



(51) МПК

H04N 19/80 (2014.01)

H04N 19/90 (2014.01)

H04N 19/186 (2014.01)

H04N 19/176 (2014.01)

G06T 7/90 (2017.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

## (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

H04N 19/80 (2018.08); H04N 19/90 (2018.08); H04N 19/186 (2018.08); H04N 19/176 (2018.08); G06T 7/90 (2018.08); H04N 19/117 (2018.08); H04N 19/124 (2018.08); H04N 19/567 (2018.08); H04N 19/70 (2018.08); H04N 19/85 (2018.08); H04N 19/94 (2018.08); G06T 5/40 (2018.08); G06T 7/44 (2018.08); G06T 9/00 (2018.08)

(21)(22) Заявка: 2016137792, 26.03.2015

(24) Дата начала отсчета срока действия патента:  
26.03.2015

Дата регистрации:  
29.01.2019

Приоритет(ы):

(30) Конвенционный приоритет:  
26.03.2014 US 61/970,838;  
25.03.2015 US 14/667,900

(43) Дата публикации заявки: 26.04.2018 Бюл. № 12

(45) Опубликовано: 29.01.2019 Бюл. № 4

(85) Дата начала рассмотрения заявки РСТ на  
национальной фазе: 26.10.2016

(86) Заявка РСТ:  
US 2015/022771 (26.03.2015)

(87) Публикация заявки РСТ:  
WO 2015/148814 (01.10.2015)

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

(72) Автор(ы):

ЦЗОУ Фэн (US),  
ПУ Вэй (US),  
ДЖОШИ Раджан Лаксман (US),  
СОЛЕ РОХАЛЬС Хоэль (US),  
КАРЧЕВИЧ Марта (US)

(73) Патентообладатель(и):

КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

(56) Список документов, цитированных в отчете  
о поиске: JIANLE CHEN et al. "Description  
of screen content coding technology proposal  
by qualcomm", Joint Collaborative Team on  
Video Coding (JCT-VC) of ITU-T SG16 WP3  
and ISO/IEC JTC1/SC29/WG11, 17th Meeting:  
Valencia, ES, 27 March - 4 April 2014,  
Document: JCTVC-Q0031, 19 л., опубл.  
28.03.2014. WEI PU et al. "ANG10:  
Simplification of palette based (см. прод.)

## (54) ОПРЕДЕЛЕНИЕ РАЗМЕРА ПАЛИТРЫ, ЗАПИСЕЙ ПАЛИТРЫ И ФИЛЬТРОВАНИЕ БЛОКОВ, КОДИРОВАННЫХ НА ОСНОВЕ ПАЛИТРЫ, ПРИ КОДИРОВАНИИ ВИДЕО

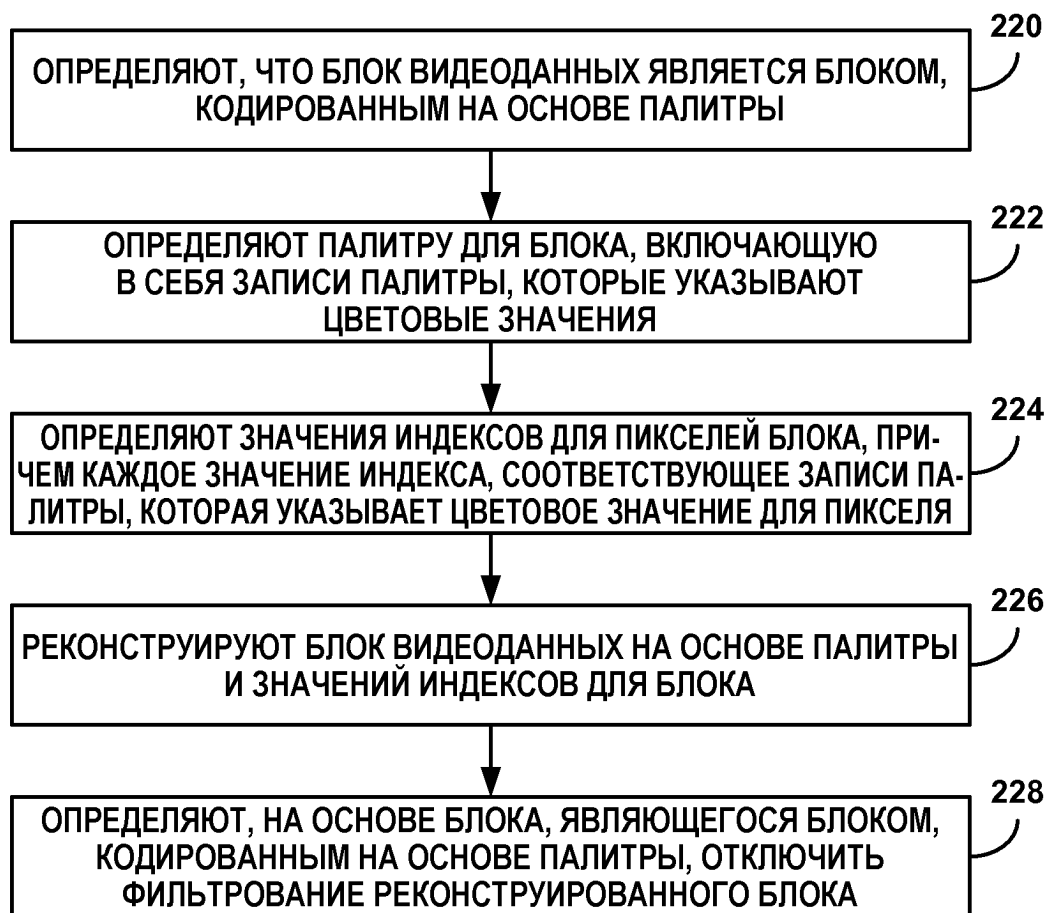
(57) Реферат:

Изобретение относится к области кодирования/декодирования видеоданных на основе палитры. Технический результат – повышение эффективности кодирования/декодирования видео за счет использования полученной палитры. Способ обработки

видеоданных содержит этапы, на которых: определяют, что блок видеоданных является блоком, кодированным на основе палитры; определяют палитру для блока, включающую в себя одну или более записей палитры, которые указывают одно или более соответственных

цветовых значений; определяют значения индексов для одного или более пикселей блока, причем каждое из значений индексов соответствует одной из записей палитры, которая указывает цветовое значение для одного из пикселей блока; реконструируют блок

видеоданных на основе палитры и значений индексов для блока и определяют на основе блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока. 4 н. и 25 з.п. ф-лы, 6 ил.



ФИГ. 5

(56) (продолжение):

coding", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 17th Meeting: Valencia, ES, 27 March - 4 April 2014, Document: JCTVC-Q0047, 4 л., опубли. 17.03.2014. US 2013/0188733 A1, 25.07.2013. US 2014/0064612 A1, 06.03.2014. US 2009/0074307 A1, 19.03.2009. RU 2420021 C2, 27.05.2011.



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.

*H04N 19/80* (2014.01)*H04N 19/90* (2014.01)*H04N 19/186* (2014.01)*H04N 19/176* (2014.01)*G06T 7/90* (2017.01)**(12) ABSTRACT OF INVENTION**

(52) CPC

*H04N 19/80* (2018.08); *H04N 19/90* (2018.08); *H04N 19/186* (2018.08); *H04N 19/176* (2018.08); *G06T 7/90* (2018.08); *H04N 19/117* (2018.08); *H04N 19/124* (2018.08); *H04N 19/567* (2018.08); *H04N 19/70* (2018.08); *H04N 19/85* (2018.08); *H04N 19/94* (2018.08); *G06T 5/40* (2018.08); *G06T 7/44* (2018.08); *G06T 9/00* (2018.08)

(21)(22) Application: **2016137792, 26.03.2015**

(24) Effective date for property rights:  
**26.03.2015**

Registration date:  
**29.01.2019**

Priority:

(30) Convention priority:  
**26.03.2014 US 61/970,838;**  
**25.03.2015 US 14/667,900**

(43) Application published: **26.04.2018** Bull. № 12(45) Date of publication: **29.01.2019** Bull. № 4(85) Commencement of national phase: **26.10.2016**

(86) PCT application:  
**US 2015/022771 (26.03.2015)**

(87) PCT publication:  
**WO 2015/148814 (01.10.2015)**

Mail address:

**129090, Moskva, ul. B.Spasskaya, 25, stroenie 3,**  
**OOO "Yuridicheskaya firma Gorodisskij i**  
**Partnery"**

(72) Inventor(s):

**ZOU Feng (US),**  
**PU Wei (US),**  
**JOSHI Rajan Laxman (US),**  
**SOLE ROJALS Joel (US),**  
**KARCZEWICZ Marta (US)**

(73) Proprietor(s):

**QUALCOMM INCORPORATED**  
**INTERNATIONAL IP ADMINISTRATION**  
**GROUP (US)**

**(54) DETERMINING PALETTE SIZE, PALETTE ENTRIES AND FILTERING OF PALETTE CODED BLOCKS IN VIDEO CODING**

(57) Abstract:

FIELD: data processing.

SUBSTANCE: invention relates to palette-based video encoding/decoding. Method for processing video data comprises steps of: determining that a block of video data is a palette coded block; determining a palette for the block including one or more palette entries that indicate one or more respective colour values; determining index values for one or more pixels of the

block, each of the index values corresponding to one of the palette entries that indicates a colour value for one of the pixels of the block; reconstructing the block of video data based on the palette and the index values for the block and determining, based on the block being a palette coded block, to disable filtering of the reconstructed block.

EFFECT: high efficiency of video encoding/



ФИГ. 5

[0001] Эта заявка испрашивает приоритет по предварительной заявке США № 61/970,838, поданной 26 марта, 2014, все содержимое которой включено в настоящий документ посредством ссылки.

## **ОБЛАСТЬ ТЕХНИКИ, К КОТОРОЙ ОТНОСИТСЯ ИЗОБРЕТЕНИЕ**

[0002] Это раскрытие относится к кодированию и декодированию видео.

## **УРОВЕНЬ ТЕХНИКИ**

[0003] Возможности цифрового видео могут быть встроены в широкий диапазон устройств, в том числе в цифровые телевизоры, системы цифрового прямого широко вещания, беспроводные широко вещательные системы, карманные персональные компьютеры (PDA), портативные или настольные компьютеры, планшетные компьютеры, устройства чтения электронных книг, цифровые камеры, цифровые записывающие устройства, проигрыватели цифровых медиаданных, устройства видеоигр, видеоигровые консоли, сотовые или спутниковые радиотелефоны, также называемые смартфонами ("интеллектуальными телефонами"), устройства видеоконференцсвязи, устройства потокового видео и тому подобные. Устройства цифрового видео реализуют способы сжатия видео, такие как описанные в стандартах, заданных посредством MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.188/MPEG-4, Часть 10, Улучшенного кодирования видео (AVC), находящегося в настоящее время в разработке стандарта Высокоэффективного кодирования видео (HEVC), и расширениях этих стандартов. Видеоустройства могут передавать, принимать, кодировать, декодировать и/или хранить цифровую видеоинформацию более эффективно посредством реализации таких способов сжатия видео.

[0004] Способы сжатия видео выполняют пространственное (внутри (intra) изображения) предсказание и/или временное (между (inter) изображениями) предсказание для снижения или удаления избыточности, свойственной видеопоследовательностям. Для блочного кодирования видео, слайс (вырезка) (т.е. видеокадр или участок видеокадра) видео может быть разбит на видеоблоки. Видеоблоки в кодированном посредством внутреннего кодирования (intra-кодированном) (I) слайсе изображения кодируются с использованием пространственного предсказания относительно опорных выборок в соседних блоках в том же изображении. Видеоблоки в кодированном посредством внешнего кодирования (inter-кодированном) (P или B) слайсе изображения могут использовать пространственное предсказание относительно опорных выборок в соседних блоках в том же изображении или временное предсказание относительно опорных выборок в других опорных изображениях. Изображения могут быть названы кадрами, и опорные изображения могут быть названы опорными кадрами.

[0005] Пространственное или временное предсказание дает в результате предсказательный блок для блока, который должен быть кодирован. Остаточные данные представляют собой разности пикселей между первоначальным блоком, который должен быть кодирован, и предсказательным блоком. Блок, кодированный посредством внешнего кодирования, кодируется согласно вектору движения, который указывает на блок опорных выборок, образующих предсказательный блок, и остаточным данным, указывающим разность между кодированным блоком и предсказательным блоком. Блок, кодированный посредством внутреннего кодирования, кодируется согласно режиму внутреннего кодирования и остаточным данным. Для дальнейшего сжатия, остаточные данные могут быть преобразованы из пиксельной области в область преобразования, давая в результате остаточные коэффициенты, которые затем могут быть квантованы. Может быть осуществлено сканирование квантованных коэффициентов, изначально размещенных в двухмерном массиве, для того, чтобы

произвести одномерный вектор коэффициентов, и для достижения даже большего сжатия может быть применено энтропийное кодирование.

[0006] Битовый поток кодирования нескольких видов может быть сгенерирован посредством кодирования видов, например, с нескольких перспектив. Были разработаны некоторые стандарты трехмерного (3D) видео, которые пользуются аспектами кодирования нескольких видов. Например, разные виды могут передавать виды для левого и правого глаза для поддержки 3D видео. В качестве альтернативы, некоторые процессы кодирования 3D видео могут применять так называемое кодирование нескольких видов плюс глубины. При кодировании нескольких видов плюс глубины, битовый поток 3D видео может содержать не только компонент вида текстуры, но также компоненты вида глубины. Например, каждый вид может содержать один компонент вида текстуры и один компонент вида глубины.

### **СУЩНОСТЬ ИЗОБРЕТЕНИЯ**

[0007] В общем, это раскрытие описывает способы кодирования видео на основе палитры. При кодировании на основе палитры, (де)кодер видео (например, видеокодер или видеodeкодер) может сформировать так называемую "палитру", как таблицу цветов для представления видеоданных конкретной области (например, заданного блока). Кодирование на основе палитры может быть особенно полезно для областей кодирования видеоданных, имеющих относительно небольшое число цветов. Вместо кодирования фактических значений пикселей или их остатков для заданного блока, (де)кодер видео может кодировать значения индексов для одного или более пикселей. Значения индексов увязывают пиксели с записями в палитре, представляющей цвета пикселей. Палитра может быть явным образом кодирована и отправлена декодеру, предсказана из предыдущих записей палитры, или их комбинации. В этом раскрытии, описаны способы определения, отключить ли фильтрацию, такое как фильтрация для удаления блочности или фильтрация с адаптивным смещением выборки (SAO), для блоков, кодированных на основе палитры, в видеокодере или видеodeкодере. В дополнение, описаны способы изменения размера палитры и записей палитры для палитры в видеокодере по меньшей мере частично на основе издержек на оптимизацию по скорости (битрейту) - искажению.

[0008] В одном примере, это раскрытие направлено на способ обработки видеоданных, содержащий определение того, что блок видеоданных является блоком, кодированным на основе палитры; определение палитры для блока, включающей в себя одну или более записей палитры, которые указывают одно или более соответствующих цветовых значений; определение значений индексов для одного или более пикселей блока, причем каждое из значений индексов соответствует одной из записей палитры, которая указывает цветовое значение для одного из пикселей блока; реконструирование блока видеоданных на основе палитры и значений индексов для блока; и определение, на основе блока, являющегося блоком, кодированным на основе палитры, отключить ли фильтрацию реконструированного блока.

[0009] В другом примере, это раскрытие направлено на устройство обработки видео, содержащее память, выполненную с возможностью хранения видеоданных; и один или более процессоров, имеющих связь с памятью. Один или более процессоров выполнены с возможностью определения того, что блок видеоданных является блоком, кодированным на основе палитры; определения палитры для блока, включающей в себя одну или более записей палитры, которые указывают одно или более соответствующих цветовых значений; определения значений индексов для одного или более пикселей блока, причем каждое из значений индексов соответствует одной из

записей палитры, которая указывает цветовое значение для одного из пикселей блока; реконструирования блока видеоданных на основе палитры и значений индексов для блока; и определения, на основе блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока.

5 [0010] В дополнительном примере, это раскрытие направлено на устройство обработки видео, содержащее средство для определения того, что блок видеоданных является блоком, кодированным на основе палитры; средство для определения палитры для блока, включающей в себя одну или более записей палитры, которые указывают одно или более соответствующих цветовых значений; средство для определения значений  
10 индексов для одного или более пикселей блока, причем каждое из значений индексов соответствует одной из записей палитры, которая указывает цветовое значение для одного из пикселей блока; средство для реконструирования блока видеоданных на основе палитры и значений индексов для блока; и средство для определения, на основе блока, являющегося блоком, кодированным на основе палитры, отключить  
15 фильтрацию реконструированного блока.

[0011] В дополнительном примере, это раскрытие направлено на компьютерно-читаемый носитель, имеющий сохраненные в нем инструкции для обработки видеоданных, которые при исполнении предписывают одному или более процессорам определять, что блок видеоданных является блоком, кодированным на основе палитры;  
20 определять палитру для блока, включающую в себя одну или более записей палитры, которые указывают одно или более соответствующих цветовых значений; определять значения индексов для одного или более пикселей блока, причем каждое из значений индексов соответствует одной из записей палитры, которая указывает цветовое значение для одного из пикселей блока; реконструировать блок видеоданных на основе палитры  
25 и значений индексов для блока; и определять, на основе блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока.

[0012] Подробности одного или более примеров данного раскрытия изложены на прилагаемых чертежах и в описании ниже. Другие признаки, цели и преимущества будут  
30 понятны из описания, чертежей и формулы изобретения.

### **КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ**

[0013] Фиг. 1 является блок-схемой, иллюстрирующей примерную систему кодирования видео, которая может использовать способы, описанные в этом раскрытии.

[0014] Фиг. 2 является блок-схемой, иллюстрирующей примерный видеокодер,  
35 который может реализовать способы, описанные в этом раскрытии.

[0015] Фиг. 3 является блок-схемой, иллюстрирующей примерный видеодекодер, который может реализовать способы, описанные в этом раскрытии.

[0016] Фиг. 4 является принципиальной схемой, иллюстрирующей пример определения записей палитры для кодирования видео на основе палитры, соответствующей способам  
40 этого раскрытия.

[0017] Фиг. 5 является схемой последовательности операций, иллюстрирующей примерное функционирование (де)кодера видео, определяющего фильтрацию блоков, кодированных на основе палитры, в соответствии со способами этого раскрытия.

[0018] Фиг. 6 является схемой последовательности операций, иллюстрирующей  
45 примерное функционирование видеокодера, определяющего палитру для блока видеоданных, кодированного на основе палитры, в соответствии со способами этого раскрытия.

### **ПОДРОБНОЕ ОПИСАНИЕ**

[0019] Это раскрытие включает в себя способы кодирования и сжатия видео. В частности, это раскрытие описывает способы для поддержки кодирования видеоконтента, особенно экранного контента, с помощью кодирования на основе палитры. Это раскрытие описывает многочисленные технологические аспекты кодирования на основе палитры, включая способы определения размера палитры, записи палитры и фильтрации блоков, кодированных на основе палитры, при кодировании видео.

[0020] При обычном кодировании видео, предполагается, что изображения должны быть непрерывными по тону и пространственно гладкими. На основе этих предположений, были разработаны различные инструменты, такие как блочное преобразование, фильтрация и т.д., и такие инструменты показали хорошую продуктивность для кодирования видео с естественным содержанием. Однако в приложениях, таких как удаленный настольный компьютер, совместная работа и беспроводное отображение, сгенерированный компьютером экранный контент (например, такой как текст или компьютерная графика) может быть доминантным контентом, который должен быть сжат. Этот тип контента имеет тенденцию в дискретному тону, и четким линиям признаков и высококонтрастным границам объекта. Предположения непрерывного тона и гладкости могут более не применяться к экранному контенту, и таким образом обычные способы кодирования видео могут быть неэффективными способами для сжатия видеоданных, включающих в себя экранный контент.

[0021] Это раскрытие описывает кодирование на основе палитры, которое может быть особенно подходящим для кодирования сгенерированного экраном контента. Например, предположим, что конкретная область видеоданных имеет относительно небольшое число цветов, (де)кодер видео (например, видеокодер или декодер) может сформировать так называемую "палитру" для представления видеоданных конкретной области. Палитра может быть выражена как таблица цветов, представляющих видеоданные конкретной области (например, заданный блок). Например, палитра может включать в себя наиболее доминантные цвета (т.е., значения пикселей) в заданном блоке. В некоторых случаях, наиболее доминантные цвета могут включать в себя один или более цветов, которые могут появляться наиболее часто внутри блока. Дополнительно, в некоторых случаях, (де)кодер видео может применить пороговое значение для определения того, должен ли цвет быть включен в блок, как один из наиболее доминантных цветов. Согласно различным аспектам кодирования на основе палитры, (де)кодер видео может кодировать значения индексов, указывающие один или более из пикселей текущего блока, вместо кодирования фактических значений пикселей или их остатков для текущего блока. В контексте кодирования на основе палитры, значения индексов указывают соответствующие записи в палитре, которые используются для представления цветов отдельных пикселей текущего блока.

