



(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2020.05.27

(21) Номер заявки
201791503

(22) Дата подачи заявки
2016.01.29

(51) Int. Cl. **H04N 19/70** (2014.01)
H04N 19/13 (2014.01)
H04N 19/593 (2014.01)
H04N 19/463 (2014.01)
H04N 19/42 (2014.01)

(54) СПОСОБ И УСТРОЙСТВО ДЛЯ КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ ВИДЕОДАНЫХ

(31) **62/110,302; 15/009,477**

(32) **2015.01.30; 2016.01.28**

(33) **US**

(43) **2018.01.31**

(86) **PCT/US2016/015663**

(87) **WO 2016/123488 2016.08.04**

(71)(73) Заявитель и патентовладелец:
КВЭЛКОММ ИНКОРПОРЕЙТЕД
(US)

(72) Изобретатель:
Карчевич Марта, Пу Вэй, Джоши
Раджан Лаксман, Серегин Вадим (US)

(74) Представитель:
Медведев В.Н. (RU)

(56) CHEN J. ET AL.: "Description of screen content coding technology proposal by Qualcomm", 17. JCT-VC MEETING; 27-3-2014 - 4-4-2014; VALENCIA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-Q0031-V3, 28 March 2014 (2014-03-28), XP030115916, Section 2.7.1 with sub-section 2.7.1.1
Rajan Joshi ET AL.: "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 2, Document

JCTVC-S1005", 12 October 2014 (2014-10-12), pages 1-350, XP55263982, 18th Meeting: Sapporo, JP, 30 June - 9 July 2014, Retrieved from the Internet:URL: http://phenix.int-evry.fr/jct/[retrieved on 2016-04-08], cited in the application, Sections 7.3.8.8, 7.4.9.6, 8.4.5.2.8, 9.3.3.13, 9.4.1.2.8; tables 9-43

Vivienne Sze ET AL.: "High Efficiency Video Coding (HEVC): Algorithms and Architectures - Chapter 8: Entropy Coding in HEVC", In: "High Efficiency Video Coding (HEVC): Algorithms and Architectures - Chapter 8: Entropy Coding in HEVC", 1 January 2014 (2014-01-01), Springer International Publishing, XP55263413, ISBN: 978-3-319-06894-7, pages 209-269, abstract, Sections 8.2.3.2, 8.3 with sub-sections 8.3.3, 8.3.3.2, and 8.8.2 with sub-sections 8.8.2.2

KARCZEWICZ M. ET AL.: "Non CE1: Grouping Palette Indices At Front", 20. JCT-VC MEETING; 10-2-2015 - 18-2-2015; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-T0065-V5, 11 February 2015 (2015-02-11), XP030117191, the whole document

JOSHI R. ET AL.: "Screen Content Coding Test Model 4 Encoder Description (SCM 4)", 20. JCT-VC MEETING; 10-2-2015 - 18-2-2015; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-T1014, 1 June 2015 (2015-06-01), XP030117420, Section 3.4 with sub-sections

(57) В изобретении способ декодирования видеоданных может включать в себя прием кодированного в режиме палитры блока видеоданных изображения; способ может включать в себя прием кодированной информации режима палитры для кодированного в режиме палитры блока видеоданных; кодированная информация режима палитры может включать в себя множество экземпляров первого синтаксического элемента и множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента; способ может включать в себя декодирование с использованием обходного режима, множества экземпляров первого синтаксического элемента перед декодированием множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, с использованием контекстного режима; способ может включать в себя декодирование с использованием контекстного режима множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, после декодирования множества экземпляров первого синтаксического элемента с использованием обходного режима.

Настоящая заявка заявляет преимущество приоритета предварительной заявки на патент США № 62/110302, поданной 30 января 2015 г., полное содержимое которой включено в данный документ посредством ссылки.

Область техники

Это раскрытие сущности изобретения относится к кодированию и декодированию контента и, более конкретно, к кодированию и декодированию контента согласно основанному на палитре режиму кодирования.

Уровень техники

Возможности цифрового видео могут быть включены в широкий диапазон устройств, в том числе средства цифрового телевидения, системы цифрового прямого вещания, системы беспроводного вещания, персональные цифровые помощники (personal digital assistant - PDA), переносные или настольные компьютеры, планшетные компьютеры, устройства для чтения электронных книг, цифровые камеры, устройства цифровой записи, устройства воспроизведения цифрового мультимедиа, игровые видеоустройства, консоли видеоигр, сотовые или спутниковые радиотелефоны, так называемые "смартфоны", устройства видео-телеконференцсвязи, устройства потокового видео и т.п. Цифровые видеоустройства реализуют технологии сжатия видео, такие как технологии, описанные в стандартах, определяемых MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, ч. 10, Усовершенствованное Кодирование Видео (Advanced Video Coding - AVC), ITU-T H.265, Высокоэффективное Кодирование Видео (High Efficiency Video Coding - HEVC) и расширения таких стандартов. Видеоустройства могут передавать, принимать, кодировать, декодировать и/или сохранять цифровую видеoinформацию более эффективно посредством реализации таких технологий сжатия видео.

