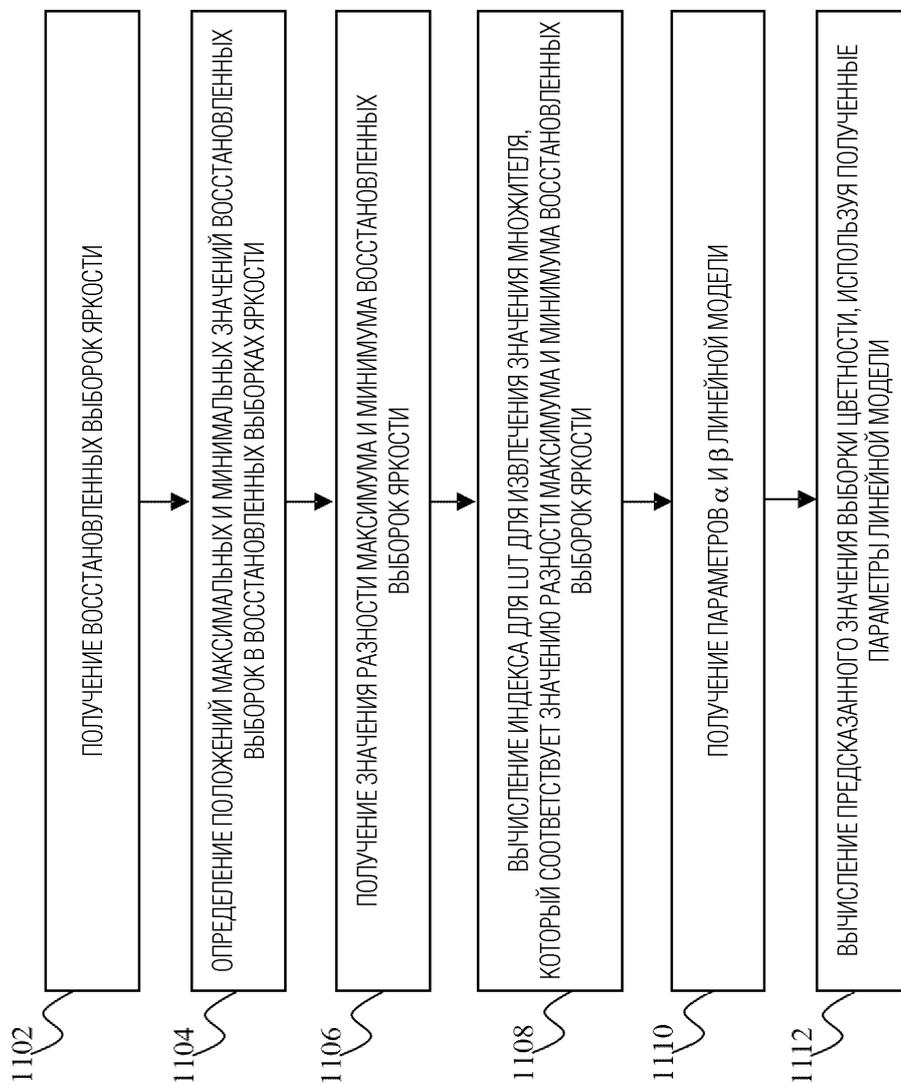
10/15



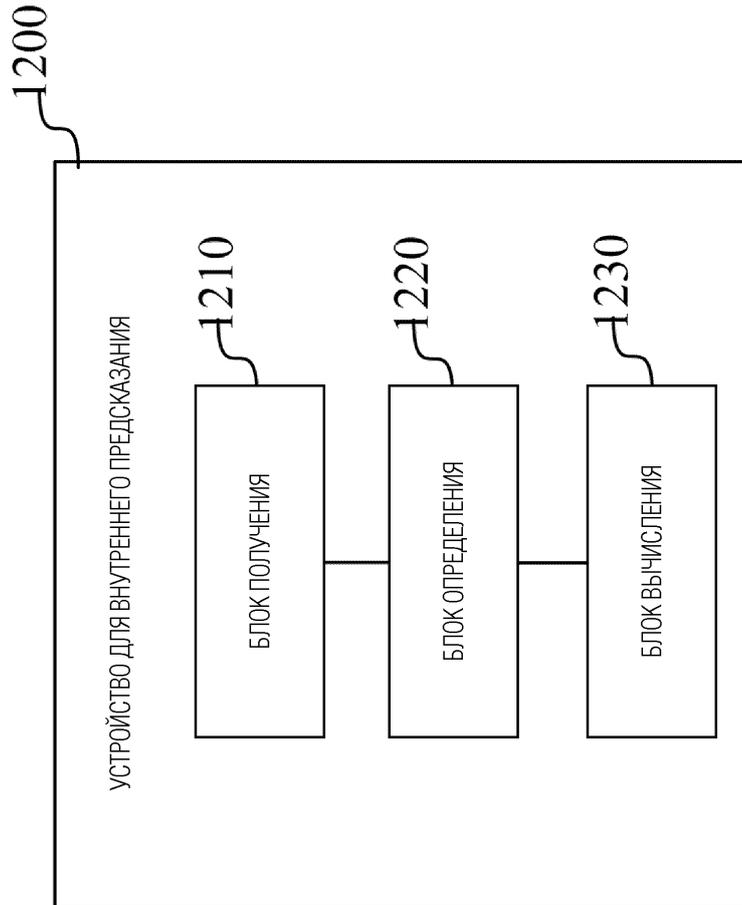