[0022] Например, видеокодер может кодировать блок видеоданных посредством определения палитры для блока, локализации записей в палитре, которые представляют цвета одного или более из пикселей блока, и кодирования блока со значениями индексов, которые указывают записи в палитре. Для тех пикселей блока с цветовыми значениями, которые увязываются с записями в палитре, видеокодер может кодировать значения индексов записей для соответствующих пикселей. Для тех пикселей блока с цветовыми значениями, которые не увязываются с записями в палитре, видеокодер может кодировать специальный индекс для пикселя и кодировать фактическое значение пикселя или его остаточное значение (или его квантованную версию). Эти пиксели могут быть



названы "беглыми пикселями". В некоторых примерах, видеокодер может сигнализировать палитру и значения индексов и любые беглые пиксели в кодированном битовом потоке. В свою очередь, видеодекодер может получить, из кодированного битового потока, палитру для блока, также как и значения индексов для одного или более из пикселей блока и значения пикселей для любых беглых пикселей блока. Видеодекодер может увязать значения индексов с записями палитры и декодировать беглые пиксели для реконструкции значений пикселей блока.

[0023] Способы кодирования видеоданных на основе палитры могут быть использованы с одним или более другими способами кодирования, такими как способы кодирования с внешним или внутренним предсказанием. Например, как описано более подробно ниже, видеокодер или видеодекодер или объединенный кодер-декодер (кодек), может быть выполнен с возможностью выполнения кодирования с внешним или внутренним предсказанием, также как и кодирования на основе палитры.

[0024] В некоторых примерах, способы кодирования на основе палитры могут быть сконфигурированы для использования с одним или более стандартами кодирования видео. Например, Высокоэффективное кодирование видео (HEVC) является новым стандартом кодирования видео, разрабатываемым посредством Объединенной команды по кодированию видео (JCT-VC) Экспертной группы по кодированию видео (VCEG) ITU-T и Экспертной группы по кинематографии (MPEG) ISO/IEC. Последний черновик текстовой спецификации HEVC описан в Bross et al., "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Last Call)", JCT-VC из ITU-T SG16 WP 3 и ISO/IEC JCT 1/SC 29/WG 11, 12ая встреча: Женева, CH, 14-23 января 2013, JCTVC-L1003\_v34 ("HEVC Draft 10"), доступный по адресу:

[http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v34.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip).

Оформленный документ стандарт HEVC опубликован как "ITU-T H.265, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video - High efficiency video coding", Сектор стандартизации телекоммуникаций Международного телекоммуникационного союза (ITU), апрель 2013.

[0025] В отношении механизма HEVC, в качестве примера, способы кодирования на основе палитры могут быть выполнены с возможностью использования в качестве режима единицы кодирования (CU). В других примерах, способы кодирования на основе палитры могут быть выполнены с возможностью использования в качестве режима PU в механизме HEVC. Соответственно, все нижеследующие раскрытые процессы, описанные в контексте режима CU, могут дополнительно или в качестве альтернативы применяться к PU. Однако, эти основанные на HEVC примеры не должны считаться ограничением текущих способов кодирования на основе палитры, описанных в настоящем документе, как таковые, способы могут быть применены для работы независимо или как часть других существующих или еще разрабатываемых систем/стандартов. В этих случаях, единицей для кодирования с помощью палитры могут быть квадратные блоки, прямоугольные блоки или даже области непрямоугольной формы.

[0026] Основная идея кодирования на основе палитры состоит в том, что для каждой CU получают палитру, которая включает в себя наиболее доминантные цвета (т.е., значения пикселей) в текущей CU. Размер палитры и записи палитры для палитры могут быть переданы из видеокодера видеодекодеру. Размер палитры и записи палитры могут быть кодированы напрямую или кодированы с предсказанием с использованием размера и записей, соответственно, палитр для одной или более соседних CU (например, кодированных CU, находящихся выше и/или слева). Пиксели CU могут затем быть кодированы на основе палитры согласно некоторому порядку сканирования. Для

каждого размещения пикселя в CU, может быть передан флаг для указания, включен ли цвет пикселя в палитру. Для тех пикселей, которые увязываются с записью в палитре, индекс палитры, ассоциированный с этой записью, может быть сигнализирован для заданного размещения пикселя в CU. Для тех пикселей с цветовыми значениями, которые не увязываются с записями в палитре (т.е., беглых пикселей), пикселю может быть присвоен специальный индекс, и фактическое значение пикселя или его остаточное значение (или его квантованная версия) могут быть переданы для заданного размещения пикселя в CU. Беглые пиксели могут быть кодированы с использованием любого существующего способа энтропийного кодирования, такого как кодирование с фиксированной длиной кодового слова, унарное кодирование и т.д.

[0027] В кодировании видео на основе палитры, (де)кодер видео (например, видеокодер или декодер) получает палитру для заданного видеоблока, который должен быть кодирован. Как рассмотрено выше, палитру получают для включения наиболее доминантных цветов или значений пикселей в заданном блоке. В некоторых примерах, для того, чтобы определить наиболее доминантные цвета, может быть использована гистограмма для кластеризации пикселей заданного блока в разные группы. Использование правильного алгоритма для получения палитры может считаться одной из ключевых процедур в кодировании на основе палитры. Способы этого раскрытия направлены на улучшения в получении палитр для кодирования на основе палитры.

[0028] Guo et al., "RCE4: Summary report of HEVC Range Extensions Core Experiments 4 (RCE4) on palette coding for screen content", Объединенная команда по кодированию видео (JCT-VC) из ITU-T SG 16 WP 3 и ISO/IEC JTC 1/SC 29/WG 11, 16ая встреча: Сан-Хосе, США, 9-17 января 2014, JCTVC-P0035, доступный по адресу [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/16\\_San\\_Jose/wg11/JCTVC-P0035-v1.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/16_San_Jose/wg11/JCTVC-P0035-v1.zip) (в дальнейшем, "JCTVC-P0035"), описывает два способа кодирования на основе палитры, сообщенной для достижения значительного уменьшения накладных расходов по оптимизации 'искажение Бьонтегарда – битрейт' (BD-rate), особенно для экранного контента. Ниже вкратце приводятся два способа.

[0029] В первом способе, описанном более подробно в Guo et al., "RCE4: Test 1. Major-color-based screen content coding", JCT-VC из ITU-T SG 16 WP 3 и ISO/IEC JTC 1/SC 29/WG 11, 16ая встреча: Сан-Хосе, США, 9-17 января 2014, JCTVC-P0108, доступном по адресу [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/16\\_San\\_Jose/wg11/JCTVC-P0108-v1.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/16_San_Jose/wg11/JCTVC-P0108-v1.zip) (в дальнейшем, "JCTVC-P0108"), для классификации пикселей CU, которые должны быть кодированы, используется алгоритм на основе гистограммы. В частности, наиболее значительными N пиковыми значениями на гистограмме являются выбранные главные цвета для CU. Значения пикселей, которые близки к одному из главных цветов, могут быть квантованы до главного цвета. Пиксели, которые не принадлежат какому-либо главному цвету, считаются беглыми пикселями, которые также квантуются перед кодированием. Для кодирования без потерь, квантование не используется.

[0030] Согласно первому способу, посредством использования классификации, пиксели CU могут быть преобразованы в индексы цветов. Число и значения главных цветов кодируются, и затем кодируются индексы цветов. Для каждой линии пикселей CU, сигнализируется флаг для указания режима кодирования. Есть три режима: горизонтальный режим, вертикальный режим и обычный режим. Если режимом является горизонтальный режим, вся линия пикселей CU совместно использует одинаковый индекс цвета, и передается индекс цвета для каждой линии пикселей. Если режимом является вертикальный режим, вся линия пикселей является такой же как линия пикселей

выше. В этом случае, ничего не передается (например, в кодированном битовом потоке ничего не передается), так как текущая линия пикселей копирует индексы цветов линии пикселей выше. Если режимом является обычный режим, флаг сигнализируется для каждого положения пикселя, чтобы указать, является ли оно таким же как положение

5 левого пикселя и пикселя выше. Если текущий пиксель отличается от левого пикселя и пикселя выше, передается индекс для текущего пикселя. В дополнение, если текущий пиксель является беглым пикселем, передается значение пикселя.

[0031] Во втором способе, описанном более подробно в Guo, et al., "RCE4: Results of Test 2 on Palette Mode for Screen Content Coding", JCT-VC из ITU-T SG 16 WP 3 и ISO/IEC

10 JTC 1/SC 29/WG 11, 16ая встреча: Сан-Хосе, США, 9-17 января 2014, JCTVC-P0198, доступном по адресу [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/16\\_San\\_Jose/wg11/JCTVC-P0198-v3.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/16_San_Jose/wg11/JCTVC-P0198-v3.zip) (в дальнейшем называемый "JCTVC-P0198"), кодирование видео на основе палитры вводится в качестве режима CU. Процесс кодирования по второму способу сначала включает в себя передачу палитры для текущей CU. Схема

15 поэлементного предсказания используется для кодирования текущей палитры на основе палитры левой CU (т.е., CU, соседней с текущей CU непосредственно слева). После этого, передаются непредсказанные записи палитры.

[0032] Процесс кодирования по второму способу также включает в себя передачу значений пикселей текущей CU. Пиксели в текущей CU кодируются в порядке растрового

20 сканирования с использованием нижеследующих трех режимов. В "режиме ряда", сначала сигнализируется индекс палитры, за которым следует "palette\_run" с длиной M. Нижеследующие M индексов палитры являются такими же как сигнализированный индекс палитры. В "режиме копирования сверху", передается значение "copy\_run" с длиной N для указания, что каждый из нижеследующих N индексов палитры является

25 таким же как соответствующий индекс палитры в линии пикселей выше. В "режиме пикселей", сначала передается флаг предсказания. Флаг предсказания, равный 1, указывает, что остаток предсказания передается для текущего значения пикселя с использованием реконструированного верхнего соседнего пикселя в качестве предиктора. Если флаг предсказания равен 0, текущее значение пикселя передается без

30 предсказания.

[0033] Способы этого раскрытия направлены на улучшения в получении палитр для кодирования на основе палитры. Более конкретно, способы этого раскрытия включают в себя изменение размера палитры и записей палитры для палитры в видеокодере, и определение, отключить ли фильтрацию блоков, кодированных на основе палитры,

35 в видеокодере или декодере.

[0034] В некоторых примерах, после определения размера палитры и записей палитры для текущего блока с использованием способа кластеризации, как описано в JCTVC-P0198, способы данного раскрытия обеспечивают видеокодеру возможность адаптивного определения размера палитры и изменения записей палитры согласно

40 издержкам на оптимизацию по скорости-искажению. Таким образом, способы обеспечивают возможность изменения размера палитры и записей палитры, которые были первоначально изменены для палитры посредством способа кластеризации. Например, способы включают в себя определение, сохранить ли или удалить заданное значение индекса палитры, определение, объединить ли заданное значение индекса с

45 соседним значением индекса, и уточнение вычисленных записей палитры. Хотя это раскрытие прежде всего описывает способы для изменения размера палитры и записей палитры как выполняемые видеокодером, в других примерах способы изменения палитры могут также быть выполнены декодером. В этом случае, например,

видеодекодер может отдельно получить и изменить палитру, или может изменить палитру, которая сигнализирована видеокодером, в соответствии со способами этого раскрытия.

[0035] В дополнение, способы обеспечивают видеокодеру и видеодекодеру возможность отключения фильтрации, такого как фильтрация для удаления блочности или фильтрация с адаптивным смещением выборки (SAO), для блоков, кодированных на основе палитры. Обычно, блоки, кодированные на основе палитры, интерпретировались также как блоки, кодированные посредством внешнего кодирования, и в этой связи фильтрация автоматически применялось к реконструированным блокам перед сохранением в буфере декодированных изображений или выводом для отображения. Согласно раскрытым способам, видеокодер и видеодекодер могут определить, что реконструированный блок является блоком, кодированным на основе палитры, и на основе этого определения отключить фильтрацию перед сохранением или выводом блока. Применение фильтрации к блокам, кодированным на основе палитры, может не привести к улучшенному выводу таким же образом как с блоками, кодированными посредством внешнего кодирования (например, если имеются относительно дискретные тоны, обычно ассоциированные с кодированием с помощью палитры). Соответственно, способы этого раскрытия могут быть использованы для уменьшения сложности кодирования для блоков, кодированных на основе палитры, без воздействия (или с относительно небольшим воздействием) на качество.

[0036] Фиг. 1 является блок-схемой, иллюстрирующей примерную систему 10 кодирования видео, которая может использовать способы этого раскрытия. Как используется в настоящем документе, термин "(де)кодер видео" ссылается в общем как на видеокодеры, так и видеодекодеры. В этом раскрытии, термины "кодирование видео" или "кодирование" могут ссылаться в основном на кодирование видео или декодирование видео. Видеокодер 20 и видеодекодер 30 системы 10 кодирования видео представляет примеры устройств, которые могут быть выполнены с возможностью выполнения способов кодирования видео на основе палитры в соответствии с различными примерами, описанными в этом раскрытии. Например, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью выборочного кодирования различных блоков видеоданных, таких как CU или PU в HEVC-кодировании, с использованием либо кодирования на основе палитры, либо кодирования не на основе палитры. Режимы кодирования не на основе палитры могут относиться к различным режимам временного кодирования с внешним предсказанием или режимам пространственного кодирования с внутренним предсказанием, таким как различные режимы кодирования, точно определенные стандартом HEVC.

[0037] Как показано на фиг. 1, система 10 кодирования видео включает в себя устройство-источник 12 и устройство-адресат 14. Устройство-источник 12 генерирует кодированные видеоданные. Соответственно, устройство-источник 12 может быть названо устройством кодирования видео или аппаратом кодирования видео. Устройство-адресат 14 может декодировать кодированные видеоданные, сгенерированные устройством-источником 12. Соответственно, устройство-адресат 14 может быть названо устройством декодирования видео или аппаратом декодирования видео. Устройство-источник 12 и устройство-адресат 14 могут быть примерами устройств кодирования видео или аппаратов кодирования видео.

[0038] Устройство-источник 12 и устройство-адресат 14 могут содержать широкий диапазон устройств, включающих в себя настольные компьютеры, устройства

беспроводной связи, мобильные вычислительные устройства, компьютеры типа ноутбук (например, переносные компьютеры), планшетные компьютеры, телеприставки, телефонные трубки, такие как так называемые "интеллектуальные" телефоны, телевизоры, камеры, устройства отображения, цифровые медиапроигрыватели, игровые

5 видеоконсоли, автомобильные компьютеры или подобные.

[0039] Устройство-адресат 14 может принимать кодированные видеоданные от устройства-источника 12 по каналу 16. Канал 16 может содержать одну или более сред или устройств, способных перемещать кодированные видеоданные из устройства-источника 12 в устройство-адресат 14. В одном примере, канал 16 может содержать

10 одну или более сред связи, которые обеспечивают устройству-источнику 12 возможность передачи кодированных видеоданных прямо на устройство-адресат 14 в реальном времени. В этом примере, устройство-источник 12 может модулировать кодированные видеоданные согласно стандарту связи, такому как протокол беспроводной связи, и может передавать модулированные видеоданные на устройство-адресат 14. Одна или

15 более сред связи могут включать в себя любые среды беспроводной и/или проводной связи, такие как радиочастотный (РЧ) спектр или одна или более физических линий передачи. Одна или более сред связи могут образовывать часть пакетной сети, такой как локальная сеть, территориальная сеть или глобальная сеть (например, Интернет). Одна или более сред связи могут включать в себя маршрутизаторы, коммутаторы,

20 базовые станции или другое оборудование, которые способствуют связи от устройства-источника 12 к устройству-адресату 14.

[0040] В другом примере, канал 16 может включать в себя носитель информации, который хранит кодированные видеоданные, сгенерированные устройством-источником 12. В этом примере, устройство-адресат 14 может осуществить доступ к носителю

25 информации посредством осуществления доступа к диску или осуществления доступа к карте. Носитель информации может включать в себя разнообразие носителей данных с локальным доступом, таких как диски Blu-ray, DVD, CD-ROM, флэш-память или другие подходящие цифровые носители информации для хранения кодированных видеоданных.

[0041] В дополнительном примере, канал 16 может включать в себя файловый сервер или другое промежуточное устройство хранения, которое хранит кодированные видеоданные, сгенерированные устройством-источником 12. В этом примере, устройство-адресат 14 может осуществлять доступ к кодированным видеоданным, хранящимся на файловом сервере или другом промежуточном хранилище, посредством потоковой

30 передачи или загрузки. Файловым сервером может быть тип сервера, способный хранить кодированные видеоданные и передавать кодированные видеоданные на устройство-адресат 14. Примерные файловые серверы включают в себя веб-серверы (например, для веб-сайта), серверы протокола передачи файлов (FTP), устройство подключаемого к сети хранилища (NAS) и локальные дисковые накопители.

[0042] Устройство-адресат 14 может осуществлять доступ к кодированным видеоданным посредством стандартного информационного соединения, такого как Интернет-соединение. Примерные типы информационных соединений могут включать в себя беспроводные каналы (например, Wi-Fi соединения), проводные соединения (например, DSL, кабельный модем и т.д.), или их комбинации, которые подходят для

40 осуществления доступа к кодированным видеоданным, хранящимся на файловом сервере. Передача кодированных видеоданных из файлового сервера может быть потоковой передачей, передачей на загрузку или их комбинацией.

[0043] Способы этого раскрытия не ограничены применениями или установками беспроводной связи. Способы могут быть применены к кодированию видео при

поддержке разнообразия применений мультимедийных средств, таких как телевизионные трансляции по радиointерфейсу, кабельные телевизионные передачи, спутниковые телевизионные передачи, потоковые передачи видео, например, через Интернет, кодирование видеоданных для хранения на носителе данных, декодирование видеоданных, хранящихся на носителе данных, или других применений. В некоторых примерах, система 10 кодирования видео может быть выполнена с возможностью поддержки односторонней или двухсторонней передачи видео для поддержки применений, таких как потоковая передача видео, проигрывание видео, трансляция видео и/или видеотелефония.

[0044] Фиг. 1 является лишь примером, и способы этого раскрытия могут применяться к настройкам кодирования видео (например, кодирование видео и декодирование видео), которые необязательно включают в себя какой-либо обмен данными между устройствами кодирования и декодирования. В других примерах, данные извлекаются из локальной памяти, передаются потоком по сети или подобным образом. Устройство кодирования видео может кодировать и сохранять данные в память, и/или устройство декодирования видео может извлекать и декодировать данные из памяти. Во многих примерах, кодирование и декодирование выполняется устройствами, которые не осуществляют связь друг с другом, а просто кодируют данные в память и/или извлекают и декодируют данные из памяти.

[0045] В примере по фиг. 1, устройство-источник 12 включает в себя видеоисточник 18, видеокодер 20 и выходной интерфейс 22. В некоторых примерах, выходной интерфейс 22 может включать в себя модулятор/демодулятор (модем) и/или передатчик. Видеоисточник 18 может включать в себя устройство захвата видео, например, видеокамеру, видеоархив, содержащий ранее захваченные видеоданные, интерфейс подачи видео для приема видеоданных от поставщика видеоконтента, и/или компьютерную графическую систему для генерирования видеоданных, или комбинацию таких источников видеоданных.

[0046] Видеокодер 20 может кодировать видеоданные из видеоисточника 18. В некоторых примерах, устройство-источник 12 напрямую передает кодированные видеоданные устройству-адресату 14 через выходной интерфейс 22. Кодированные видеоданные могут также быть сохранены на носителе информации или файловый сервер для дальнейшего доступа устройством-адресатом 14 для декодирования и/или проигрывания.

[0047] В примере по фиг. 1, устройство 14 адресат включает в себя входной интерфейс 28, видеodeкодер 30 и устройство 32 отображения. В некоторых примерах, входной интерфейс 28 может включать в себя приемник и/или модем. Входной интерфейс 28 может принимать кодированные видеоданные по каналу 16. Устройство 32 отображения может быть интегрировано с устройством-адресатом 14 или может быть внешним для него. В общем, устройство 32 отображения отображает декодированные видеоданные. Устройство 32 отображения может содержать разнообразие устройств отображения, таких как жидко-кристаллический дисплей (LCD), плазменный дисплей, дисплей на органических светодиодах (OLED) или другие типы устройств отображения.

[0048] Это раскрытие может в основном относиться к "сигнализации" или "передаче" видеокодером 20 некоторой информации другому устройству, такому как видеodeкодер 30. Термин "сигнализация" или "передача" может в основном относиться к обмену синтаксическими элементами и/или другими данными, используемыми для декодирования сжатых видеоданных. Такой обмен может происходить в реальном времени или почти в реальном времени. В качестве альтернативы, такой обмен может

происходить в некоторый промежуток времени, как например, может происходить при сохранении синтаксических элементов на компьютерно-читаемый носитель информации в кодированном битовом потоке во время кодирования, которые затем могут быть извлечены устройством декодирования в любое время после сохранения на этот носитель. Таким образом, хотя видеodeкодер 30 можно назвать "принимающим" некоторую информацию, прием информации необязательно происходит в реальном или почти в реальном времени и может быть извлечен из носителя спустя некоторое время после сохранения.