Технологии сжатия видео выполняют пространственное (интра-кадровое) предсказание и/или временное (интер-кадровое) предсказание для уменьшения или устранения избыточности, присущей видеопоследовательностям. Для основанного на блоках кодирования видео, видеослайс (т.е. видеокادر или часть видеокадра) может быть разбит на видеоблоки. Видеоблоки в интра-кодированном (I) слайсе изображения кодируются с использованием пространственного предсказания в отношении опорных отсчетов в соседних блоках в том же самом изображении. Видеоблоки в интер-кодированном (P или B) слайсе изображения могут использовать пространственное предсказание в отношении опорных отсчетов в соседних блоках в том же самом изображении или временное предсказание относительно опорных отсчетов в других опорных изображениях. Изображения могут называться кадрами, и опорные изображения могут называться опорными кадрами.

Пространственное или временное предсказание приводит в результате к предиктивному блоку для блока, подлежащего кодированию. Остаточные данные представляют пиксельные разности между исходным блоком, подлежащим кодированию, и предиктивным блоком. Интер-кодированный блок кодируют согласно вектору движения, который указывает на блок опорных отсчетов, формирующих предиктивный блок, а остаточные данные указывают на различие между кодированным блоком и предиктивным блоком. Интра-кодированный блок кодируют согласно режиму интра-кодирования и остаточным данным. Для дополнительного сжатия остаточные данные могут быть преобразованы из пиксельной области в область преобразования, что приводит к остаточным коэффициентам, которые затем могут быть квантованы. Квантованные коэффициенты, исходно размещаемые в двумерном массиве, могут сканироваться для создания одномерного вектора коэффициентов, и энтропийное кодирование может применяться для достижения еще большего сжатия.

Контент, такой как изображение, может кодироваться и декодироваться с использованием режима палитры. В общем, режим палитры является технологией, включающей в себя использование палитры для представления контента. Контент может кодироваться таким образом, чтобы контент был представлен картой индексов, которая включает в себя значения, соответствующие палитре. Карта индексов может быть декодирована для восстановления контента.

Сущность изобретения

Технологии этого раскрытия сущности изобретения относятся к основанному на палитре кодированию контента. Например, в основанном на палитре кодировании контента, кодер контента (например, кодер контента, такой как видеокодер или видеodeкодер) может формировать "палитру" в виде таблицы цветов для представления видеоданных конкретной области (например, данного блока). Основанное на палитре кодирование контента может быть, например, особенно полезным для кодирования областей видеоданных, имеющих относительно малое число цветов. В отличие от кодирования фактических пиксельных значений (или их остаточных данных) кодер контента может кодировать индексы палитры (например, индексные значения) для одного или нескольких пикселей, которые связывают пиксели с записями в палитре, представляющими цвета пикселей. Технологии, описанные в этом раскрытии сущности изобретения, могут включать в себя технологии для различных комбинаций одного или нескольких из сигнализации основанных на палитре режимов кодирования, передачи палитр, получения палитр, получения значения непередаваемых синтаксических элементов, передачи основанных на палитре карт кодирования и других синтаксических элементов, предсказания записей палитры, кодирования серий индексов палитры, информации о палитре энтропийного кодирования и различных других технологий кодирова-

ного режима в кодированный битовый поток.

В другом примере это раскрытие сущности изобретения описывает устройство для кодирования видеоданных, причем устройство содержит память, выполненную с возможностью хранения видеоданных; и видеокодер, связанный с памятью, причем видеокодер выполнен с возможностью: определения того, что блок видеоданных, хранимый в памяти, подлежит кодированию в режиме палитры; кодирования блока видеоданных с использованием режима палитры в кодированный битовый поток, причем то, что видеокодер выполнен с возможностью кодирования блока видеоданных с использованием режима палитры, содержит то, что видеокодер выполнен с возможностью генерирования информации режима палитры для блока видеоданных, причем информация режима палитры включает в себя множество экземпляров первого синтаксического элемента и множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента; кодирования, с использованием обходного режима, множества экземпляров первого синтаксического элемента в кодированный битовый поток перед кодированием множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, в кодированный битовый поток с использованием контекстного режима; и кодирования, с использованием контекстного режима, множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, в кодированный битовый поток, после кодирования множества экземпляров первого синтаксического элемента с использованием обходного режима в кодированный битовый поток.