ФИГ. 9



ФИГ. 10

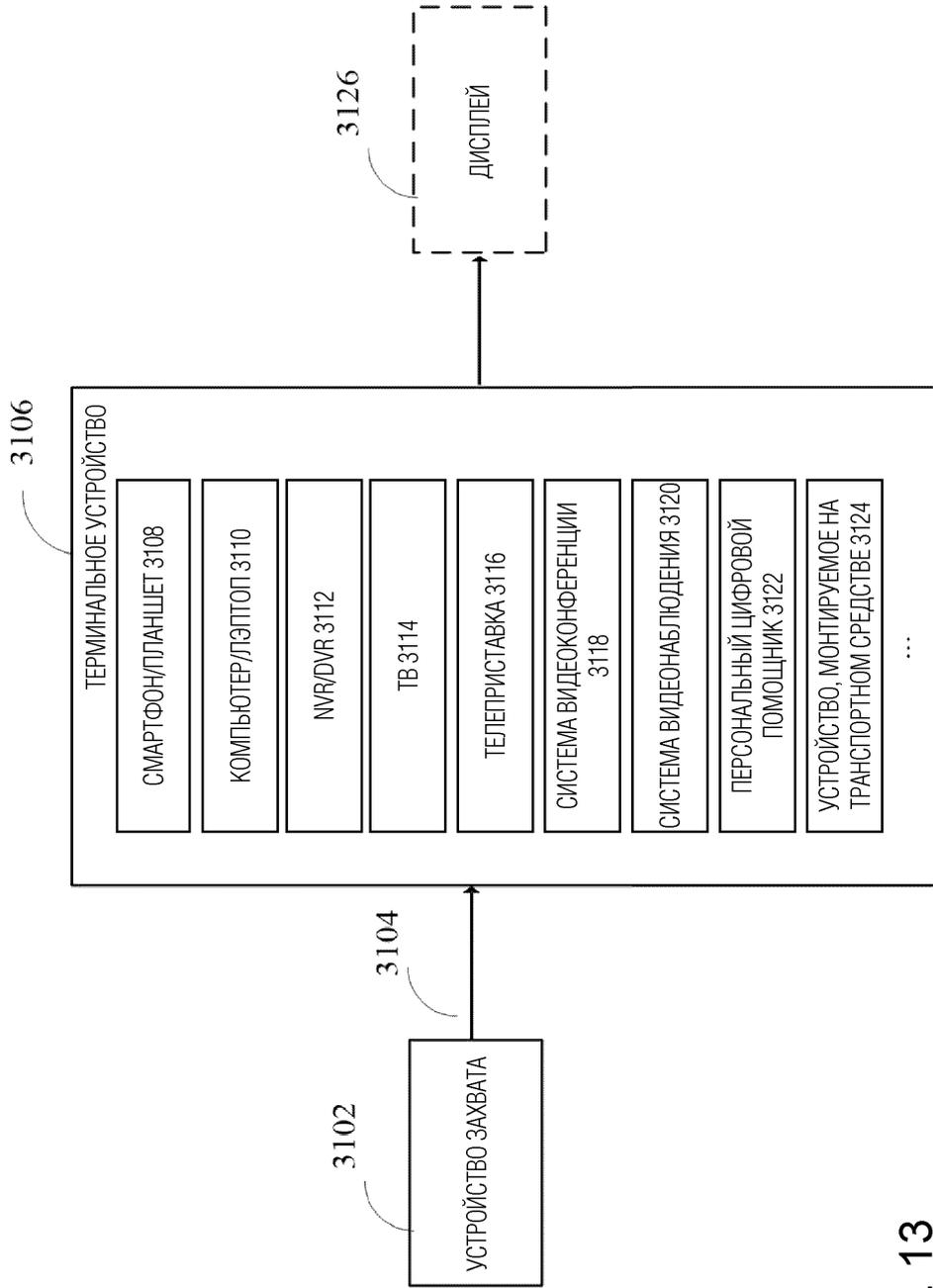


ФИГ. 11