[0049] Каждый из видеокодера 20 и видеodeкодера 30 может быть реализован в виде любой из разнообразия подходящих компоновок схем, таких как один или более микропроцессоров, процессоры цифровой обработки сигналов (DSP), специализированные интегральные микросхемы (ASIC), программируемые пользователем вентильные матрицы (FPGA), дискретная логика, аппаратные средства, или любой их комбинации. Если способы частично реализованы в программном обеспечении, устройство может хранить инструкции для программного обеспечения в подходящем долговременном компьютерно-читаемом носителе информации и может исполнять инструкции в аппаратных средствах с использованием одного или более процессоров для выполнения способов этого раскрытия. Любое из вышеприведенного (включая аппаратные средства, программное обеспечение, комбинацию аппаратных средств и программного обеспечения и т.д.) может считаться одним или более процессорами. Каждый из видеокодера 20 и видеodeкодера 30 может быть включен в один или более кодеров или декодеров, любой из которых может быть интегрирован как часть объединенного кодера/декодера (кодека) в соответствующем устройстве, таком как устройство беспроводной связи.

[0050] В некоторых примерах, видеокодер 20 и видеodeкодер 30 функционируют согласно стандарту сжатия видео, такому как стандарт HEVC, упомянутый выше, и описанному в стандарте HEVC. В дополнение к базовому стандарту HEVC, прилагаются постоянные усилия для производства расширений масштабируемого кодирования видео, кодирования нескольких изображений видео и 3D кодирования для HEVC. В дополнение, для расширений стандарта HEVC могут быть предусмотрены режимы кодирования на основе палитры, например, которые описаны в этом раскрытии. В некоторых примерах, способы кодирования на основе палитры, описанные в этом раскрытии, могут быть применены к кодерам и декодерам, выполненным с возможностью функционирования согласно другим способам кодирования видео, таким как стандарт ITU-T-H.188/AVC или будущие стандарты. Соответственно, применение режима кодирования на основе палитры для кодирования единиц кодирования (CU) или единиц предсказания (PU) в HEVC-кодеке описано в целях примера.

[0051] В HEVC и других стандартах кодирования видео, видеопоследовательность обычно включает в себя последовательность изображений. Изображения могут также называться "кадрами". Изображение может включать в себя массивы выборок, обозначенные SL, SCb и SCr. SL является двухмерным массивом (т.е., блоком) выборок яркости. SCb является двухмерным массивом выборок цветовой насыщенности Cb. SCr является двухмерным массивом выборок цветовой насыщенности Cr. Выборки цветовой насыщенности могут также называться в настоящем документе выборками "цветности". В других случаях, изображение может быть монохромным и может включать в себя только массив выборок яркости.

[0052] Чтобы сгенерировать кодированное представление изображения, видеокодер

20 может сгенерировать набор единиц дерева кодирования (СТУ). Каждую из СТУ может составлять блок дерева кодирования выборок яркости, два соответствующие блока дерева кодирования выборок цветности и синтаксические структуры, используемые для кодирования выборок блоков дерева кодирования. Блок дерева кодирования может быть  $N \times N$  блоком выборок. СТУ может также называться "древовидным блоком" или "наибольшей единицей кодирования" (LCU). СТУ HEVC могут быть в широком смысле аналогичными макроблокам других стандартов, таких как H.188/AVC. Однако, СТУ необязательно ограничена конкретным размером и может включать в себя одну или более единиц кодирования (CU). Слайс может включать в себя целое число СТУ, упорядоченных последовательно в порядке растрового сканирования.

[0053] Чтобы сгенерировать кодированную СТУ, видеокодер 20 может рекурсивно выполнить разбиение дерева квадрантов в отношении блоков дерева кодирования СТУ для разделения блоков дерева кодирования на блоки кодирования, отсюда название "единицы дерева кодирования". Блок кодирования является  $N \times N$  блоком выборок. CU может составлять блок кодирования выборок яркости и два соответствующих блока кодирования выборок цветности изображения, которое имеет массив выборок яркости, массив выборок  $C_b$  и массив выборок  $C_r$ , и синтаксические структуры, используемые для кодирования выборок блоков кодирования. Видеокодер 20 может разбить блок кодирования CU на один или более блоков предсказания. Блок предсказания может быть прямоугольным (т.е., квадратным или неквадратным) блоком выборок, в отношении которого применяется такое же предсказание. Единицу предсказания (PU) из CU может составлять блок предсказания выборок яркости, два соответствующих блока предсказания выборок цветности изображения, и синтаксические структуры, используемые для предсказания выборок блока предсказания. Видеокодер 20 может генерировать предсказательные блоки яркости,  $C_b$  и  $C_r$  для блоков предсказания яркости,  $C_b$  и  $C_r$  каждой PU из CU.

[0054] Видеокодер 20 может использовать внутреннее предсказание или внешнее предсказание, чтобы сгенерировать предсказательные блоки для PU. Если видеокодер 20 использует внутреннее предсказание, чтобы сгенерировать предсказательные блоки PU, видеокодер 20 может сгенерировать предсказательные блоки PU на основе декодированных выборок изображения, ассоциированного с PU.

[0055] Если видеокодер 20 использует внешнее предсказание, чтобы сгенерировать предсказательные блоки PU, видеокодер 20 может сгенерировать предсказательные блоки PU на основе декодированных выборок одного или более изображений, отличных от изображения, ассоциированного с данной PU. Видеокодер 20 может использовать однонаправленное предсказание или двунаправленное предсказание чтобы сгенерировать предсказательные блоки PU. Когда видеокодер 20 использует однонаправленное предсказание, чтобы сгенерировать предсказательные блоки для PU, PU может иметь одиночный MV. Когда видеокодер 20 использует двунаправленное предсказание, чтобы сгенерировать предсказательные блоки для PU, PU может иметь два MV.

[0056] После того, как видеокодер 20 сгенерирует предсказательные блоки яркости,  $C_b$  и  $C_r$  для одной или более PU из CU, видеокодер 20 может сгенерировать остаточный блок яркости для CU. Каждая выборка в остаточном блоке яркости CU указывает разницу между выборкой яркости в одном из предсказательных блоков яркости CU и соответствующей выборкой в первоначальном блоке кодирования яркости CU. В дополнение, видеокодер 20 может сгенерировать остаточный блок  $C_b$  для CU. Каждая



выборка в остаточном блоке Сб из СУ может указывать разницу между выборкой Сб в одном из предсказательных блоков Сб из СУ и соответствующей выборкой в первоначальном блоке кодирования Сб из СУ. Видеокодер 20 может также сгенерировать остаточный блок Сг для СУ. Каждая выборка в остаточном блоке Сг из СУ может указывать разницу между выборкой Сг в одном из предсказательных блоков Сг из СУ и соответствующей выборкой в первоначальном блоке кодирования Сг из СУ.

[0057] Кроме того, видеокодер 20 может использовать разбиение дерева квадрантов для разбора остаточных блоков яркости, Сб и Сг из СУ на один или более блоков преобразования яркости, Сб и Сг. Блок преобразования может быть прямоугольным блоком выборок, в отношении которого применяется такое же преобразование. Единицу преобразования (ТУ) из СУ может составлять блок преобразования выборок яркости, два соответствующих блока преобразования выборок цветности и синтаксические структуры, используемые для преобразования выборок блока преобразования. Таким образом, каждая ТУ из СУ может быть ассоциирована с блоком преобразования яркости, блоком преобразования Сб и блоком преобразования Сг. Блок преобразования яркости, ассоциированный с ТУ, может быть подблоком остаточного блока яркости СУ. Блок преобразования Сб может быть подблоком остаточного блока Сб из СУ. Блок преобразования Сг может быть подблоком остаточного блока Сг из СУ.

[0058] Видеокодер 20 может применить одно или более преобразований к блоку преобразования яркости ТУ для генерирования блока коэффициентов яркости для ТУ. Блок коэффициентов может быть двумерным массивом коэффициентов преобразования. Коэффициент преобразования может быть скалярной величиной. Видеокодер 20 может применить одно или более преобразований к блоку преобразования Сб из ТУ для генерирования блока коэффициентов Сб для ТУ. Видеокодер 20 может применить одно или более преобразований к блоку преобразования Сг из ТУ для генерирования блока коэффициентов Сг для ТУ.

[0059] После генерирования блока коэффициентов (например, блока коэффициентов яркости, блока коэффициентов Сб или блока коэффициентов Сг), видеокодер 20 может квантовать блок коэффициентов. Квантование обычно относится к процессу, в котором коэффициенты преобразования квантуются, чтобы по возможности уменьшить величину данных, используемых для представления коэффициентов преобразования, обеспечивая дополнительное сжатие. После того как видеокодер 20 квантует блок коэффициентов, видеокодер 20 может энтропийно кодировать синтаксические элементы, указывающие квантованные коэффициенты преобразования. Например, видеокодер 20 может выполнить контекстно-зависимое адаптивное двоичное арифметическое кодирование (САВАС) в отношении синтаксических элементов, указывающих квантованные коэффициенты преобразования. Видеокодер 20 может вывести энтропийно кодированные синтаксические элементы в битовом потоке.

[0060] Видеокодер 20 может вывести битовый поток, который включает в себя энтропийно кодированные синтаксические элементы. Битовый поток может включать в себя последовательность битов, которая формирует представление кодированных изображений и ассоциированных данных. Битовый поток может содержать последовательность единиц уровня сетевой абстракции (NAL). Каждая из единиц NAL включает в себя заголовок единицы NAL и инкапсулирует полезные данные необработанной байтовой последовательности (RBSP). Заголовок единицы NAL может включать в себя синтаксический элемент, который указывает код типа единицы NAL. Код типа единицы NAL, точно определенный посредством заголовка единицы NAL для единицы NAL, указывает тип единицы NAL. RBSP могут быть синтаксической

структурой, содержащей целое число байтов, которое инкапсулировано внутри единицы NAL. В некоторых случаях, RBSP включает в себя нулевые биты.

[0061] Разные типы единиц NAL могут инкапсулировать разные типы RBSP.

Например, первый тип единицы NAL может инкапсулировать RBSP для набора параметров изображения (PPS), второй тип единицы NAL может инкапсулировать RBSP для кодированного слайса, третий тип единицы NAL может инкапсулировать RBSP для SEI и т.д. Единицы NAL, которые инкапсулируют RBSP для данных кодирования видео (в противоположность RBSP для наборов параметров и сообщений SEI), могут называться единицами NAL уровня кодирования видео (VCL).

[0062] Видеодекодер 30 может принять битовый поток, сгенерированный видеодекодером 20. В дополнение, видеодекодер 30 может осуществлять синтаксический анализ битового потока для декодирования синтаксических элементов из битового потока. Видеодекодер 30 может реконструировать изображения видеоданных по меньшей мере частично на основе синтаксических элементов декодированных из битового потока. Процесс реконструирования видеоданных может быть в общем противоположным процессу, выполняемому видеодекодером 20.

[0063] Например, видеодекодер 30 может использовать MV из PU для определения предсказательных блоков выборок для PU текущей CU. В дополнение, видеодекодер 30 может осуществить обратное квантование блоков коэффициентов преобразования, ассоциированных с TU текущей CU. Видеодекодер 30 может выполнить обратные преобразования в отношении блоков коэффициентов преобразования, чтобы реконструировать блоки преобразования, ассоциированные с TU текущей CU. Видеодекодер 30 может реконструировать блоки кодирования текущей CU посредством добавления выборок предсказательных блоков выборок для PU текущей CU к соответствующим выборкам блоков преобразования из TU текущей CU. Посредством реконструирования блоков кодирования для каждой CU изображения, видеодекодер 30 может реконструировать изображение.

[0064] В некоторых примерах, видеодекодер 20 и видеодекодер 30 может быть выполнен с возможностью выполнения кодирования на основе палитры. Например, в кодировании на основе палитры, вместо выполнения способов кодирования с внутренним предсказанием или внешним предсказанием, описанных выше, видеодекодер 20 и видеодекодер 30 может кодировать так называемую палитру как таблицу цветов для представления видеоданных конкретной области (например, заданного блока). Каждый пиксель может быть ассоциирован с записью палитры, которая представляет цвет пикселя. Например, видеодекодер 20 и видеодекодер 30 может кодировать индекс, который связывает значение пикселя с соответствующим значением в палитре.

[0065] В примере выше, видеодекодер 20 может кодировать блок видеоданных посредством определения палитры для блока, локализации записи в палитре для представления значения каждого пикселя, и кодирования палитры со значениями индексов для пикселей, связывающими значение пикселя с палитрой. Видеодекодер 30 может получить, из кодированного битового потока, палитру для блока, также как и значения индексов для пикселей блока. Видеодекодер 30 может связать значения индексов пикселей с записями палитры для реконструирования значений пикселей блока.

[0066] В некоторых примерах, видеодекодер 20 может кодировать один или более синтаксических элементов, указывающих число последовательных пикселей в заданном порядке сканирования, которые имеют одинаковое значение пикселя. Строка значений пикселей с похожими значениями может быть названа "рядом". В примере в целях

иллюстрации, если два последовательных пикселя в заданном порядке сканирования имеют разные значения, ряд равняется нулю. Если два последовательных пикселя в заданном порядке сканирования имеют одинаковое значение, но третий пиксель в порядке сканирования имеет другое значение, ряд равняется единице. Видеодекодер 30 может получить синтаксические элементы, указывающие ряд из кодированного битового потока, и использовать данные для определения числа размещений последовательных пикселей, которые имеют одинаковое значение индекса.

[0067] В некоторых примерах, видеокодер 20 и видеодекодер 30 могут выполнить копирование линии для одной или более записей из карты значений индексов. Например, видеокодер 20 может указать, что значение пикселя для конкретной записи в карте индексов равно записи в линии над конкретной записью. Видеокодер 20 может также указывать, в качестве ряда, число индексов в порядке сканирования, которые равны записи в линии над конкретной записью. В этом примере, видеокодер 20 и/или видеодекодер 30 могут копировать значения индексов из точно определенной соседней линии и из точно определенного числа записей для линии из кодируемой в текущий момент карты.

[0068] Согласно способам этого раскрытия, видеокодер 20 и видеодекодер 30 могут определить, отключить ли или нет фильтрацию реконструированного блока, на основе того, является ли реконструированный блок блоком, кодированным на основе палитры. В дополнение, видеокодер 20 может получить палитру для блока, кодированного на основе палитры, и определить, изменить ли по меньшей мере одно из размера палитры или одну или более записей палитры для палитры, по меньшей мере частично на основе издержек на оптимизацию по скорости-искажению. В частности, в некоторых примерах, видеокодер 20 и видеодекодер 30 может быть выполнен с возможностью выполнения способов этого раскрытия, которые описаны более подробно ниже относительно Фиг. 4-6.

[0069] Фиг. 2 является блок-схемой, иллюстрирующей примерный видеокодер 20, который может реализовать способы этого раскрытия. Фиг. 2 предоставляется в целях разъяснения и не должна рассматриваться как ограничивающая способы, которые приведены в качестве примера в широком смысле и описаны в этом раскрытии. В целях разъяснения, это раскрытие описывает видеокодер 20 в контексте HEVC-кодирования. Однако, способы этого раскрытия могут быть применимы к другим стандартам или способам кодирования.

[0070] Видеокодер 20 представляет пример устройства, которое может быть выполнено с возможностью выполнения способов кодирования видео на основе палитры в соответствии с различными примерами, описанными в этом раскрытии. Например, видеокодер 20 может быть выполнен с возможностью выборочного кодирования различных блоков видеоданных, таких как CU или PU в HEVC-кодировании, с использованием либо кодирования на основе палитры, либо кодирования не на основе палитры. Режимы кодирования не на основе палитры могут относиться к различным режимам временного кодирования с внешним предсказанием или режимам пространственного кодирования с внутренним предсказанием, таким как различные режимы кодирования, точно определенные стандартом HEVC. Видеокодер 20, в одном примере, может быть выполнен с возможностью генерирования палитры, имеющей записи, указывающие значения пикселей, осуществления выбора значений пикселей в палитре для представления значений пикселей по меньшей мере в некоторых расположениях блока видеоданных, и сигнализирования информации, ассоциирующей по меньшей мере некоторые из расположений блока видеоданных с записями в палитре,

соответствующей, соответственно, выбранным значениям пикселей. Сигнализируемая информация может быть использована видеodeкодером 30 для декодирования видеоданных.

[0071] В примере по Фиг. 2, видеокодер 20 включает в себя память 98 видеоданных, блок 100 обработки предсказания, блок 102 генерирования остатков, блок 104 обработки преобразования, блок 106 квантования, блок 108 обратного предсказания, блок 110 обработки обратного преобразования, блок 112 реконструкции, блок 114 фильтрации, буфер 116 декодированных изображений и блок 118 энтропийного кодирования. Блок 100 обработки предсказания включает в себя блок 120 обработки внешнего предсказания и блок 126 обработки внутреннего предсказания. Блок 120 обработки внешнего предсказания включает в себя блок оценки движения и блок компенсации движения (не показаны). Видеокодер 20 также включает в себя блок 122 кодирования на основе палитры, выполненный с возможностью выполнения различных аспектов способов кодирования на основе палитры, описанных в этом раскрытии. В других примерах, видеокодер 20 может включать в себя больше, меньше или другие функциональные компоненты.

[0072] Память 98 видеоданных может хранить видеоданные, которые должны быть кодированы посредством компонентов видеокодера 20. Видеоданные, хранящиеся в памяти 98 видеоданных, могут быть получены, например, из видеоисточника 18. Буфером 116 декодированных изображений может быть память опорных изображений, которая хранит опорные видеоданные для использования при кодировании видеоданных видеокодером 20, например, в режимах внутреннего или внешнего кодирования. Память 98 видеоданных и буфер 116 декодированных изображений могут быть сформированы посредством любого из разнообразия устройств памяти, таких как динамическая оперативная память (DRAM), включая синхронную DRAM (SDRAM), магниторезистивную RAM (MRAM), резистивную RAM (RRAM), или другие типы устройств памяти. Память 98 видеоданных и буфер 116 декодированных изображений могут быть обеспечены одним и тем же устройством памяти или отдельными устройствами памяти. В различных примерах, память 98 видеоданных может быть на чипе с другими компонентами видеокодера 20, или вне чипа относительно этих компонентов.

[0073] Видеокодер 20 может принимать видеоданные. Видеокодер 20 может кодировать каждую CTU в слайсе изображения видеоданных. Каждая из CTU может быть ассоциирована с блоками дерева кодирования (CTB) яркости равного размера и соответствующими CTB изображения. Как часть кодирования CTU, блок 100 обработки предсказания может выполнить разбиения дерева квадрантов для деления CTB из CTU на прогрессивно небольшие блоки. Меньшим блоком могут быть блоки кодирования CU. Например, блок 100 обработки предсказания может разбить CTB, ассоциированный с CTU, на четыре подблока равного размера, разбить один или более подблоков на четыре подподблока равного размера и т.д.

[0074] Видеокодер 20 может кодировать CU из CTU для генерирования кодированных предсказаний CU (т.е., кодированных CU). В качестве части кодирования CU, блок 100 обработки предсказания может разбить блоки кодирования, ассоциированные с CU среди одной или более PU из CU. Таким образом, каждая PU может быть ассоциирована с блоком предсказания яркости и соответствующими блоками предсказания цветности. Видеокодер 20 и видеodeкодер 30 могут поддерживать PU, имеющие различные размеры. Как указано выше, размер CU может ссылаться на размер блока кодирования яркости CU, и размер PU может ссылаться на размер блока предсказания яркости PU.

Предполагая, что размер конкретной CU составляет  $2N \times 2N$ , видеокодер 20 и видеодекодер 30 могут поддерживать размеры PU  $2N \times 2N$  или  $N \times N$  для внутреннего предсказания, и симметричные размеры PU  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$  или аналогичные для внешнего предсказания. Видеокодер 20 и видеодекодер 30 могут также поддерживать

асимметричное разбиение для размеров PU  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  и  $nR \times 2N$  для внешнего предсказания.

[0075] Блок 120 обработки внешнего предсказания может сгенерировать предсказательные данные для PU посредством выполнения внешнего предсказания в отношении каждой PU из CU. Предсказательные данные для PU могут включать в себя предсказательные блоки выборки PU и информацию движения для PU. Блок 121 внешнего предсказания может выполнять разные операции для PU из CU в зависимости от того, находится ли PU в I-слайсе, P-слайсе или B-слайсе. В I-слайсе все PU предсказаны, посредством внутреннего предсказания. Поэтому, если PU находится в I-слайсе, блок 121 внешнего предсказания не выполняет внешнее предсказание в отношении PU. Таким образом, для блоков, кодированных в I-режиме, предсказанный блок формируется с использованием пространственного предсказания из ранее кодированных соседних блоков внутри одного кадра.