Подробности одного или нескольких примеров раскрытия сущности изобретения представлены в сопутствующих чертежах и описании, приведенном ниже. Другие признаки, объекты и преимущества раскрытия сущности изобретения станут очевидными из описания и чертежей, а также из формулы изобретения.

Краткое описание чертежей

Фиг. 1 является блок-схемой, показывающей иллюстративную систему кодирования видео, которая может использовать технологии, описанные в этом раскрытии сущности изобретения.

Фиг. 2 является блок-схемой, показывающей иллюстративный видеокодер, который может реализовать технологии, описанные в этом раскрытии сущности изобретения.

Фиг. 3 является блок-схемой, показывающей иллюстративный видеодекодер, который может реализовать технологии, описанные в этом раскрытии сущности изобретения.

Фиг. 4 является концептуальной схемой, показывающей пример определения записей палитры для основанного на палитре кодирования видео, совместимый с технологиями этого раскрытия сущности изобретения.

Фиг. 5 является концептуальной схемой, показывающей пример определения индексов для палитры для блока пикселей, совместимый с технологиями этого раскрытия сущности изобретения.

Фиг. 6 является концептуальной схемой, показывающей пример определения максимальной длины серии в режиме копирования сверху, предполагая растровый порядок сканирования, совместимый с технологиями этого раскрытия сущности изобретения.

Фиг. 7 является таблицей, показывающей изменения для порядка кодирования синтаксических элементов для режима палитры.

Фиг. 8 является блок-схемой последовательности операций, показывающей иллюстративный процесс для декодирования видеоданных, совместимый с технологиями для основанного на палитре кодирования видео этого раскрытия сущности изобретения.

Фиг. 9 является блок-схемой последовательности операций, показывающей иллюстративный процесс для кодирования видеоданных, совместимый с технологиями для основанного на палитре кодирования видео этого раскрытия сущности изобретения.

Подробное описание

Аспекты этого раскрытия сущности изобретения направлены на технологии для кодирования контента (например, кодирования видео). Конкретно, это раскрытие сущности изобретения описывает технологии для основанного на палитре кодирования данных контента (например, видеоданных) и технологии для контекстно-адаптивного двоичного арифметического кодирования (context-based adaptive binary arithmetic coding - CABAC) информации кодирования палитры. В различных примерах этого раскрытия сущности изобретения технологии этого раскрытия сущности изобретения могут быть направлены на процессы предсказания или кодирования блока в режиме палитры для улучшения эффективности кодирования и/или уменьшения сложности кодека, как описано более подробно ниже. Например, раскрытие сущности изобретения описывает технологии, связанные с группированием индексов палитры (таким как усовершенствованное группирование индексов палитры).

В процессе CABAC, например, описанном в публикации D. Marpe, H. Schwarz и T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard", в IEEE Trans. Cir. & Sys. Video Tech., т. 13, № 7, июль 2003, существуют два режима: (1) обходной режим и (2) контекстный режим. В обходном режиме не существует процесса обновления контекста. Таким образом, обходной режим может достичь более высокой производительности, чем контекстный режим, посредством использования параллелизма уровня аппаратного обеспечения или ISA. Это преимущество обходного

режима становится большим, поскольку число обходных бинов, которые могут быть обработаны вместе, увеличивается.

В текущей конструкции кодирования режима палитры, описанной в публикации R. Joshi и J. Xu, "High efficient video coding (HEVC) screen content coding: Draft 2", JCTVC-S1005, при кодировании контента экрана синтаксические элементы `palette_index_idc` (идентификатор_индекса_палитры) и `palette_escape_val` (переходное_значение_палитры) кодируют в обходном режиме CABAC и чередуют с другими синтаксическими элементами, такими как `palette_run_msb_id_plus1`, которые кодированы в контекстном режиме CABAC. Это раскрытие сущности изобретения описывает технологии группирования вместе кодированных в обходном режиме синтаксических элементов. При использовании здесь, термины "кодированный в обходном режиме" и "кодированный в контекстном режиме" являются взаимозаменяемыми с терминами "обходным путем кодированный" и "контекстно кодированный".

При использовании здесь, примеры термина "контент" могут быть заменены на термин "видео", а примеры термина "видео" могут быть заменены на термин "контент". Это является истинным независимо от того, используются ли термины "контент" или "видео" в форме прилагательного, существительного или другой части речи. Например, ссылка на "кодер контента" также включает в себя ссылку на "видеокодер", а ссылка на "видеокодер" также включает в себя ссылку на "кодер контента". Подобным образом, ссылка на "контент" также включает в себя ссылку на "видео", а ссылка на "видео" также включает в себя ссылку на "контент".