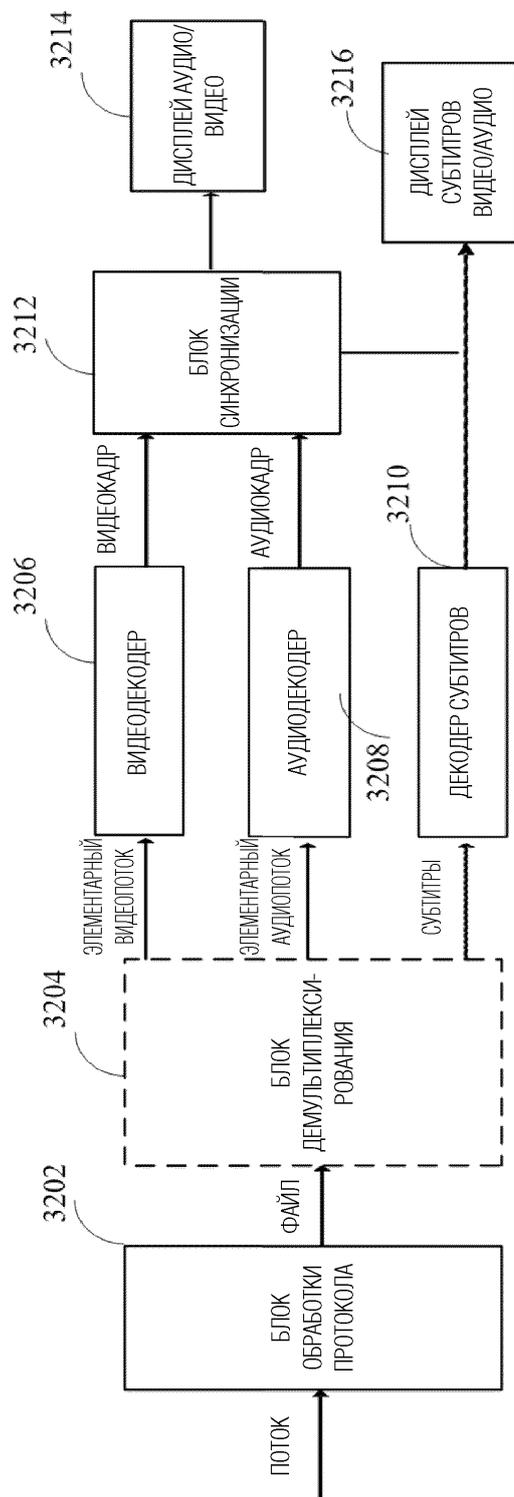
ФИГ. 12

14/15



ФИГ. 13

15/15



ФИГ. 14