[0076] Если PU находится в P-слайсе, блок оценки движения из блока 120 обработки внешнего предсказания может изучить опорные изображения в списке опорных изображений (например, "RefPicList0") на предмет опорной области для PU. Опорная область для PU может быть областью, внутри опорного изображения, которая содержит блоки выборки, которые наиболее близко соответствуют блокам выборки PU. Блок оценки движения может сгенерировать опорный индекс, который указывает положение в RefPicList0 опорного изображения, содержащего опорную область для PU. В дополнение, блок оценки движения может сгенерировать MV, который указывает пространственное смещение между блоком кодирования PU и опорным размещением, ассоциированным с опорной областью. Например, MV может быть двухмерным вектором, который обеспечивает сдвиг от координат в текущем декодированном изображении к координатам в опорном изображении. Блок оценки движения может выводить опорный индекс и MV в качестве информации движения PU. Блок компенсации движения из блока 120 обработки внешнего предсказания может сгенерировать предсказательные блоки выборки PU на основе фактических или интерполированных выборок в опорном размещении, указанном вектором движения PU.

[0077] Если PU находится в B-слайсе, блок оценки движения может выполнить однонаправленное предсказание или двунаправленное предсказание для PU. Чтобы выполнить однонаправленное предсказание для PU, блок оценки движения может изучить опорные изображения из RefPicList0 или второго списка опорных изображений ("RefPicList1") на предмет опорной области для PU. Блок оценки движения может вывести, в качестве информации движения PU, опорный индекс, который указывает положение опорного изображения, которое содержит опорную область, в RefPicList0 или RefPicList1, MV, который указывает пространственное смещение между блоком выборки PU и опорным размещением, ассоциированным с опорной областью, и один или более указателей направления предсказания, которые указывают, находится ли опорное изображение в RefPicList0 или RefPicList1. Блок компенсации движения из блока 120 обработки внешнего предсказания может сгенерировать предсказательные блоки выборки PU по меньшей мере частично на основе фактических или интерполированных выборок в опорной области, указанной вектором движения PU.

[0078] Для выполнения двунаправленного внешнего предсказания для PU, блок

оценки движения может изучить опорные изображения в RefPicList0 на предмет опорной области для PU и может также изучить опорные изображения в RefPicList1 на предмет другой опорной области для PU. Блок оценки движения может сгенерировать индексы опорных изображений, которые указывают положения опорных изображений, которые содержат опорные области, в RefPicList0 и RefPicList1. В дополнение, блок оценки движения может сгенерировать MV, которые указывают пространственные смещения между опорным размещением, ассоциированным с опорными областями, и блоком выборки PU. Информация движения PU может включать в себя опорные индексы и MV для PU. Блок компенсации движения может сгенерировать предсказательные блоки

10  
15  
20  
25  
30

выборки PU по меньшей мере частично на основе фактических или интерполированных выборок в опорной области, указанной вектором движения PU.

[0079] Блок 126 обработки внутреннего предсказания может сгенерировать предсказательные данные для PU посредством выполнения внутреннего предсказания в отношении PU. Предсказательные данные для PU могут включать в себя предсказательные блоки выборок для PU и различные синтаксические элементы. Блок 126 обработки внутреннего предсказания может выполнять внутреннее предсказание в отношении PU в I-слайсах, P-слайсах и B-слайсах.

[0080] Чтобы выполнить внутреннее предсказание в отношении PU, блок 126 обработки внутреннего предсказания может использовать многочисленные режимы внутреннего предсказания, чтобы сгенерировать многочисленные наборы предсказательных данных для PU. Чтобы использовать режим внутреннего предсказания для генерирования набора предсказательных данных для PU, блок 126 обработки внутреннего предсказания может расширить выборки из блоков выборок соседних PU среди блоков выборок PU в направлении, ассоциированном с режимом внутреннего предсказания. Соседние PU могут быть сверху, сверху и справа, сверху и слева, или

20  
25  
30

слева от PU, предполагая порядок кодирования слева направо, сверху вниз для PU, CU и STU. Блок 126 обработки внутреннего предсказания может использовать различное число режимов внутреннего предсказания, например, 33 режима направленного внутреннего предсказания. В некоторых примерах, число режимов внутреннего предсказания может зависеть от размера области, ассоциированной с PU.

[0081] Блок 100 обработки предсказания может выбрать предсказательные данные для PU из CU из предсказательных данных, сгенерированных блоком 120 обработки внешнего предсказания для PU, или предсказательных данных, сгенерированных блоком 126 обработки внутреннего предсказания для PU. В некоторых примерах, блок 100 обработки предсказания выбирает предсказательные данные для PU из CU на основе показателей зависимости искажений от скорости передачи из наборов предсказательных данных. Предсказательные блоки выборок выбранных предсказательных данных могут быть названы в настоящем документе выбранными предсказательными блоками

35  
40

выборок.

[0082] Блок 102 генерирования остатков может сгенерировать, на основе блока кодирования яркости, Sb и Cr из CU и выбранных предсказательных блоков яркости, Sb и Cr из PU из CU, остаточные блоки яркости, Sb и Cr из CU. Например, блок 102 генерирования остатков может сгенерировать остаточные блоки CU так, чтобы каждая выборка в остаточных блоках имела значение, равное разнице между выборкой в блоке кодирования CU и соответствующей выборкой в соответствующем выбранном предсказательном блоке выборки PU из CU.

45

[0083] Блок 104 обработки преобразования может выполнить разбиение дерева квадрантов для разбиения остаточных блоков, ассоциированных с CU, на блоки

преобразования, ассоциированные с TU из CU. Таким образом, TU может быть ассоциирована с блоком преобразования яркости и двумя блоками преобразования цветности. Размеры и положения блоков преобразования яркости и цветности TU из CU могут или не могут быть на основе размеров и положений блоков предсказания PU из CU. Структура дерева квадрантов, известная как "остаточное дерево квадрантов" (RQT) может включать в себя узлы, ассоциированные с каждой из областей. TU из CU могут соответствовать концевым узлам RQT.

[0084] Блок 104 обработки преобразования может сгенерировать блоки коэффициентов преобразования для каждой TU из CU посредством применения одного или более преобразований к блокам преобразования TU. Блок 104 обработки преобразования может применять различные преобразования к блоку преобразования, ассоциированному с TU. Например, блок 104 обработки преобразования может применить к блоку преобразования дискретное косинусное преобразование (DCT), направленное преобразование или принципиально аналогичное преобразование. В некоторых примерах, блок 104 обработки преобразования не применяет преобразования к блоку преобразования. В таких примерах, блок преобразования может быть интерпретирован как блок коэффициентов преобразования.

[0085] Блок 106 квантования может квантовать коэффициенты преобразования в блоке коэффициентов. Процесс квантования может уменьшить глубину в битах, ассоциированную с некоторыми или всеми коэффициентами преобразования. Например, n-битовый коэффициент преобразования может быть округлен в меньшую сторону до m-битового коэффициента преобразования во время квантования, где n больше, чем m. Блок 106 квантования может квантовать блок коэффициентов, ассоциированный с TU из CU, на основе значения параметра квантования (QP), ассоциированного с CU. Видеокодер 20 может регулировать степень квантования, применяемую к блокам коэффициентов, ассоциированным с CU, посредством регулирования значения QP, ассоциированного с CU. Квантование может вносить потери информации, таким образом квантованные коэффициенты преобразования могут иметь меньшую точность, чем первоначальные коэффициенты преобразования.

[0086] Блок 108 обратного преобразования и блок 110 обработки обратного преобразования могут применить обратное квантование и обратные преобразования к блоку коэффициентов, соответственно, чтобы реконструировать остаточный блок из блока коэффициентов. Блок 112 реконструкции может добавить реконструированный остаточный блок к соответствующим выборкам из одного или более предсказательных блоков выборок, сгенерированных блоком 100 обработки предсказания, чтобы произвести реконструированный блок преобразования, ассоциированный с TU. Посредством реконструирования блоков преобразования для каждой TU из CU таким образом, видеокодер 20 может реконструировать блоки кодирования CU.

[0087] Блок 114 фильтрации может выполнить одну или более операций удаления блочности для уменьшения артефактов блочности в блоках кодирования, ассоциированных с CU. Буфер 116 декодированных изображений может сохранить реконструированные блоки кодирования после того, как блок 114 фильтрации выполнит одну или более операций удаления блочности в отношении реконструированных блоков кодирования. Блок 120 внешнего предсказания может использовать опорное изображение, которое содержит реконструированные блоки кодирования для выполнения внешнего предсказания в отношении PU других изображений. В дополнение, блок 126 обработки внутреннего предсказания может использовать реконструированные блоки кодирования в буфере 116 декодированных

изображений для выполнения внутреннего предсказания в отношении других PU в том же изображении как и CU.

[0088] Блок 118 энтропийного кодирования может принимать данные от других функциональных компонентов видеокодера 20. Например, блок 118 энтропийного кодирования может принимать блоки коэффициентов из блока 106 квантования и может принимать синтаксические элементы из блока 100 обработки предсказания. Блок 118 энтропийного кодирования может выполнить одну или более операций энтропийного кодирования в отношении данных, чтобы сгенерировать энтропийно кодированные данные. Например, блок 118 энтропийного кодирования может выполнить операцию контекстно-адаптированного кодирования с переменной длиной кодового слова (CAVLC), операцию CABAC, операцию перекодирования с переменной (V2V) длиной кодового слова, операцию зависимого от синтаксиса контекстно-адаптивного двоичного арифметического кодирования (SBAC), операцию кодирования с энтропией разбиения по вероятностному интервалу (PIPE), операцию кодирования методом экспоненциального кода Голомба, или любую другую операцию энтропийного кодирования в отношении данных. Видеокодер 20 может вывести битовый поток, который включает в себя энтропийно кодированные данные, сгенерированные блоком 118 энтропийного кодирования. Например, битовый поток может включать в себя данные, которые представляют собой RQT для CU.

[0089] В соответствии с различными примерами этого раскрытия, видеокодер 20 может быть выполнен с возможностью выполнения кодирования на основе палитры. В отношении механизма HEVC, в качестве примера, способы кодирования на основе палитры могут быть выполнены с возможностью использования в качестве режима единицы кодирования (CU). В других примерах, способы кодирования на основе палитры могут быть выполнены с возможностью использования в качестве режима PU в механизме HEVC. Соответственно, все из раскрытых процессов, описанных в настоящем документе (по всему этому раскрытию), в контексте режима CU могут, дополнительно или в качестве альтернативы, применяться к PU. Однако, эти основанные на HEVC примеры не должны считаться ограничением текущих способов кодирования на основе палитры, описанных в настоящем документе, как таковые, способы могут быть применены для работы независимо или как часть других существующих или еще разрабатываемых систем/стандартов. В этих случаях, единицей для кодирования с помощью палитры могут быть квадратные блоки, прямоугольные блоки или даже области непрямоугольной формы.

[0090] Блок 122 кодирования на основе палитры, например, может выполнить кодирование на основе палитры, когда выбран режим кодирования на основе палитры, например, для CU или PU. Например, блок 122 кодирования на основе палитры может быть выполнен с возможностью генерирования палитры, имеющей записи, указывающее значения пикселей, осуществления выбора значений пикселей в палитре для представления значений пикселей по меньшей мере в некоторых положениях блока видеоданных, и сигнализирования информации, ассоциирующей по меньшей мере некоторые из положений блока видеоданных с записями в палитре, соответствующими, соответственно, выбранным значениям пикселей. Хотя различные функции описаны как выполняемые блоком 122 кодирования на основе палитры, некоторые или все такие функции могут быть выполнены другими блоками обработки или комбинацией разных блоков обработки.

[0091] Согласно некоторым аспектам этого раскрытия, блок 122 кодирования на основе палитры может быть выполнен с возможностью получения палитры для блока,



кодированного на основе палитры, и определения, изменить ли по меньшей мере одно из размера палитры или одной или более записей палитры для палитры, по меньшей мере частично на основе издержек на оптимизацию по скорости-искажению. Например, после того, как записи палитры и размер палитры определены для текущего блока с использованием способа кластеризации значений пикселей, описанного более подробно ниже относительно Фиг. 4, блок 122 кодирования на основе палитры определяет, уменьшить ли размер палитры для палитры, первоначально определенный способом кластеризации значений пикселей. В одном примере, блок 122 кодирования на основе палитры определяет, сохранить ли или удалить заданное значение индекса, на основе сравнения издержек на оптимизацию по скорости-искажению для кодирования пикселя с использованием заданного значения индекса или кодирования пикселя как беглого пикселя, включая кодирование квантованного значения пикселя для пикселя. В другом примере, блок 122 кодирования на основе палитры определяет, объединить ли заданное значение индекса с соседним значением индекса (например, объединить значения индексов для формирования единого значения индекса), на основе сравнения издержек на оптимизацию по скорости-искажению для кодирования пикселя с использованием заданного значения индекса или кодирования пикселя с использованием соседнего значения индекса.

[0092] В качестве другого примера, блок 122 кодирования на основе палитры определяет, изменить ли одну или более из записей палитры для палитры, первоначально определенные способом кластеризации значений пикселей. В этом примере, блок 122 кодирования на основе палитры определяет репрезентативную палитру для блока, кодированного на основе палитры, с использованием способа кластеризации k-средних, который описан более подробно ниже относительно Фиг. 4. Репрезентативная палитра включает в себя репрезентативные записи палитры, указывающие средние значения пикселей для пикселей блока. Блок 122 кодирования на основе палитры определяет, изменить ли заданную запись палитры из первоначальной палитры, на основе среднего значения пикселя, указанного посредством соответствующей репрезентативной записи палитры из репрезентативной палитры.

[0093] Способы определения размера палитры и записей палитры для палитры для блоков, кодированных на основе палитры, описаны более подробно ниже относительно Фиг. 4 и 6. Хотя это раскрытие прежде всего описывает способы для изменения размера палитры и записей палитры как выполняемые видеокодером, в других примерах способы изменения палитры могут также быть выполнены видеодекодером, таким как видеодекодер 30. В этом случае, например, видеодекодер 30 может отдельно получить и изменить палитру, или может изменить палитру, которая сигнализирована видеокодером 20, в соответствии со способами этого раскрытия.

[0094] Согласно другим аспектам этого раскрытия, видеокодер 20 может быть выполнен с возможностью определения, отключить ли фильтрацию реконструированных блоков, кодированных на основе палитры, блоком 114 фильтрации перед сохранением реконструированных блоков в буфере 116 декодированных изображений. Например, видеокодер 20 может отключить фильтрацию для удаления блочности и/или SAO-фильтрацию блоков, кодированных на основе палитры, блоком 114 фильтрации. Обычно, блоки, кодированные на основе палитры, интерпретировались также как блоки, кодированные посредством внешнего кодирования, и в этой связи фильтрация автоматически применялось к реконструированным блокам перед сохранением в буфере декодированных изображений или выводом для отображения. Согласно раскрытым способам, видеокодер 20 может

определить, что реконструированный блок является блоком, кодированным на основе палитры, и на основе этого определения отключить фильтрацию блоком 114 фильтрации перед сохранением блока в буфере 116 декодированных изображений. Способы определения фильтрации блоков, кодированных на основе палитры, описаны

5 более подробно ниже относительно Фиг. 4 и 5.

[0095] Фиг. 3 является блок-схемой, иллюстрирующей примерный видеodeкодер 30, который сконфигурирован для реализации способов этого раскрытия. Фиг. 3 предоставляется в целях разъяснения и не является ограничивающей в отношении способов, которые приведены в качестве примера в широком смысле и описаны в этом

10 раскрытии. В целях разъяснения, это раскрытие описывает видеodeкодер 30 в контексте HEVC-кодирования. Однако, способы этого раскрытия могут быть применимы к другим стандартам или способам кодирования.

[0096] Видеodeкодер 30 представляет пример устройства, которое может быть выполнено с возможностью выполнения способов кодирования видео на основе палитры

15 в соответствии с различными примерами, описанными в этом раскрытии. Например, видеodeкодер 30 может быть выполнен с возможностью выборочного декодирования различных блоков видеоданных, таких как CU или PU в HEVC-кодировании, с использованием либо кодирования на основе палитры, либо кодирования не на основе палитры. Режимы кодирования не на основе палитры могут относиться к различным

20 режимам временного кодирования с внешним предсказанием или режимам пространственного кодирования с внутренним предсказанием, таким как различные режимы кодирования, точно определенные стандартом HEVC. Видеodeкодер 30, в одном примере, может быть выполнен с возможностью генерирования палитры, имеющей записи, указывающие значения пикселей, приема информации, ассоциирующей

25 по меньшей мере некоторые положения блока видеоданных с записями в палитре, осуществления выбора значений пикселей в палитре на основе информации, и реконструирования значений пикселей блока на основе выбранных значений пикселей.

[0097] В примере по Фиг. 3, видеodeкодер 30 включает в себя память 148 видеоданных, блок 150 энтропийного декодирования, блок 152 обработки предсказания, блок 154

30 обратного квантования, блок 156 обработки обратного преобразования, блок 158 реконструкции, блок 160 фильтрации и буфер 162 декодированных изображений. Блок 152 обработки предсказания включает в себя блок 164 компенсации движения и блок 166 обработки внутреннего предсказания. Видеodeкодер 30 также включает в себя блок 165 декодирования на основе палитры, выполненный с возможностью

35 выполнения различных аспектов способов кодирования на основе палитры, описанных в этом раскрытии. В других примерах, видеodeкодер 30 может включать в себя больше, меньше или другие функциональные компоненты.

[0098] Память 148 видеоданных может хранить видеоданные, такие как битовый поток кодированного видео, которые должны быть декодированы компонентами

40 видеodeкодера 30. Видеоданные, хранящиеся в памяти 148 видеоданных, могут быть получены, например, из компьютерно-читаемого носителя 16, например, из локального видеoисточника, такого как камера, посредством обмена видеоданными по проводной или беспроводной сети, или посредством осуществления доступа к физическим носителям данных. Память 148 видеоданных может сформировать буфер кодированных

45 изображений (CPB), который хранит кодированные видеоданные из битового потока кодированного видео. Буфером 162 декодированных изображений может быть память опорных изображений, которая хранит опорные видеоданные для использования при декодировании видеоданных видеodeкодером 30, например, в режимах внутреннего

или внешнего кодирования. Память 148 видеоданных и буфер 162 декодированных изображений могут быть сформированы посредством любого из разнообразия устройств памяти, таких как динамическая оперативная память (DRAM), включая синхронную DRAM (SDRAM), магниторезистивную RAM (MRAM), резистивную RAM (RRAM), или другие типы устройств памяти. Память 148 видеоданных и буфер 162 декодированных изображений могут быть обеспечены одним и тем же устройством памяти или отдельными устройствами памяти. В различных примерах, память 148 видеоданных может быть на чипе с другими компонентами видеodeкодера 30, или вне чипа относительно этих компонентов.

[0099] Память 148 видеоданных, т.е., СРВ, может принимать и сохранять кодированные видеоданные (например, единицы NAL) битового потока. Блок 150 энтропийного декодирования может принять кодированные видеоданные (например, единицы NAL) из памяти 148 видеоданных и может осуществить синтаксический анализ единиц NAL для декодирования синтаксических элементов. Блок 150 энтропийного декодирования может энтропийно декодировать энтропийно кодированные синтаксические элементы в единицах NAL. Блок 152 обработки предсказания, блок 154 обратного квантования, блок 156 обработки обратного преобразования, блок 158 реконструирования и блок 160 фильтрования могут сгенерировать декодированные видеоданные на основе синтаксических элементов, полученных (например, извлеченных) из битового потока.

[0100] Единицы NAL битового потока могут включать в себя единицы NAL кодированного слайса. Как часть декодирования битового потока, блок 150 энтропийного декодирования может извлечь и энтропийно декодировать синтаксические элементы из единиц NAL кодированного слайса. Каждый из кодированных слайсов может включать в себя заголовок слайса и данные слайса. Заголовок слайса может содержать синтаксические элементы, относящиеся к слайсу. Синтаксические элементы в заголовке слайса могут включать в себя синтаксический элемент, который идентифицирует PPS, ассоциированный с изображением, который содержит слайс.

[0101] В дополнение к декодированию синтаксических элементов из битового потока, видеodeкодер 30 может выполнить операцию реконструкции в отношении неразбитой CU. Чтобы выполнить операцию реконструкции в отношении неразбитой CU, видеodeкодер 30 может выполнить операцию реконструкции в отношении каждой TU из CU. Посредством выполнения операции реконструкции для каждой TU из CU, видеodeкодер 30 может реконструировать остаточные блоки CU.