При использовании здесь, "контент" относится к любому типу контента. Например, "контент" может относиться к видео, контенту экрана, изображению, любому графическому контенту, любому отображаемому контенту, или к любым данным, соответствующим этому (например, видеоданным, данным контента экрана, данным изображения, данным графического контента, данным отображаемого контента, и т.п.).

При использовании здесь, термин "видео" может относиться к контенту экрана, перемещаемому контенту, множеству изображений, которые могут быть представлены в последовательности, или к любым данным, соответствующим этому (например, данным контента экрана, данным перемещаемого контента, видеоданным, данным изображений и т.п.).

При использовании здесь, термин "изображение" может относиться к единственному изображению, одному или нескольким изображениям, одному или нескольким изображениям среди множества изображений, соответствующих видео, одному или нескольким изображениям среди множества изображений, не соответствующих видео, множеству изображений, соответствующих видео (например, всем изображениям, соответствующим видео, или менее чем всем изображениям, соответствующим видео), подчасти единственного изображения, множеству подчастей единственного изображения, множеству подчастей, соответствующих множеству изображений, одному или нескольким графическим примитивам, данным изображений, графическим данным и т.п.

В традиционном кодировании видео предполагается, что изображения имеют непрерывный спектр тонов и являются пространственно сглаженными. На основе этих предположений были разработаны различные инструменты, такие как основанные на блоках преобразования, фильтрация и другие инструменты кодирования, и такие инструменты показали хорошую производительность для видео естественного контента. Однако в применениях, таких как удаленный рабочий стол, совместная работа и беспроводное отображение, генерируемый компьютером контент экрана может быть основным контентом, подлежащим сжатию. Этот тип контента экрана, как правило, имеет дискретный тон, резкие линии и высококонтрастные границы объектов. Предположение непрерывного спектра тонов и сглаженности может быть уже неприменимым, и, таким образом, традиционные технологии кодирования видео могут быть неэффективными при сжатии контента (например, контента экрана).

В одном примере основанного на палитре кодирования видео видеокодер может кодировать блок видеоданных посредством определения палитры для блока (например, кодирования палитры явным образом, предсказания палитры, или их комбинации), размещения записи в палитре для представления значения (значений) одного или нескольких пикселей, и кодирования как палитры, так и блока с использованием индексных значений, которые указывают на запись в палитре, используемую для представления пиксельных значений блока. В некоторых примерах видеокодер может сигнализировать палитру и/или индексные значения в кодированный битовый поток. В свою очередь, видеодекодер может получить, из кодированного битового потока, палитру для блока, а также индексные значения для отдельных пикселей блока. Видеодекодер может связать индексные значения пикселей с записями палитры для восстановления различных пиксельных значений блока.

Например, можно предположить, что конкретная область видеоданных имеет относительно малое число цветов. Видеокодер (например, видеокодер или видеодекодер) может кодировать (например, кодировать или декодировать) так называемую "палитру" для представления видеоданных конкретной области. Палитра может быть выражена в виде индекса (например, таблицы) цветов или пиксельных значений, представляющих видеоданные конкретной области (например, данного блока). Видеокодер может кодировать индекс, который связывает одно или несколько пиксельных значений с соответствующим значением в палитре. Каждый пиксел может быть связан с записью в палитре, которая представляет цвет пик-

села. Например, палитра может включать в себя наиболее преобладающие пиксельные значения в данном блоке. В некоторых случаях, наиболее преобладающие пиксельные значения могут включать в себя одно или несколько пиксельных значений, которые встречаются наиболее часто в пределах блока. Дополнительно, в некоторых случаях, видеокодер может применить пороговое значение для определения того, должно ли быть пиксельное значение включено в качестве одного из наиболее преобладающих пиксельных значений в блоке. Согласно различным аспектам основанного на палитре кодирования видеокодер может кодировать индексные значения, указывающие на одно или несколько пиксельных значений текущего блока, вместо кодирования фактических пиксельных значений или их остаточных данных для текущего блока видеоданных. В контексте основанного на палитре кодирования, индексные значения указывают на соответствующие записи в палитре, которые используются для представления отдельных пиксельных значений текущего блока. Описание, приведенное выше, предназначено для обеспечения общего описания основанного на палитре кодирования видео.