[0102] Как часть выполнения операции реконструкции в отношении TU из CU, блок 154 обратного квантования может осуществить обратное квантование, т.е. деквантование, блоков коэффициентов, ассоциированных с TU. Блок 154 обратного квантования может использовать значение QR, ассоциированное с CU из TU, для определения степени квантования и, аналогично, степени обратного квантования для применения блоком 154 обратного квантования. То есть, степенью сжатия, т.е., соотношением числа битов, используемых для представления первоначальной последовательности, и сжатых битов, можно управлять посредством регулирования значения QR, используемого при квантовании коэффициентов преобразования. Степень сжатия может также зависеть от используемого способа энтропийного кодирования.

[0103] После того, как блок 154 обратного квантования осуществляет обратное квантование блока коэффициентов, блок 156 обработки обратного преобразования может применить одно или более обратных преобразований к блоку коэффициентов для того, чтобы сгенерировать остаточный блок, ассоциированный с TU. Например,

блок 156 обработки обратного преобразования может применить обратное DCT, обратное целочисленное преобразование, обратное преобразование Карунена-Лоэва (KLT), обратное вращательное преобразование, обратное направленное преобразование или другое обратное преобразование к блоку коэффициентов.

5 [0104] Если PU кодирована с использованием внутреннего предсказания, блок 166 обработки внутреннего предсказания может выполнить внутреннее предсказание, чтобы сгенерировать предсказательные блоки для PU. Блок 166 обработки внутреннего предсказания может использовать режим внутреннего предсказания, чтобы сгенерировать предсказательные блоки яркости, Cb и Cr для PU на основе блоков предсказания пространственно соседних PU. Блок 166 обработки внутреннего предсказания может определить режим внутреннего предсказания для PU на основе одного или более синтаксических элементов, декодированных из битового потока.

10 [0105] Блок 152 обработки предсказания может сконструировать первый список опорных изображений (RefPicList0) и второй список опорных изображений (RefPicList1) на основе синтаксических элементов, извлеченных из битового потока. Кроме того, если PU кодирована с использованием внешнего предсказания, блок 150 энтропийного декодирования может извлечь информацию движения для PU. Блок 164 компенсации движения может определить, на основе информации движения PU, одну или более опорных областей для PU. Блок 164 компенсации движения может сгенерировать предсказательные блоки яркости, Cb и Cr для PU на основе блоков выборок в одном или более опорных блоках для PU.

15 [0106] Блок 158 реконструирования может использовать блоки преобразования яркости, Cb и Cr, ассоциированные с TU из CU, и предсказательные блоки яркости, Cb и Cr для PU из CU, т.е., либо данные с внутренним предсказанием, либо данные с внешним предсказанием, в зависимости от случая, чтобы реконструировать блоки кодирования яркости, Cb и Cr из CU. Например, блок 158 реконструирования может добавить выборки блоков преобразования яркости, Cb и Cr к соответствующим выборкам предсказательных блоков яркости, Cb и Cr, чтобы реконструировать блоки кодирования яркости, Cb и Cr из CU.

20 [0107] Блок 160 фильтрации может выполнить операцию удаления блочности для уменьшения артефактов блочности, ассоциированных с блоками кодирования яркости, Cb и Cr из CU. Витербокодер 30 может сохранить блоки кодирования яркости, Cb и Cr из CU в буфере 162 декодированных изображений. Буфер 162 декодированных изображений может предоставить опорные изображения для последующей компенсации движения, внутреннего предсказания и представления на устройстве отображения, таком как устройство 32 отображения по Фиг. 1. Например, витербокодер 30 может выполнить, на основе блоков яркости, Cb и Cr в буфере 162 декодированных изображений, операции внутреннего предсказания или внешнего предсказания в отношении PU из других CU. Таким образом, витербокодер 30 может извлечь, из битового потока, уровни коэффициентов преобразования значимого блока коэффициентов яркости, осуществить обратное квантование уровней коэффициентов преобразования, применить преобразование к уровням коэффициентов преобразования, чтобы сгенерировать блок преобразования, сгенерировать блок кодирования по меньшей мере частично на основе блока преобразования, и вывести блок кодирования для отображения.

45 [0108] В соответствии с различными примерами этого раскрытия, витербокодер 30 может быть выполнен с возможностью выполнения кодирования на основе палитры. Блок 165 декодирования на основе палитры, например, может выполнить декодирование

на основе палитры, когда выбран режим декодирования на основе палитры, например, для CU или PU. Например, блок 165 декодирования на основе палитры может быть выполнен с возможностью генерирования палитры, имеющей записи, указывающие значения пикселей, приема информации, ассоциирующей по меньшей мере некоторые положения блока видеоданных с записями в палитре, осуществления выбора значений пикселей в палитре на основе информации, и реконструирования значений пикселей блока на основе выбранных значений пикселей. Хотя различные функции описаны как выполняемые блоком 165 декодирования на основе палитры, некоторые или все такие функции могут быть выполнены другими блоками обработки или комбинацией разных блоков обработки.

[0109] Согласно аспектам этого раскрытия, видеodeкодер 30 может быть выполнен с возможностью определения, отключить ли фильтрацию реконструированных блоков, кодированных на основе палитры, блоком 160 фильтрации перед сохранением реконструированных блоков в буфере 162 декодированных изображений. Например, видеodeкодер 30 может отключить фильтрацию для удаления блочности и/или SAO-фильтрацию блоков, кодированных на основе палитры, блоком 160 фильтрации. Обычно, блоки, кодированные на основе палитры, интерпретировались также как блоки, кодированные посредством внешнего кодирования, и в этой связи фильтрация автоматически применялась к реконструированным блокам перед сохранением в буфере декодированных изображений или выводом для отображения. Согласно раскрытым способам, видеodeкодер 30 может определить, что реконструированный блок является блоком, кодированным на основе палитры, и на основе этого определения отключить фильтрацию блоком 160 фильтрации перед сохранением блока в буфере 162 декодированных изображений. Способы определения фильтрации блоков, кодированных на основе палитры, описаны более подробно ниже относительно Фиг. 4 и 5.

[0110] Фиг. 4 является принципиальной схемой, иллюстрирующей пример определения палитры для кодирования видеоданных, соответствующей способам этого раскрытия. Пример по Фиг. 4 включает в себя изображение 178, имеющее первую единицу 180 кодирования (CU), кодированную как блок, кодированный на основе палитры (PAL), ассоциированный с первыми палитрами 184, и вторую CU 188, кодированную как блок, кодированный на основе палитры (PAL), ассоциированный со вторыми палитрами 192. Изображение 178 также включает в себя внутренний блок 196, кодированный с помощью режима кодирования с внутренним преобразованием, и внешний блок 200, который кодирован с помощью режима кодирования с внешним предсказанием. В целях разъяснения способы по Фиг. 4 описаны в контексте видеodeкодера 20 (Фиг. 1 и Фиг. 2) и видеodeкодера 30 (Фиг. 1 и Фиг. 3) и относительно стандарта HEVC. Однако, следует понимать, что способы этого раскрытия не ограничены таким образом, и могут быть применены другими процессорами и/или устройствами кодирования видео в других процессах и/или стандартах кодирования видео.

[0111] В общем, палитра ссылается на число значений пикселей, которые являются доминантными и/или репрезентативными для кодируемой в текущий момент CU, CU 188 в примере по Фиг. 4. Первые палитры 184 и вторые палитры 192 показаны как включающие в себя многочисленные палитры. В некоторых примерах, согласно аспектам этого раскрытия, (де)кодер видео (такой как видеodeкодер 20 или видеodeкодер 30) может кодировать палитры отдельно для каждого цветового компонента CU. Например, видеodeкодер 20 может кодировать палитру для компонента яркости (Y) из CU, другую палитру для компонента цветности (U) из CU, и еще одну палитру для

компонента цветности (V) из CU. В этом примере, записи Y-палитры могут представлять Y-значения пикселей CU, записи U-палитры могут представлять U-значения пикселей CU, и записи V-палитры могут представлять V-значения пикселей CU.

5 [0112] В других примерах, видеокодер 20 может кодировать единую палитру для всех цветовых компонентов CU. В этом примере, видеокодер 20 может кодировать палитру, имеющую i-ю запись, которая является тройным значением, включающим в себя  $Y_i$ ,  $U_i$  и  $V_i$ . В этом случае, палитра включает в себя значения для каждого из компонентов пикселей. Соответственно, представление палитр 184 и 192 как набора палитр, имеющего многочисленные отдельные палитры, является лишь одним примером  
10 и не предназначено для ограничения.

[0113] В примере по Фиг. 4, первые палитры 184 включают в себя три записи 202, 204, 206 палитры, имеющие значение индекса 1, значение индекса 2 и значение индекса 3, соответственно. Записи 202, 204, 206 палитры связывают значения индексов со значениями пикселей (т.е., цветовыми значениями), включающими в себя значение A пикселя, значение B пикселя и значение C пикселя, соответственно. Как описано в  
15 настоящем документе, вместо кодирования фактических значений пикселей первой CU 180, (де)кодер видео (такой как видеокодер 20 или видеodeкодер 30) может использовать кодирование на основе палитры, чтобы кодировать пиксели блока с использованием значений индексов 1, 2, 3. То есть, для каждого положения пикселя первой CU 180,  
20 видеокодер 20 может кодировать значение индекса для пикселя, где значение индекса соответствует записи палитры, которая указывает значение пикселя для пикселя в одной или более из первых палитр 184. Видеodeкодер 30 может получить значения индексов из битового потока и реконструировать значения пикселей с использованием значений индексов и одной или более из первых палитр 184. Таким образом, первые палитры  
25 184 передаются видеокодером 20 в битовом потоке кодированных видеоданных для использования видеodeкодером 30 при декодировании на основе палитры.

[0114] В некоторых примерах, видеокодер 20 и видеodeкодер 30 могут определить вторые палитры 192 на основе первых палитр 184. Например, видеокодер 20 и/или видеodeкодер 30 могут локализовать один или более блоков, исходя из которых  
30 определяются предсказательные палитры, в этом примере первые палитры 184. В некоторых примерах, таких как пример, проиллюстрированный на Фиг. 4, видеокодер 20 и/или видеodeкодер 30 могут локализовать ранее кодированную CU, такую как левая соседняя CU (первая CU 180), при определении предсказательной палитры для второго CU 188.

35 [0115] В примере по Фиг. 4, вторые палитры 192 включают в себя три записи палитры 208, 210, 212, имеющие значение индекса 1, значение индекса 2 и значение индекса 3, соответственно. Записи 208, 210, 212 палитры связывают значения индексов со значениями пикселей, включающими в себя значение A пикселя, значение B пикселя и значение D пикселя, соответственно. В этом примере, видеокодер 20 может кодировать  
40 один или более синтаксических элементов, указывающих, какие записи первых палитр 184 включены во вторые палитры 192. В примере по Фиг. 4, один или более синтаксических элементов проиллюстрированы как вектор 216. Вектор 216 имеет число ассоциированных двоичных элементов (или двоичных битов), причем каждый бит указывает, используется ли предиктор палитры, ассоциированный с этим двоичным  
45 элементом для предсказания записи текущей палитры. Например, вектор 216 указывает, что первые две записи первых палитр 184 (202 и 204) включены во вторые палитры 192 (значение "1" в первых двух двоичных элементах вектора 216), тогда как третья запись первых палитр 184 не включена во вторые палитры 192 (значение "0" в третьем двоичном

элементе вектора 216). В примере по Фиг. 4, вектор является булевым вектором.

[0116] Способы этого раскрытия направлены на улучшения в получении палитр для кодирования на основе палитры. Более конкретно, способы этого раскрытия включают в себя изменение размера палитры и записей палитры для палитры в видеокодере 20, и определение, отключить ли фильтрование блоков, кодированных на основе палитры, в видеокодере 20 и/или видеокодере 30. В примере по Фиг. 4, видеокодер 20 может выполнить способы изменения палитры этого раскрытия для получения первых палитр 184. Первые палитры 184 затем передаются видеокодером 20 в битовом потоке кодированных видеоданных для использования видеокодером 30 при декодировании на основе палитры. Хотя изначально описаны в этом раскрытии относительно видеокодера 20, в других примерах способы изменения палитры этого раскрытия могут быть выполнены и видеокодером 20, и видеокодером 30. В этом случае, например, видеокодер 30 может отдельно получить первые палитры 184 или может изменить первые палитры 184, которые переданы видеокодером 20, в соответствии со способами этого раскрытия. Хотя описаны относительно первых палитр 184 в целях разъяснения, способы могут быть выполнены с использованием разнообразия других палитр с другими конфигурациями и/или размерами.

[0117] В режиме кодирования на основе палитры, как описано в JCTVC-P0035, JCTVC-P0108 и JCTVC-P0198 (приведено выше), видеокодер 20 может получить палитру для блока, кодированного на основе палитры, например, CU 180, с использованием способа кластеризации значений пикселей. Использование правильного алгоритма для получения палитры, например, первых палитр 184, является одной из ключевых процедур при кодировании в режиме палитры. Например, видеокодер 20 может получить гистограмму, включающую в себя множество записей, причем каждая запись включает в себя соответствующее цветовое значение и соответствующее количество пикселей блока, которые увязываются с цветовым значением. В качестве одного примера, видеокодер 20 может определить гистограмму  $H$  блока видеоданных. Гистограмма  $H$  может быть представлена посредством  $H = \{(v_i, f_i), i = \{0, 1, 2, \dots, M\}\}$ , где  $M+1$  является числом разных значений пикселей в текущем блоке,  $v_i$  является значением пикселя  $i$ -ого пикселя, и  $f_i$  является числом появлений  $v_i$  (т.е., сколько пикселей в текущем блоке имеют значение  $v_i$  пикселя). В некоторых примерах, каждая из множества записей гистограммы представляет количество пикселей, которые имеют одинаковое значение пикселя. В некоторых примерах, каждая из множества записей гистограммы представляет количество пикселей, при квантовании, которые имеют одинаковое значение пикселя.

[0118] После получения гистограммы, видеокодер 20 может инициализировать палитру  $P$ , чтобы была пустой, т.е.,  $P = \emptyset$ , инициализировать значение индекса палитры в значение ноль, т.е.  $idx = 0$ , и/или инициализировать значение индекса гистограммы в значение ноль, т.е., задать  $j = 0$ . Видеокодер 20 затем сортирует записи гистограммы в убывающем порядке на основе соответствующего количества пикселей для каждой из записей, так чтобы первая запись гистограммы включала в себя цветовое значение, имеющее наибольшее количество пикселей, относительно других цветовых значений. Например, видеокодер 20 может сортировать гистограмму  $H$  согласно убывающему порядку  $f_i$ , так чтобы значения пикселей, имеющие больше появлений в текущем блоке, помещались вблизи передней части гистограммы. Отсортированная гистограмма может быть представлена посредством  $H_o = \{(u_i, f_i), i = \{0, 1, 2, \dots, M\}, f_i \geq f_{i+1}\}$ .

[0119] Видеокодер 20 может затем определить размер палитры и записи палитры для палитры для текущего блока на основе записей гистограммы. Например, видеокодер

20 может вставить первую запись  $(j, u_j)$  отсортированной гистограммы в палитру, т.е.  $P = P \cup \{(idx, u_j)\}$ . Видеокодер 20 может затем увеличить индекс палитры, т.е.,  $idx = idx + 1$ , и увеличить индекс гистограммы, т.е.,  $j = j + 1$ . Видеокодер 20 может определить, находится ли значение пикселя, соответствующее следующей записи в отсортированной гистограмме, по соседству любого из значений пикселей, уже включенных в палитру, или нет. Например, видеокодер 20 может определить значение расстояния, которое соответствует разности между значением пикселя, соответствующим следующей записи в отсортированной гистограмме, и одним или более из значений пикселей, уже включенных в палитру. Если определенное разностное значение больше, чем порог, или равно ему, т.е.,  $Distance(u_j, x) \geq Thresh$ , видеокодер 20 может вставить новую запись в палитру, которая указывает значение пикселя, соответствующее следующей записи в отсортированной гистограмме. Если определенное разностное значение не удовлетворяет порогу, т.е.,  $Distance(u_j, x) < Thresh$ , видеокодер 20 определяет, что значение пикселя, соответствующее следующей записи в отсортированной гистограмме, находится по соседству со значением пикселя, уже включенным в палитру, и не будет добавлять новую запись в палитру. Видеокодер 20 может затем увеличить индекс гистограммы, т.е.,  $j = j + 1$ .

[0120] В некоторых примерах, видеокодер 20 может остановить определение палитры (например, остановить добавление записей в палитру), если одно или более условий удовлетворяют одному или более соответствующим порогам. В качестве одного примера, видеокодер 20 может остановить определение палитры, когда размер палитры больше, чем предварительно заданное значение. В качестве другого примера, видеокодер 20 может остановить определение палитры, когда значение индекса палитры равняется числу разных значений пикселей в текущем блоке, т.е.,  $idx = M$ . В качестве другого примера, видеокодер 20 может остановить определение палитры, когда значение индекса гистограммы равняется числу разных значений пикселей в текущем блоке, т.е.,  $j = M$ .

[0121] В соответствии со способами этого раскрытия, после определения размера палитры и записей палитры для палитры, видеокодер 20 может определить, изменить ли по меньшей мере одно из размера палитры или одной или более из записей палитры для палитры по меньшей мере частично на основе издержек на оптимизацию по скорости-искажению. Таким образом, способы обеспечивают видеокодеру 20 возможность определения, изменить ли размер палитры и записи палитры, первоначально определенные для палитры.

[0122] Например, видеокодер 20 может определить, сохранить ли или удалить заданное значение индекса, которое соответствует заданной одной из записей палитры, например, значение индекса 1 записи 202 в палитрах 184, на основе издержек на оптимизацию по скорости-искажению для кодирования пикселя CU 180 с использованием значения индекса 1 или кодирования того же пикселя CU 180 как беглого пикселя (т.е., непосредственно квантуя значение пикселя и передавая квантованное значение для пикселя). В этом примере, определение удалить значение индекса 1 и соответствующую запись 202 палитры 202 из палитр 184 уменьшает размер палитры, например, палитры 184 будут включать в себя только записи 204 и 206.

[0123] В одном примере, видеокодер 20 вычисляет первые издержки на оптимизацию по скорости-искажению для кодирования пикселя текущего блока с использованием заданного значения индекса, которое соответствует заданной одной из записей палитры, которая указывает цветное значение для пикселя, например,  $RD(idx\_coding, t)$  вычисляется для каждого индекса палитры  $t$ , когда пиксели, увязанные с индексом



палитры  $t$ , кодируются с индексом  $t$ . Видеокодер 20 затем вычисляет вторые издержки на оптимизацию по скорости-искажению для кодирования того же пикселя текущего блока как беглого пикселя, например,  $RD(escape\_coding, t)$ , вычисляется для каждого индекса палитры  $t$ , когда пиксели, принадлежащие индексу палитры  $t$ , кодируются с помощью беглого режима, включающего в себя кодирование специального значения индекса и непосредственное квантование значений пикселей и передачу кодированных значений для пикселей.

[0124] Видеокодер 20 затем сравнивает вычисления первых и вторых издержек на оптимизацию по скорости-искажению чтобы определить, сохранить ли или удалить заданное значение индекса и соответствующую одну из записей палитры для палитры. Например, если первые издержки на оптимизацию по скорости-искажению выше, чем вторые издержки на оптимизацию по скорости-искажению видеокодер 20 может удалить заданный индекс и соответствующую одну из записей палитры из палитры, например, значение индекса 1 записи 202 из палитр 184. В этом примере, цветовое значение А записи 202 вероятно появляется нечасто в CU 180, т.е., имеет низкое количество ассоциированных пикселей CU 180, относительно других цветовых значений в палитрах 184.

[0125] В некоторых примерах, видеокодер 20 может выполнить это определение сохранения или удаления для каждой записи 202, 204, 206 палитры для палитр 184, но во многих случаях вероятно должны быть удалены только записи палитры с "низким появлением", которые имеют небольшое количество ассоциированных пикселей. Это потому, что издержки кодирования большей палитры, которая включает в себя записи палитры с низким появлением, и издержки кодирования пикселей с использованием больших значений индексов, которые соответствуют записям палитры с низким появлением, вероятно должны быть выше, чем кодирование нескольких пикселей как беглых пикселей. Таким образом, способы обеспечивают видеокодеру 20 возможность удаления одной или более записей палитры из первоначально полученной палитры, которые указывают цветовые значения, которые появляются нечасто в текущем блоке, который должен быть кодирован.

[0126] В качестве другого примера, видеокодер 20 может определить, объединить ли заданное значение индекса, которое соответствует заданной одной из записей палитры, например, значение индекса 1 записи 202 в палитрах 184, с соседним значением индекса, которое соответствует соседней одной из записей палитры, например, значением индекса 2 записи 204 в палитрах 184, на основе издержек на оптимизацию по скорости-искажению для кодирования пикселя CU 180 с использованием значения индекса 1 или с использованием соседнего значения индекса 2. В одном случае, соседнее значение индекса 2 записи 204 может быть ближайшим значением индекса в том, что касается наименьшего евклидова расстояния от заданного значения индекса 1 записи 202. В другом случае, соседнее значение индекса 2 записи 204 может быть ближайшим значением индекса в том, что касается порядка индексов относительно заданного значения индекса 1 записи 202. В этом примере, определение объединить заданное значение индекса 1 и соответствующую запись 202 палитры для палитр 184 с соседним значением индекса 2 и соответствующей соседней записью 204 палитры для палитр 184 уменьшает размер палитры, например, палитры 184 будут только включать в себя записи 204 и 206.

[0127] В одном примере, видеокодер 20 вычисляет первые издержки на оптимизацию по скорости-искажению для кодирования пикселя текущего блока с использованием заданного значения индекса, которое соответствует заданной одной из записей палитры,

которая указывает первое цветовое значение для пикселя, например,  $RD(separate\_coding, t)$  вычисляется для каждого индекса палитры  $t$ , когда пиксели, увязанные с индексом палитры  $t$ , кодируются с индексом  $t$ . Видеокодер 20 затем вычисляет вторые издержки на оптимизацию по скорости-искажению для кодирования того же пикселя текущего блока с использованием соседнего значения индекса, которое соответствует соседней одной из записей палитры, которая указывает второе цветовое значение для пикселя, например,  $RD(merged\_coding, t \rightarrow m)$  вычисляется для индекса  $t$  палитры, когда пиксели, увязанные с индексом  $t$  палитры, кодированы с соседним индексом  $m$  палитры.

[0128] Видеокодер 20 затем сравнивает вычисления первых и вторых издержек на оптимизацию по скорости-искажению чтобы определить, объединить ли заданное значение индекса и соответствующую одну из записей палитры для палитры с соседним значением индекса и соседней одной из записей палитры из палитры. Например, если первые издержки на оптимизацию по скорости-искажению выше, чем вторые издержки на оптимизацию по скорости-искажению видеокодер 20 может объединить заданный индекс и соответствующую одну из записей палитры, например, значение индекса 1 записи 202, с соседним значением индекса и соседней одной из записей палитры, например, значением индекса 2 записи 204. В этом примере, первое цветовое значение А заданной записи 202 палитры и второе цветовое значение В соседней записи 204 палитры вероятно являются по существу аналогичными значениями. Например, первое цветовое значение А и второе цветовое значение В, когда квантованы, могут быть одинаковым цветовым значением.

[0129] В некоторых примерах, объединение заданного значения индекса 1 записи 202 с соседним значением индекса 2 записи 204 может привести к удалению записи 202 из палитр 184 и увязыванию любых пикселей, которые ранее увязаны с первым цветовым значением А заданной записи 202, со вторым цветовым значением В соседней записи 204. В других примерах, объединение заданного значения индекса 1 записи 202 с соседним значением индекса 2 записи 204 может привести к удалению из палитр 184 как заданной записи 202, так и соседней записи 204, и добавлению новой записи палитры к палитрам 184. Новая запись палитры может указывать цветовое значение, которое является средним, или некоторую другую комбинацию первого цветового значения А заданной записи 202 и второго цветового значения В соседней записи 204. В этом примере, любые пиксели, которые ранее увязаны с записью 202 или записью 204, будут увязаны с новой записью палитры.

[0130] В некоторых примерах, видеокодер 20 может выполнить это определение объединения для каждой записи 202 204, 206 палитры для палитр 184, но, во многих случаях, вероятно должны быть объединены только "очень схожие" записи палитры, которые указывают по существу аналогичные цветовые значения (т.е., цветовые значения, которые могут быть квантованы в одинаковое цветовое значение). Это потому, что издержки кодирования большей палитры, которая включает в себя очень схожие записи палитры, и издержки кодирования пикселей с использованием отдельных значений индексов, которые соответствуют каждой из очень схожих записей палитры, вероятно должны быть выше, чем кодирование пикселей с использованием единого, объединенного значения индекса. Таким образом, способы обеспечивают видеокодеру 20 возможность объединения одной или более записей палитры в первоначально полученной палитре, которые указывают по существу аналогичные цветовые значения в текущем блоке, который должен быть кодирован.

[0131] В качестве дополнительного примера, видеокодер 20 может определить, изменить ли одну или более из записей палитры для палитры, например, записи 202,

204, 206 в палитрах 184, на основе репрезентативной палитры, определенной с использованием способа кластеризации k-средних. Видеокодер 20 может использовать способ кластеризации k-средних для вычисления записей палитры для репрезентативной палитры для каждого индекса палитры, где число итераций способа кластеризации k-средних основано на пороге. Например, для каждой из записей палитры первоначально полученной палитры, например, записей 202, 204, 206 в палитрах 184, видеокодер 20 вычисляет среднее значение пикселя одного или более пикселей CU 180, которые увязаны с соответствующим цветовым значением, например, цветовым значением А, В, С, указанным посредством соответствующей одной из записей палитры, например, записей 202, 204, 206. Видеокодер 20 затем определяет репрезентативную палитру для CU 180, включающую в себя репрезентативные записи палитры, указывающие репрезентативные средние значения пикселей.

[0132] В некоторых случаях, видеокодер 20 может выполнить другую, вторую итерацию способа кластеризации k-средних, включающую в себя увязывание или классификацию пикселей блока 180 CU с использованием репрезентативной палитры, и снова вычисление среднего значения пикселя для каждой из репрезентативных записей палитры, для того, чтобы получить итоговую версию репрезентативной палитры. В некоторых случаях, видеокодер 20 может выполнить некоторое число итераций способа кластеризации k-средних, которое определено пороговым значением. Пороговое значение может быть задано согласно допустимым вычислительным издержкам на видеокодере 20.

[0133] После определения репрезентативной палитры для блока видеоданных, которая включает в себя репрезентативные записи палитры, указывающие репрезентативные средние значения пикселей, видеокодер 20 выполняет поиск значения ближайшего пикселя блока, увязанного с каждым из индексов палитры, и ближайшее значение пикселя рассматривается как цветовое значение в итоговой палитре для репрезентативного одного из индексов палитры. Например, для заданной одной из репрезентативных записей палитры для репрезентативной палитры, видеокодер 20 определяет значение пикселя одного из пикселей, увязанных с заданной одной из репрезентативных записей палитры, которое является ближайшим к среднему значению пикселя, указанному заданной одной из репрезентативных записей палитры. На основе ближайшего значения пикселя, отличающегося от цветового значения, указанного соответствующей одной из записей палитры для палитры, видеокодер 20 выбирает ближайшее значение пикселя в качестве итогового цветового значения, указанного заданной одной из записей палитры для палитры.

[0134] Таким образом, видеокодер 20 определяет изменить заданную одну из записей палитры, например, запись 202 палитры в первоначальных палитрах 184, на основе ближайшего значения пикселя к среднему значению пикселя в репрезентативной палитре, отличающегося от цветового значения А, указанного заданной записью 202 палитры в первоначально полученных палитрах 184. Если ближайшее значение пикселя, определенное на основе репрезентативной палитры, и цветовое значение А, указанное заданной записью 202 палитры в первоначально полученных палитрах 184, одинаковы, то видеокодеру 20 не требуется изменять заданную запись 202 палитры в первоначальных палитрах 184.

[0135] Дополнительно, в соответствии со способами этого раскрытия, видеокодер 20 и/или видеодекодер 30 может определить, отключить ли фильтрование блоков, кодированных на основе палитры, например, CU 180 или CU 188. Например, видеокодер 20 и/или видеодекодер 30 может отключить фильтрование, такое как фильтрование для

удаления блочности и/или фильтрование с адаптивным смещением выборки (SAO), реконструированных версий CU 180, кодированной с помощью палитры, перед сохранением в буфере декодированных изображений для дальнейшего использования в качестве опорного блока или, в случае видеodeкодера 30, вывода для отображения пользователю. В некоторых примерах, видеodeкодер 20 и/или видеodeкодер 30 может только отключить фильтрование для удаления блочности CU 180. В других примерах, видеodeкодер 20 и/или видеodeкодер 30 может отключить SAO-фильтрование CU 180. В еще одних примерах, видеodeкодер 20 и/или видеodeкодер 30 может отключить как фильтрование для удаления блочности, так и SAO-фильтр, и любые другие типы фильтрования, для CU 180.

[0136] Обычно, блоки, кодированные на основе палитры, интерпретировались также как блоки, кодированные посредством внешнего кодирования, для которых фильтрование автоматически применяется к реконструированным блокам перед сохранением в буфере декодированных изображений или выводом для отображения. Согласно раскрытым способам, видеodeкодер 20 и/или видеodeкодер 30 может не применять фильтрование к блокам, кодированным на основе палитры, например, CU 180 и CU 188, но может продолжить применение фильтрования к блокам, кодированным посредством внешнего кодирования, например, внешнему блоку 200. Например, видеodeкодер 20 и/или видеodeкодер 30 может определить, что реконструированный блок является блоком, кодированным на основе палитры, например, CU 180 или CU 188, и отключить фильтрование реконструированного блока на основе определения режима кодирования на основе палитры. В некоторых случаях, способы могут добавить условное утверждение или проверку в синтаксис уровня CU, так чтобы если текущая CU является блоком, кодированным на основе палитры, то фильтрование отключалось для текущей CU.

[0137] Хотя способы по Фиг. 4 описаны в контексте CU (HEVC), следует понимать, что способы могут также быть применены к единицам предсказания (PU) или в других процессах и/или стандартах кодирования видео.

[0138] Фиг. 5 является схемой последовательности операций, иллюстрирующей примерное функционирование видеodeкодера, определяющего фильтрование блоков, кодированных на основе палитры, в соответствии со способами этого раскрытия. Примерное функционирование, проиллюстрированное на Фиг. 5 может быть выполнено либо посредством видеodeкодера 20 с Фиг. 2, либо видеodeкодера 30 с Фиг. 3.

[0139] Примерное функционирование по Фиг. 5 будет сначала описано в отношении видеodeкодера 30. Видеodeкодер 30 принимает кодированный битовый поток от видеodeкодера, такого как видеodeкодер 20, который включает в себя кодированные блоки видеоданных по меньшей мере для одного изображения и один или более синтаксических элементов, ассоциированных с видеоданными. Видеodeкодер 30 определяет, что текущий блок видеоданных, который должен быть декодирован, является блоком, кодированным на основе палитры (220). В некоторых примерах, видеodeкодер 30 может принять по меньшей мере один синтаксический элемент (например, флаг) в битовом потоке, который указывает, является ли каждый блок видеоданных блоком, кодированным на основе палитры. В других примерах, видеodeкодер 30 может принять один или более синтаксических элементов, указывающих тип кодирования, используемый для кодирования каждого блока видеоданных, например, является ли каждый блок блоком, кодированным на основе палитры, блоком, кодированным посредством внешнего кодирования, или блоком, кодированным посредством внутреннего кодирования и т.д.

[0140] Когда текущий блок, который должен быть декодирован, является блоком,

кодированным на основе палитры, блок 165 декодирования на основе палитры видеodeкодера 30 определяет палитру для текущего блока, где палитра включает в себя одну или более записей палитры, которые указывают одно или более соответствующих цветовых значений (222). Как описано более подробно ниже, соответствующие цветовые значения, включенные в палитру, могут быть главными цветовыми значениями, которые в текущем блоке возникают наиболее часто. Блок 165 декодирования на основе палитры может определить палитру согласно размеру палитры и записям палитры, принятым в кодированном битовом потоке. Блок 165 декодирования на основе палитры затем определяет значения индексов для одного или более пикселей текущего блока, принятого в кодированном битовом потоке, прием каждое из значений индексов, соответствующее одной из записей палитры, которая указывает цветовое значение для одного из пикселей текущего блока (224).

[0141] Видеodeкодер 30 реконструирует текущий блок видеоданных на основе определенной палитры и определенных значений индексов для текущего блока (226). Например, видеodeкодер 30 может увязать определенные значения индексов с записями палитры для реконструирования значений пикселей текущего блока. Видеodeкодер 30 может затем сохранить видеоданные реконструированного блока в буфере 162 декодированных изображений и впоследствии вывести видеоданные реконструированного блока для отображения.

[0142] Обычно, блоки, кодированные на основе палитры, интерпретировались также как блоки, кодированные посредством внешнего кодирования, и в этой связи фильтрация автоматически применялась к реконструированным блокам перед сохранением в буфере декодированных изображений или выводом для отображения. Согласно раскрытым способам, видеodeкодер 30 определяет, на основе текущего блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока перед сохранением или выводом реконструированного блока (228). В некоторых примерах, видеodeкодер 30 может отключить только фильтрацию для удаления блочности текущего блока, кодированного на основе палитры, посредством блока 160 фильтрации видеodeкодера 30. В других примерах, видеodeкодер 30 может отключить SAO-фильтрацию текущего блока, кодированного на основе палитры. В еще других примерах, видеodeкодер 30 может отключить все типы фильтрации, выполняемые блоком 160 фильтрации для текущего блока, кодированного на основе палитры.

[0143] Примерное функционирование по Фиг. 5 теперь будет описано в отношении видеodeкодера 20. Видеodeкодер 20 определяет, что текущий блок видеоданных должен быть декодирован как блок, кодированный на основе палитры (220). В некоторых примерах, видеodeкодер 20 может сигнализировать по меньшей мере один синтаксический элемент (например, флаг) в кодированном битовом потоке, который указывает, является ли каждый блок видеоданных блоком, кодированным на основе палитры. В других примерах, видеodeкодер 20 может сигнализировать один или более синтаксических элементов в кодированном битовом потоке, указывающих тип кодирования, используемый для кодирования каждого блока видеоданных, например, является ли каждый блок блоком, кодированным на основе палитры, блоком, кодированным посредством внешнего кодирования, или блоком, кодированным посредством внутреннего кодирования и т.д.

[0144] Когда текущий блок должен быть кодирован, как блок, кодированный на основе палитры, блок 122 кодирования на основе палитры видеodeкодера 20 определяет палитру для текущего блока, где палитра включает в себя одну или более записей

палитры, которые указывают одно или более соответствующих цветовых значений (222). Как описано более подробно ниже, соответствующие цветовые значения, включенные в палитру, могут быть главными цветовые значениями, которые в текущем блоке возникают наиболее часто. Блок 122 кодирования на основе палитры может  
 5 определить размер палитры и записи палитры для палитры с использованием способа кластеризации значений пикселей, такого как способ кластеризации, описанный более подробно выше относительно Фиг. 4. В соответствии со способами этого раскрытия, блок 122 кодирования на основе палитры может определить, изменить ли по меньшей мере одно из размера палитры или одну или более из записей палитры для палитры по  
 10 меньшей мере частично на основе издержек на оптимизацию по скорости-искажению. Определение изменения описывается более подробно ниже относительно Фиг. 6.

[0145] После определения палитры, блок 122 кодирования на основе палитры определяет значения индексов для одного или более пикселей текущего блока, причем каждое из значений индексов соответствующее одной из записей палитры, которая  
 15 указывает цветовое значение для одного из пикселей текущего блока (224). Например, видеокодер 20 может кодировать текущий блок видеоданных посредством локализации записей в палитре, которые представляют цвета одного или более из пикселей текущего блока, и кодирования блока со значениями индексов, которые указывают записи в палитре. Видеокодер 20 может сигнализировать палитру и значения индексов для  
 20 текущего блока, кодированного на основе палитры, в кодированном битовом потоке видеodeкодера, такому как видеodeкодер 30.

[0146] Видеокодер 20 затем реконструирует текущий блок видеоданных на основе определенной палитры и определенных значений индексов для текущего блока в цикле кодирования(226). Например, видеокодер 20 может увязать значения индексов с записями  
 25 палитры для реконструирования значений пикселей текущего блока. Видеокодер 20 может затем сохранить видеоданные реконструированного блока в буфере 116 декодированных изображений. Согласно раскрытым способам, видеокодер 20 определяет, на основе текущего блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока перед сохранением  
 30 реконструированного блока (228). В некоторых примерах, видеокодер 20 может отключить только фильтрацию для удаления блочности текущего блока, кодированного на основе палитры, посредством блока 114 фильтрации видеокодера 20. В других примерах, видеокодер 20 может отключить SAO-фильтрацию текущего блока, кодированного на основе палитры. В еще других примерах, видеокодер 20 может  
 35 отключить все типы фильтрации, выполняемые блоком 114 фильтрации для текущего блока, кодированного на основе палитры.

[0147] Фиг. 6 является схемой последовательности операций, иллюстрирующей примерное функционирование видеокодера, определяющего палитру для блока видеоданных, кодированного на основе палитры, например, этап 220 по Фиг. 5, в  
 40 соответствии со способами этого раскрытия. Примерная операция, проиллюстрированная на Фиг. 6, описана по отношению к видеокодеру 20 с Фиг. 2.

[0148] Как описано выше, когда текущий блок видеоданных должен быть кодирован как блок, кодированный на основе палитры, блок 122 кодирования на основе палитры видеокодера 20 определяет палитру для текущего блока, где палитра включает в себя  
 45 одну или более записей палитры, которые указывают одно или более соответствующих цветовых значений и которые соответствуют соответствующим значениям индексов. Для определения палитры, блок 122 кодирования на основе палитры видеокодера 20 сначала определяет размер палитры и записи палитры для палитры с использованием

способа кластеризации значений пикселей (230). Например, как описано более подробно выше относительно Фиг. 4, для текущего блока видеоданных, блок 122 кодирования на основе палитры может определить гистограмму, включающую в себя множество записей, причем каждая запись включает в себя соответствующее цветовое значение и соответствующее количество пикселей блока, которые увязываются с цветовым значением, отсортировать записи гистограммы в убывающем порядке на основе соответствующего количества пикселей для каждой из записей, так чтобы первая запись гистограммы включала в себя цветовое значение, имеющее наибольшее количество пикселей относительно других цветовых значений, и определить размер палитры и записи палитры для палитры на основе записей гистограммы.

[0149] В соответствии со способами этого раскрытия, после определения размера палитры и записей палитры для палитры, блок 122 кодирования на основе палитры может определить, изменить ли по меньшей мере одно из размера палитры или одной или более из записей палитры для палитры по меньшей мере частично на основе издержек на оптимизацию по скорости-искажению. Например, блок 122 кодирования на основе палитры может определить, сохранить ли или удалить заданное значение индекса, которое соответствует заданной одной из записей палитры для палитры, на основе издержек на оптимизацию по скорости-искажению для кодирования пикселя блока с использованием заданного значения индекса или как беглого пикселя (232). В этом примере, определение удалить заданное значение индекса и соответствующую одну из записей палитры из палитры уменьшает размер палитры.

[0150] В некоторых случаях, блок 122 кодирования на основе палитры может выполнить сравнение издержек на оптимизацию по скорости-искажению чтобы определить, удалить ли или сохранить заданное значение индекса, как следует ниже. Блок 122 кодирования на основе палитры вычисляет первые издержки на оптимизацию по скорости-искажению при кодировании пикселя текущего блока с использованием заданного значения индекса, которое соответствует заданной одной из записей палитры, которая указывает цветовое значение для пикселя. Блок 122 кодирования на основе палитры затем вычисляет вторые издержки на оптимизацию по скорости-искажению для кодирования того же пикселя текущего блока как беглого пикселя, включая кодирование специального значения индекса и квантованного значения пикселя для пикселя. Блок 122 кодирования на основе палитры определяет, сохранить ли или удалить заданное значение индекса и соответствующую одну из записей палитры для палитры, на основе сравнения вычислений первых и вторых издержек на оптимизацию по скорости-искажению. Например, если первые издержки на оптимизацию по скорости-искажению выше, чем вторые издержки на оптимизацию по скорости-искажению блок 122 кодирования на основе палитры может удалить заданный индекс и соответствующую одну из записей палитры из палитры. В этом примере, цветовое значение, указанное заданной одной из записей палитры, вероятно появляется нечасто в текущем блоке, т.е., имеет небольшое количество ассоциированных пикселей текущего блока, относительно других цветовых значений в палитре.

[0151] В другом примере, блок 122 кодирования на основе палитры может определить, объединить ли заданное значение индекса, которое соответствует заданной одной из записей палитры для палитры, с соседним значением индекса, которое соответствует соседней одной из записей палитры для палитры, на основе издержек на оптимизацию по скорости-искажению для кодирования пикселя блока с использованием заданного значения индекса или с использованием соседнего значения индекса (234). В одном примере, соседним значением индекса может быть значение ближайшего индекса, в

том, что касается наименьшего евклидова расстояния от заданного значения индекса. В другом примере, соседним значением индекса может быть значение ближайшего индекса, в том, что касается порядка индексов, относительно заданного значения индекса. В этом примере, определение объединить заданное значение индекса и соответствующую одну из записей палитры для палитры с соседним значением индекса и соседней одной из записей палитры для палитры уменьшает размер палитры.

[0152] В некоторых случаях, блок 122 кодирования на основе палитры может выполнить сравнение издержек на оптимизацию по скорости-искажению чтобы определить, объединить ли заданное значение индекса, как следует ниже. Блок 122 кодирования на основе палитры вычисляет первые издержки на оптимизацию по скорости-искажению при кодировании пикселя текущего блока с использованием заданного значения индекса, которое соответствует заданной одной из записей палитры, которая указывает первое цветовое значение для пикселя. Блок 122 кодирования на основе палитры затем вычисляет вторые издержки на оптимизацию по скорости-искажению для кодирования того же пикселя текущего блока с использованием соседнего значения индекса, которое соответствует соседней одной из записей палитры, которая указывает второе цветовое значение для пикселя. Блок 122 кодирования на основе палитры определяет, объединить ли заданное значение индекса и соответствующую одну из записей палитры для палитры с соседним значением индекса и соседней одной из записей палитры для палитры, на основе сравнения вычислений первых и вторых издержек на оптимизацию по скорости-искажению. Например, если первые издержки на оптимизацию по скорости-искажению выше, чем вторые издержки на оптимизацию по скорости-искажению блок 122 кодирования на основе палитры может объединить заданный индекс и соответствующую одну из записей палитры с соседним значением индекса и соседней одной из записей палитры. В этом примере, первое цветовое значение, указанное заданной одной из записей палитры, и второе цветовое значение, указанное соседней одной из записей палитры, вероятно по существу являются одинаковыми значениями. Например, первое цветовое значение и второе цветовое значение, когда квантованы, могут быть одинаковым цветовым значением.

[0153] В дополнительном примере, блок 122 кодирования на основе палитры может определить, изменить ли одну или более записей палитры для палитры, на основе репрезентативной палитры, определенной с использованием способа кластеризации k-средних (236). Например, как описано более подробно выше относительно Фиг. 4, для каждой из записей палитры первоначально полученной палитры, блок 122 кодирования на основе палитры вычисляет среднее значение пикселя одного или более пикселей блока, которые увязаны с соответствующим цветовым значением, указанным соответствующей одной из записей палитры. Блок 122 кодирования на основе палитры затем определяет репрезентативную палитру для блока видеоданных, включающую в себя репрезентативные записи палитры, указывающие соответствующие средние значения пикселей. в некоторых случаях, блок 122 кодирования на основе палитры может выполнить некоторое число итераций способа кластеризации k-средних, которое определено пороговым значением.

[0154] После определения репрезентативной палитры для блока видеоданных, которая включает в себя репрезентативные записи палитры, указывающие соответствующие средние значения пикселей, блок 122 кодирования на основе палитры определяет, для заданной одной из репрезентативных записей палитры для репрезентативной палитры, значение пикселя одного из пикселей, увязанных с заданной одной из репрезентативных записей палитры, которое является ближайшим к среднему значению пикселя,



указанному заданной одной из репрезентативных записей палитры. На основе ближайшего значения пикселя, отличающегося от цветового значения, указанного соответствующей одной из записей палитры для палитры, блок 122 кодирования на основе палитры выбирает ближайшее значение пикселя в качестве итогового цветового значения, указанного заданной одной из записей палитры для палитры. Таким образом, блок 122 кодирования на основе палитры определяет изменить заданную одну или более записей палитры, на основе сравнения ближайшего значения пикселя со средним значением пикселя в репрезентативной палитре и цветового значения, указанного заданной одной из записей палитры в первоначально полученной палитре.

[0155] Следует понимать, что в зависимости от примера, некоторые аспекты или события по любому из способов, описанных в настоящем документе, могут быть выполнены в другой последовательности, могут быть добавлены, объединены или оставлены все вместе (например, не все описанные действия или события необходимы для применения способов на практике). Более того, в некоторых примерах, действия или события могут быть выполнены параллельно, а не последовательно, например, посредством многопоточковой обработки, обработки прерываний, или многочисленных процессоров. В дополнение, хотя в целях ясности некоторые аспекты этого раскрытия описаны как выполняемые одиночным модулем или блоком, следует понимать, что способы этого раскрытия могут быть выполнены посредством комбинации блоков или модулей, ассоциированных с видеокодером.

[0156] Некоторые аспекты этого раскрытия в целях иллюстрации были описаны относительно разрабатываемого стандарта HEVC. Однако, способы, описанные в этом раскрытии, могут быть полезны для других процессов кодирования видео, в том числе другого стандарта или проприетарных процессов кодирования видео, еще не разработанных.

[0157] Способы, описанные выше могут быть выполнены видеокодером 20 (Фиг. 1 и 2) и/или видеodeкодером 30 (Фиг. 1 и 3), каждый из которых может быть в общем назван видеокодером. Аналогично, кодирование видео может относиться к кодированию видео или декодированию видео, в зависимости от случая.

[0158] Хотя конкретные комбинации различных аспектов способов описаны выше, эти комбинации предоставлены лишь для иллюстрации примеров способов, описанных в этом раскрытии. Соответственно, способы этого раскрытия не должны ограничиваться этими примерными комбинациями и могут охватывать любую мыслимую комбинацию различных аспектов способов, описанных в этом раскрытии.

[0159] В одном или более примерах, описанные функции могут быть реализованы в аппаратных средствах, программном обеспечении, программно-аппаратных средствах или любой их комбинации. В случае реализации в программном обеспечении, функции могут храниться как одна или более инструкций или код на компьютерно-читаемом носителе или передаваться на него и могут быть исполнены аппаратным блоком обработки. Компьютерно-читаемые носители могут включать в себя компьютерно-читаемые носители информации, которые соответствуют материальному носителю, такому как носители данных, или среды связи, включающие в себя любую среду, которая способствует передаче компьютерной программы из одного места в другое, например, согласно протоколу связи. Таким образом, компьютерно-читаемые носители в общем могут соответствовать (1) материальным компьютерно-читаемым носителям информации, которые являются постоянными, или (2) среде связи, такой как сигнал или несущая волна. Носителями данных могут являться любые доступные носители, к которым может быть осуществлен доступ посредством одного или более компьютеров,

или одного или более процессоров для извлечения инструкций, кода и/или структур данных для реализации способов, описанных в этом раскрытии. Компьютерный программный продукт может включать в себя компьютерно-читаемый носитель.

[0160] В качестве примера, и а не ограничения, такие компьютерно-читаемые носители информации могут содержать RAM, ROM, EEPROM, CD-ROM или другое хранилище на оптических дисках, хранилище на магнитных дисках или другие магнитные устройства хранения, флэш-память, или любой другой носитель, который может быть использован для хранения желаемого программного кода в форме инструкций или структур данных, и к которому может быть осуществлен доступ посредством компьютера. Также, любое соединение правильно называть компьютерно-читаемым носителем. Например, если инструкции передаются с Web-сайта, сервера или другого удаленного источника с использованием коаксиального кабеля, оптоволоконного кабеля, витой пары, цифровой абонентской линией связи (DSL) или беспроводных технологий, таких как инфракрасная связь, радио и микроволновая, то коаксиальный кабель, оптоволоконный кабель, витая пара, DSL или беспроводные технологии, такие как инфракрасная связь, радио и микроволновая, включаются в определение носителя. Однако следует понимать, что компьютерно-читаемые среды хранения и среды хранения данных не включают в себя соединения, несущие волны, сигналы или другие промежуточные среды, но вместо этого направлены на непромежуточные, материальные среды хранения. Термин "диск" в настоящем документе используется в отношении и магнитных, и оптических дисков, которые включают в себя компакт-диск (CD), лазерный диск, оптический диск, универсальный цифровой диск (DVD), флоппи-диск и blu-ray диск, причем магнитные диски обычно воспроизводят данные магнитным образом, в то время как оптические диски воспроизводят данные оптически с помощью лазера. Комбинации вышеуказанного должны также быть включены в объем компьютерно-читаемых носителей.

[0161] Инструкции могут быть исполнены одним или более процессорами, такими как один или более процессоров цифровой обработки сигналов (DSP), микропроцессоров общего назначения, специализированных интегральных микросхем (ASIC), программируемых пользователем вентильных матриц (FPGA), или другими эквивалентными схемами интегрированной или дискретной логики. Следовательно, термин "процессор", как используется в настоящем документе, может относиться к любой из вышеупомянутых структур или любой другой структуре, подходящей для реализации способов, описанных в настоящем документе. К тому же, в некоторых аспектах, описанная в настоящем документе функциональность может быть предоставлена внутри выделенных аппаратных средств и/или программных модулей, сконфигурированных для кодирования и декодирования, или включена в объединенный кодек. Также, способы могут быть полностью реализованы в одной и больше схемах или логических элементах.

[0162] Способы этого раскрытия могут быть реализованы в широком спектре устройств или аппаратов, включающих в себя беспроводную гарнитуру, интегральную схему (IC) или набор IC (например, набор микросхем). Различные компоненты, модули или блоки описаны в этом раскрытии, чтобы подчеркнуть функциональные аспекты устройств, сконфигурированных для выполнения раскрытых способов, но необязательно требуют реализацию посредством разных аппаратных блоков. Скорее, как описано выше, различные блоки могут быть объединены в аппаратном блоке кодека или предоставлены группой взаимодействующих аппаратных блоков, включающих в себя один или более процессоров, как описано выше, совместно с подходящим программным обеспечением и/или программно-аппаратными средствами.

[0163] Были описаны различные примеры. Эти и другие примеры входят в объем нижеследующих пунктов формулы изобретения.

(57) Формула изобретения

- 5 1. Способ обработки видеоданных, содержащий этапы, на которых:
  - определяют, что блок видеоданных является блоком, кодированным на основе палитры;
  - определяют палитру для блока, включающую в себя одну или более записей палитры, которые указывают одно или более соответственных цветовых значений;
  - 10 определяют значения индексов для одного или более пикселей блока, причем каждое из значений индексов соответствует одной из записей палитры, которая указывает цветовое значение для одного из пикселей блока;
  - реконструируют блок видеоданных на основе палитры и значений индексов для блока и
  - 15 определяют на основе блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока.
2. Способ по п. 1, в котором этап, на котором определяют отключить фильтрацию реконструированного блока, содержит этап, на котором отключают по меньшей мере одно из фильтрации реконструированного блока для удаления блочности и
- 20 фильтрации реконструированного блока с адаптивным смещением выборки (SAO).
3. Способ по п. 1, в котором этап, на котором реконструируют блок, содержит этап, на котором реконструируют блок видеоданных посредством видеодекодера, причем способ дополнительно содержит, на основе отключенного фильтрации, по меньшей мере одно из этапа, на котором выводят видеоданные посредством видеодекодера для
- 25 отображения, или этапа, на котором сохраняют видеоданные посредством видеодекодера в буфере декодированных изображений без фильтрации реконструированного блока видеоданных.
4. Способ по п. 1, в котором этап, на котором реконструируют блок, содержит этап, на котором реконструируют блок видеоданных посредством видеокодера, причем
- 30 способ дополнительно содержит, на основе отключенного фильтрации, этап, на котором сохраняют видеоданные посредством видеокодера в буфере декодированных изображений без фильтрации реконструированного блока видеоданных.
5. Способ по п. 1, в котором этап, на котором определяют палитру для блока, содержит этапы, на которых:
  - 35 определяют, посредством видеокодера, размер палитры и записи палитры для палитры с использованием способа кластеризации значений пикселей и
  - определяют, изменить ли по меньшей мере одно из размера палитры и одной или более из записей палитры для палитры по меньшей мере частично на основе издержек на оптимизацию по скорости-искажению.
- 40 6. Способ по п. 5, в котором этап, на котором определяют, изменить ли размер палитры, содержит этап, на котором определяют, сохранить ли или удалить заданное значение индекса, которое соответствует заданной одной из записей палитры для палитры, при этом удаление заданного значения индекса и соответствующей одной из записей палитры для палитры уменьшает размер палитры.
- 45 7. Способ по п. 6, дополнительно содержащий этапы, на которых:
  - вычисляют первые издержки на оптимизацию по скорости-искажению для кодирования по меньшей мере одного пикселя блока с использованием заданного значения индекса, которое соответствует заданной одной из записей палитры, которая

указывает цветовое значение для этого по меньшей мере одного пикселя;

вычисляют вторые издержки на оптимизацию по скорости-искажению для кодирования данного по меньшей мере одного пикселя блока как беглого пикселя, в том числе кодируют квантованное значение пикселя для этого по меньшей мере одного пикселя; и

определяют, сохранить ли или удалить заданное значение индекса и соответствующую одну из записей палитры для палитры, на основе сравнения вычислений первых и вторых издержек на оптимизацию по скорости-искажению.

8. Способ по п. 5, в котором этап, на котором определяют, изменить ли размер палитры, содержит этап, на котором определяют, объединить ли заданное значение индекса, которое соответствует заданной одной из записей палитры для палитры, с соседним значением индекса, которое соответствует соседней одной из записей палитры для палитры, при этом объединение заданного значения индекса и соответствующей одной из записей палитры для палитры с соседним значением индекса и соседней одной из записей палитры для палитры уменьшает размер палитры.

9. Способ по п. 8, в котором соседним значением индекса является одно из значения ближайшего индекса в смысле наименьшего евклидова расстояния от заданного значения индекса или значения ближайшего индекса в смысле порядка индексов относительно заданного значения индекса.

10. Способ по п. 8, дополнительно содержащий этапы, на которых:

вычисляют первые издержки на оптимизацию по скорости-искажению для кодирования по меньшей мере одного пикселя блока с использованием заданного значения индекса, которое соответствует заданной одной из записей палитры, которая указывает первое цветовое значение для этого по меньшей мере одного пикселя;

вычисляют вторые издержки на оптимизацию по скорости-искажению для кодирования данного по меньшей мере одного пикселя блока с использованием соседнего значения индекса, которое соответствует соседней одной из записей палитры, которая указывает второе цветовое значение для этого по меньшей мере одного пикселя, при этом второе цветовое значение аналогично первому цветовому значению; и

определяют, объединить ли заданное значение индекса и соответствующую одну из записей палитры для палитры с соседним значением индекса и соседней одной из записей палитры для палитры, на основе сравнения вычислений первых и вторых издержек на оптимизацию по скорости-искажению.

11. Способ по п. 5, в котором этап, на котором определяют, изменить ли одну или более записей палитры для палитры, содержит этапы, на которых:

определяют репрезентативную палитру для блока видеоданных с использованием способа кластеризации k-средних, причем репрезентативная палитра включает в себя репрезентативные записи палитры, указывающие соответственные средние значения пикселей;

определяют, для заданной одной из репрезентативных записей палитры для репрезентативной палитры, значение пикселя одного из пикселей, увязанных с заданной одной из репрезентативных записей палитры, которое является ближайшим к среднему значению пикселя, указанному заданной одной из репрезентативных записей палитры; и

на основе того, что ближайшее значение пикселя отличается от цветового значения, указанного соответствующей одной из записей палитры для палитры, выбирают ближайшее значение пикселя в качестве итогового цветового значения, указанного заданной одной из записей палитры для палитры.

12. Способ по п. 11, в котором этап, на котором определяют репрезентативную палитру с использованием способа кластеризации k-средних, содержит этапы, на которых:

вычисляют, для каждой из записей палитры для палитры, среднее значение пикселя одного или более пикселей блока, которые увязаны с соответственным цветовым значением, указанным соответственной одной из записей палитры; и

определяют репрезентативную палитру для блока видеоданных, включающую в себя репрезентативные записи палитры, включающие в себя соответственные средние значения пикселей, при этом число итераций способа кластеризации k-средних определяется посредством порогового значения.

13. Способ по п. 5, в котором этап, на котором определяют размер палитры и записи палитры для палитры с использованием способа кластеризации значений пикселей, содержит этапы, на которых:

определяют для блока видеоданных гистограмму, включающую в себя множество записей, причем каждая запись включает в себя соответственное цветовое значение и соответственное количество пикселей блока, которые увязываются с цветовым значением;

сортируют записи гистограммы в убывающем порядке на основе соответственного количества пикселей для каждой из записей, так чтобы первая запись гистограммы включала в себя цветовое значение, имеющее наибольшее количество пикселей относительно других цветовых значений; и

определяют размер палитры и записи палитры для палитры на основе записей гистограммы.

14. Устройство обработки видео, содержащее:

память, приспособленную для хранения видеоданных; и

один или более процессоров, находящихся на связи с памятью, выполненных с возможностью:

определения того, что блок видеоданных является блоком, кодированным на основе палитры;

определения палитры для блока, включающей в себя одну или более записей палитры, которые указывают одно или более соответственных цветовых значений;

определения значений индексов для одного или более пикселей блока, причем каждое из значений индексов соответствует одной из записей палитры, которая указывает цветовое значение для одного из пикселей блока;

реконструирования блока видеоданных на основе палитры и значений индексов для блока и

определения на основе блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока.

15. Устройство по п. 14, в котором один или более процессоров выполнены с возможностью отключения по меньшей мере одного из фильтрации реконструированного блока для удаления блочности и фильтрации реконструированного блока с адаптивным смещением выборки (SAO).

16. Устройство по п. 14, при этом устройство содержит видеodeкодер, причем один или более процессоров видеodeкодера выполнены с возможностью, на основе отключенного фильтрации, реконструировать блок видеоданных и выполнять по меньшей мере одно из вывода видеоданных для отображения или сохранения видеоданных в буфере декодированных изображений без фильтрации реконструированного блока видеоданных.

17. Устройство по п. 14, при этом устройство содержит видеокодер, причем один или более процессоров видеокодера выполнены с возможностью, на основе отключенного фильтрования, реконструировать блок видеоданных и сохранять видеоданные в буфере декодированных изображений без фильтрования

5 реконструированного блока видеоданных.

18. Устройство по п. 14, при этом устройство содержит видеокодер, причем для определения палитры для блока один или более процессоров видеокодера выполнены с возможностью:

10 определения размера палитры и записей палитры для палитры с использованием способа кластеризации значений пикселей и

определения того, изменить ли по меньшей мере одно из размера палитры и одной или более из записей палитры для палитры по меньшей мере частично на основе издержек на оптимизацию по скорости-искажению.

19. Устройство по п. 18, в котором для определения того, изменить ли размер палитры, один или более процессоров выполнены с возможностью определения того, сохранить ли или удалить заданное значение индекса, которое соответствует заданной одной из записей палитры для палитры, при этом удаление заданного значения индекса и соответствующей одной из записей палитры для палитры уменьшает размер палитры.

20. Устройство по п. 19, в котором один или более процессоров выполнены с

20 возможностью:

вычисления первых издержек на оптимизацию по скорости-искажению для кодирования по меньшей мере одного пикселя блока с использованием заданного значения индекса, которое соответствует заданной одной из записей палитры, которая указывает цветное значение для этого по меньшей мере одного пикселя;

25 вычисления вторых издержек на оптимизацию по скорости-искажению для кодирования данного по меньшей мере одного пикселя блока как беглого пикселя, включая кодирование квантованного значения пикселя для этого по меньшей мере одного пикселя; и

30 определения того, сохранить ли или удалить заданное значение индекса и соответствующую одну из записей палитры для палитры, на основе сравнения вычислений первых и вторых издержек на оптимизацию по скорости-искажению.

21. Устройство по п. 18, в котором для определения того, изменить ли размер палитры, один или более процессоров выполнены с возможностью определения того, объединить ли заданное значение индекса, которое соответствует заданной одной из записей палитры для палитры, с соседним значением индекса, которое соответствует соседней одной из записей палитры для палитры, при этом объединение заданного значения индекса и соответствующей одной из записей палитры для палитры с соседним значением индекса и соседней одной из записей палитры для палитры уменьшает размер палитры.

22. Устройство по п. 21, в котором соседним значением индекса является одно из значения ближайшего индекса в смысле наименьшего евклидова расстояния от заданного значения индекса или значения ближайшего индекса в смысле порядка индексов относительно заданного значения индекса.

23. Устройство по п. 21, в котором один или более процессоров выполнены с

45 возможностью:

вычисления первых издержек на оптимизацию по скорости-искажению для кодирования по меньшей мере одного пикселя блока с использованием заданного значения индекса, которое соответствует заданной одной из записей палитры, которая указывает первое цветное значение для этого по меньшей мере одного пикселя;

вычисления вторых издержек на оптимизацию по скорости-искажению для кодирования данного по меньшей мере одного пикселя блока с использованием соседнего значения индекса, которое соответствует соседней одной из записей палитры, которая указывает второе цветовое значение для этого по меньшей мере пикселя, при этом второе цветовое значение аналогично первому цветовому значению; и

определения того, объединить ли заданное значение индекса и соответствующую одну из записей палитры для палитры с соседним значением индекса и соседней одной из записей палитры для палитры, на основе сравнения вычислений первых и вторых издержек на оптимизацию по скорости-искажению.

24. Устройство по п. 18, в котором для определения того, изменить ли одну или более записей палитры для палитры, один или более процессоров выполнены с возможностью:

определять репрезентативную палитру для блока видеоданных с использованием способа кластеризации k-средних, причем репрезентативная палитра включает в себя репрезентативные записи палитры, указывающие соответственные средние значения пикселей;

определять, для заданной одной из репрезентативных записей палитры для репрезентативной палитры, значение пикселя одного из пикселей, увязанных с заданной одной из репрезентативных записей палитры, которое является ближайшим к среднему значению пикселя, указанному заданной одной из репрезентативных записей палитры;

и

на основе того, что ближайшее значение пикселя отличается от цветового значения, указанного соответствующей заданной одной из записей палитры для палитры, выбирать ближайшее значение пикселя в качестве итогового цветового значения, указанного заданной одной из записей палитры для палитры.

25. Устройство по п. 24, в котором, для определения репрезентативной палитры с использованием способа кластеризации k-средних, один или более процессоров выполнены с возможностью:

вычислять, для каждой из записей палитры для палитры, среднее значение пикселя одного или более пикселей блока, которые увязаны с соответствующим цветовым значением, указанным соответствующей одной из записей палитры; и

определять репрезентативную палитру для блока видеоданных, включающую в себя репрезентативные записи палитры, включающие в себя соответствующие средние значения пикселей, при этом число итераций способа кластеризации k-средних определяется посредством порогового значения.

26. Устройство по п. 18, в котором, для определения размера палитры и записей палитры для палитры с использованием способа кластеризации значений пикселей, один или более процессоров выполнены с возможностью:

определять для блока видеоданных гистограмму, включающую в себя множество записей, причем каждая запись включает в себя соответственное цветовое значение и соответственное количество пикселей блока, которые увязываются с цветовым значением;

сортировать записи гистограммы в убывающем порядке на основе соответственного количества пикселей для каждой из записей, так чтобы первая запись гистограммы включала в себя цветовое значение, имеющее наибольшее количество пикселей относительно других цветовых значений; и

определять размер палитры и записи палитры для палитры на основе записей гистограммы.

27. Устройство по п. 14, при этом устройство содержит по меньшей мере одно из:

интегральной схемы;  
микропроцессора или  
устройства беспроводной связи.

28. Устройство обработки видеоданных, содержащее:

- 5 средство для определения того, что блок видеоданных является блоком, кодированным на основе палитры;  
средство для определения палитры для блока, включающей в себя одну или более записей палитры, которые указывают одно или более соответственных цветовых значений;
- 10 средство для определения значений индексов для одного или более пикселей блока, причем каждое из значений индексов соответствует одной из записей палитры, которая указывает цветовое значение для одного из пикселей блока;  
средство для реконструирования блока видеоданных на основе палитры и значений индексов для блока и
- 15 средство для определения на основе блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока.
29. Долговременный машиночитаемый носитель, на котором сохранены инструкции для обработки видеоданных, которые при их исполнении предписывают одному или более процессорам:
- 20 определять, что блок видеоданных является блоком, кодированным на основе палитры;  
определять палитру для блока, включающую в себя одну или более записей палитры, которые указывают одно или более соответственных цветовых значений;  
определять значения индексов для одного или более пикселей блока, причем каждое
- 25 из значений индексов соответствует одной из записей палитры, которая указывает цветовое значение для одного из пикселей блока;  
реконструировать блок видеоданных на основе палитры и значений индексов для блока и
- 30 определять на основе блока, являющегося блоком, кодированным на основе палитры, отключить фильтрацию реконструированного блока.

35

40

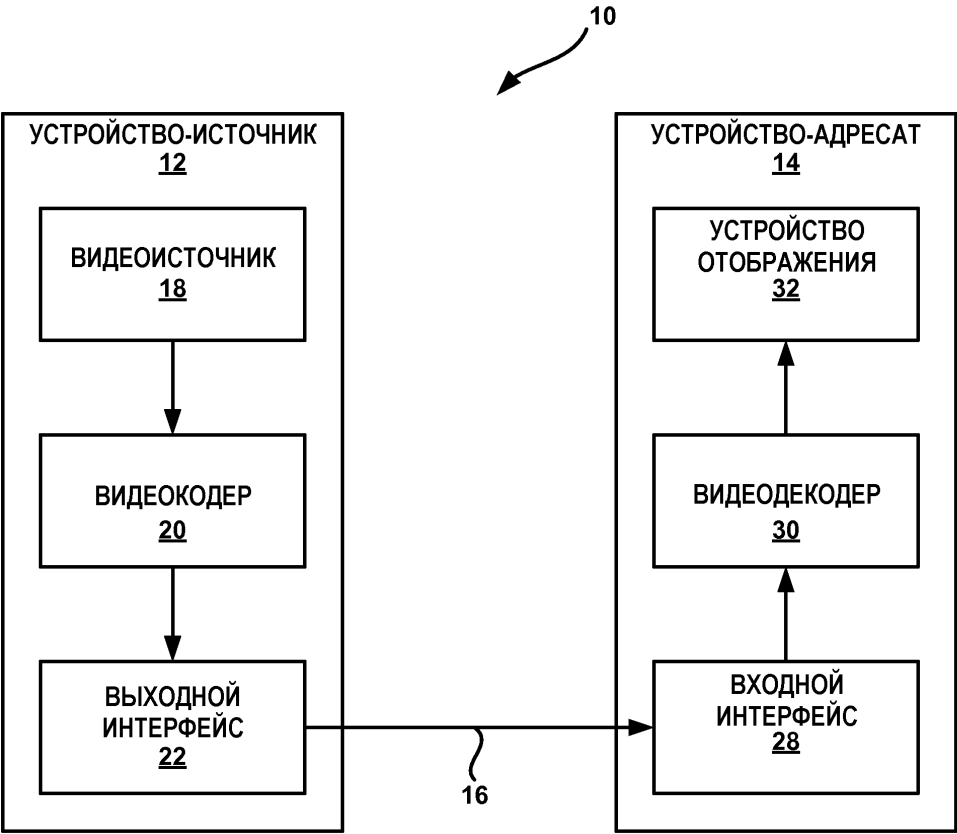
45



1

536516

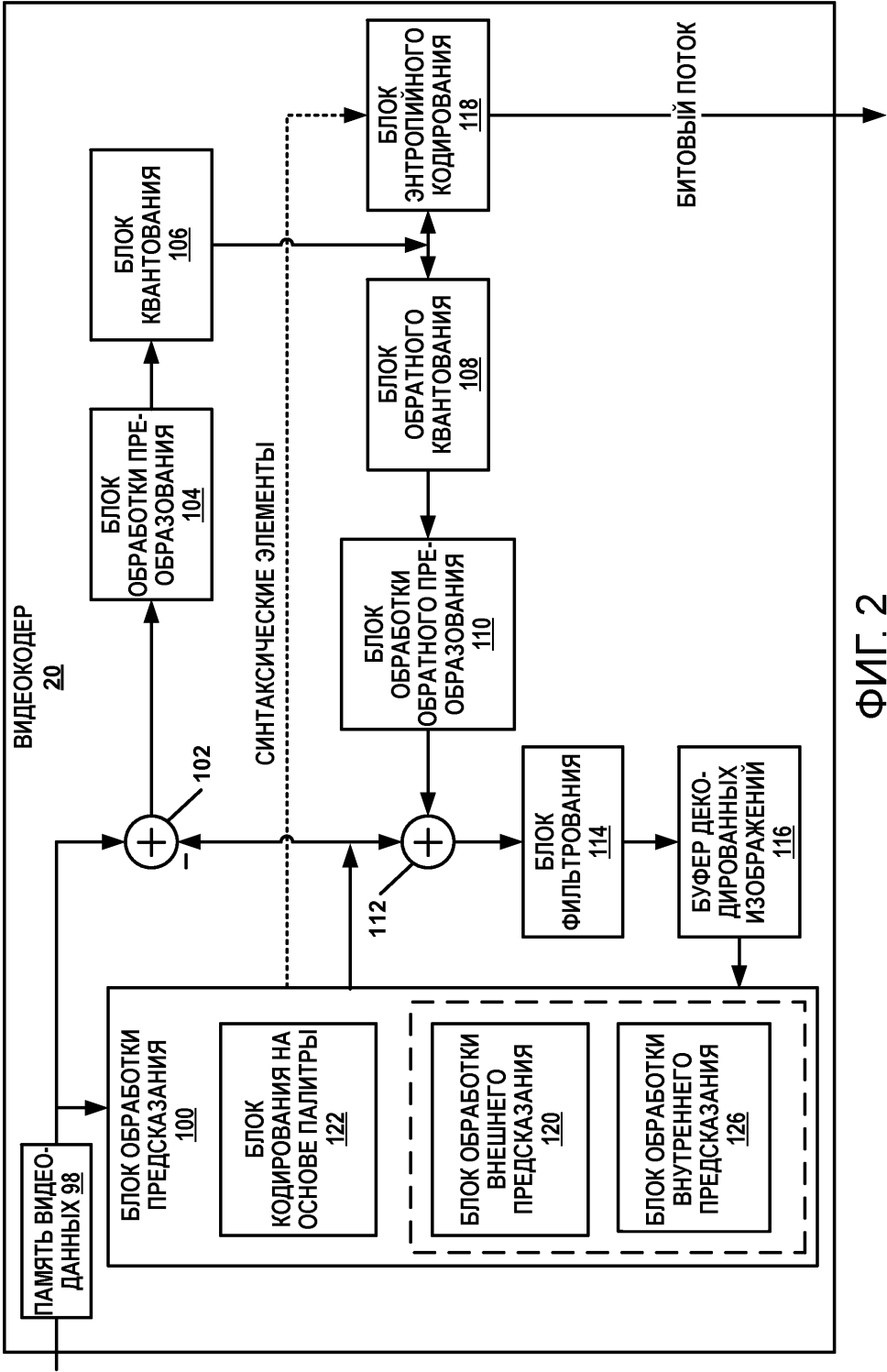
1/6



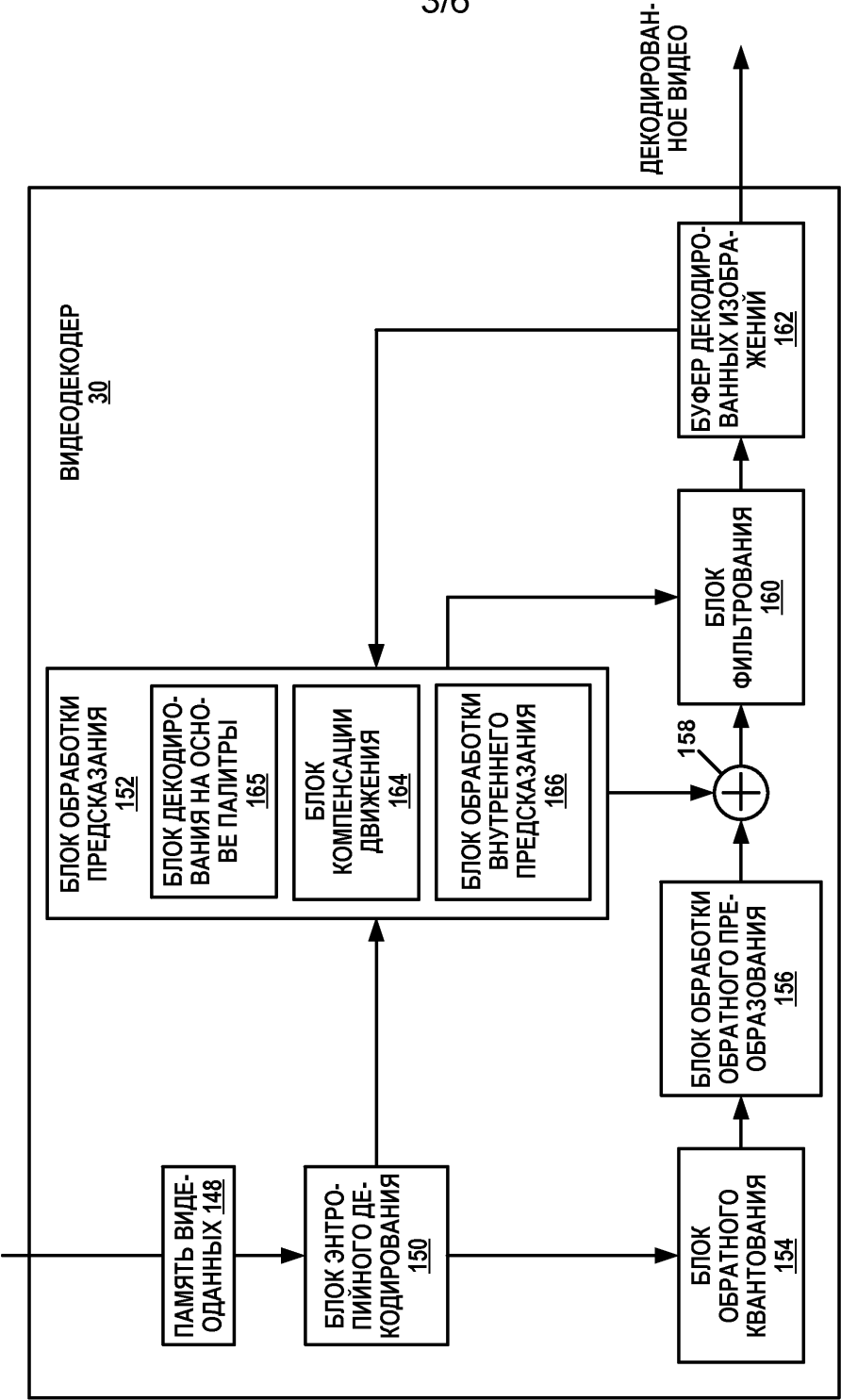
ФИГ. 1

2

2/6

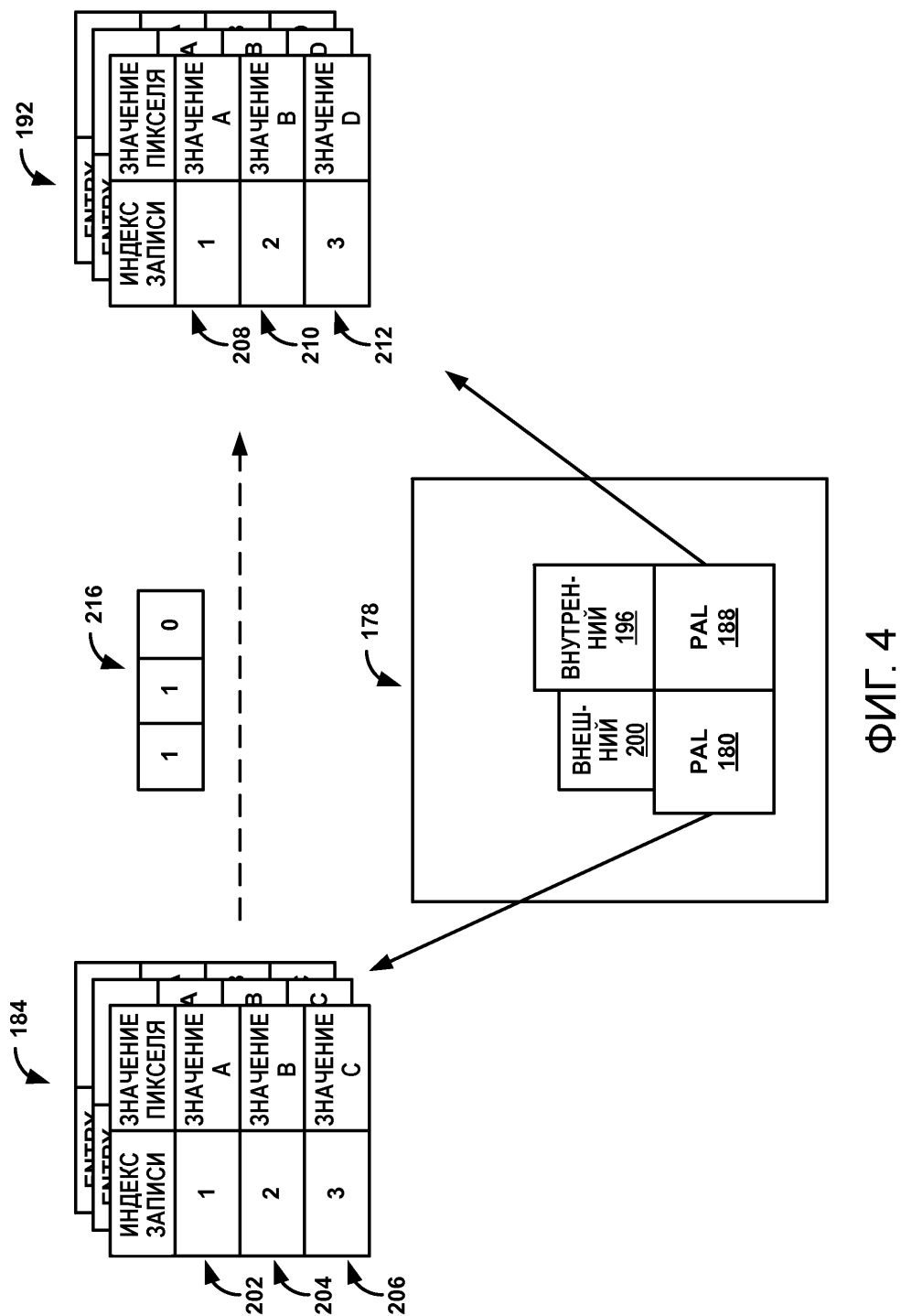


3/6



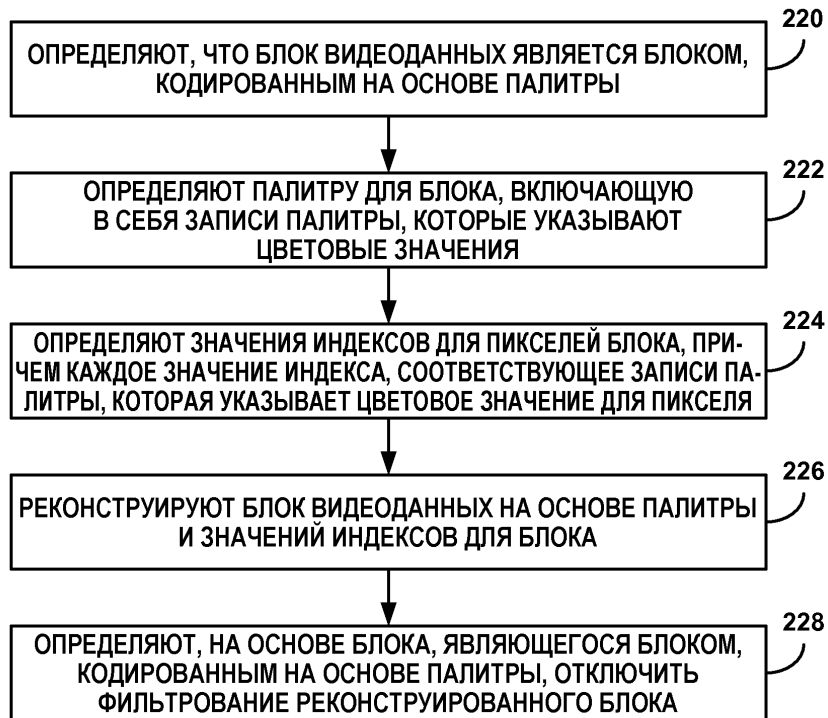
ФИГ. 3

4/6



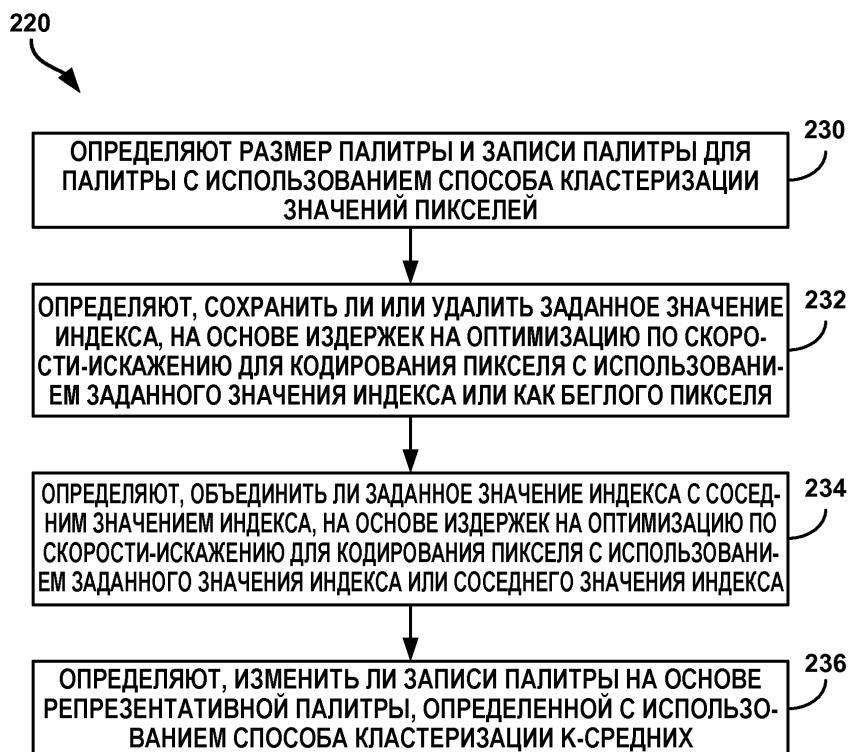
ФИГ. 4

5/6



ФИГ. 5

6/6



ФИГ. 6