Основанное на палитре кодирование может быть особенно подходящим для кодирования генерируемого на экране контента или другого контента, где один или несколько более традиционных инструментов кодирования являются неэффективными. Технологии основанного на палитре кодирования видеоданных могут быть использованы с одной или несколькими другими технологиями кодирования, такими как технологии для интер- или интра-предиктивного кодирования. Например, как описано более подробно ниже, кодер или декодер, или объединенный кодер-декодер (кодек), может быть выполнен с возможностью выполнения интер- и интра-предиктивного кодирования, а также основанного на палитре кодирования.

В некоторых примерах технологии основанного на палитре кодирования могут быть выполнены с возможностью использования с одним или несколькими стандартами кодирования видео. Например, Высокоэффективное Кодирование Видео (HEVC) является новым стандартом кодирования видео, разработанным Объединенной группой для совместной работы в области кодирования видео (JCT-VC) Экспертной группы в области кодирования видео (VCEG) ITU-T и Экспертной группы по киноизображению (MPEG) ISO/IEC. Законченный документ стандарта HEVC опубликован в виде "ITU-T H.265, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video High efficiency video coding", Сектор стандартизации электросвязи Международного союза электросвязи (ITU), апрель 2013.

Для обеспечения более эффективного кодирования генерируемого на экране контента JCT-VC разрабатывает расширение для стандарта HEVC, называемое стандартом Кодирования контента экрана (Screen Content Coding - SCC) HEVC. Последний рабочий проект стандарта SCC HEVC, называемый "Рабочим проектом 2 SCC HEVC" или "WD2", описан в документе JCTVC-S1005, R. Joshi и J. Xu, "HEVC screen content coding draft text 2", Объединенной группы для совместной работы в области кодирования видео (JCT-VC) ITU-T SG 16 IP 3 и ISO/IEC JTC 1/SC 29/WG 11, 19th Meeting: Страсбург, FR, 17-24 октября 2014.

В отношении инфраструктуры HEVC, в качестве одного примера, технологии основанного на палитре кодирования могут быть выполнены с возможностью использования в виде режима единицы кодирования (coding unit - CU). В других примерах технологии основанного на палитре кодирования могут быть выполнены с возможностью использования в виде режима единицы предсказания (prediction unit - PU) в инфраструктуре HEVC. Соответственно все раскрытые ниже процессы, описанные в контексте режима CU, могут дополнительно или альтернативно применяться к PU. Однако эти основанные на HEVC примеры не должны рассматриваться как запрет или ограничение технологий основанного на палитре кодирования, описанных здесь, поскольку такие технологии могут быть применены для независимой работы или в виде части других существующих или еще подлежащих разработке систем/стандартов. В этих случаях единицами для кодирования палитры могут быть квадратные блоки, прямоугольные блоки или даже области непрямоугольной формы.

В некоторых примерах палитра может быть получена для одной или нескольких CU, PU, или любой области данных (например, любого блока данных). Например, палитра может содержать (или может состоять из) наиболее преобладающие пиксельные значения в текущей CU, причем CU является областью данных для этого конкретного примера. Размер и элементы палитры сначала передают от видеокодера к видеодекодеру. Размер и/или элементы палитры могут быть прямо кодированными или предиктивно кодированными с использованием размера и/или элементов палитры в соседних CU (например, верхней и/или левой кодированной CU). После этого, пиксельные значения в CU кодируют на основе палитры согласно некоторому порядку сканирования. Для каждого пиксельного местоположения в CU может быть сначала передан флаг (например, `palette_flag` (флаг_палитры) или `escape_flag` (флаг_перехода)) для указания на то, включено ли пиксельное значение в палитру. Для тех пиксельных значений, которые отображаются в записи в палитре, индекс палитры, связанный с этой записью, сигнализируют для данного пиксельного местоположения в CU. Вместо отправки флага (например, `palette_flag` или `escape_flag`), для тех пиксельных значений, которые не существуют в палитре, пикселу может быть присвоен специальный индекс, и фактическое пиксельное значение (возможно, в квантованной форме) может быть передано для данного пиксельного местоположения в CU. Эти пиксели называются "переходными пиксе-

лами". Переходной пиксел может быть кодирован с использованием любого существующего способа энтропийного кодирования, такого как кодирование с фиксированной длиной, унарное кодирование и т.д. В некоторых примерах одна или несколько технологий, описанных здесь, могут использовать флаг, такой как `palette_flag` или `escape_flag`. В других примерах одна или несколько технологий, описанных здесь, могут не использовать флаг, такой как `palette_flag` или `escape_flag`.

Отсчеты в блоке видеоданных могут быть обработаны (например, отсканированы) с использованием горизонтального растрового порядка сканирования или другого порядка сканирования. Например, видеокодер может преобразовать двумерный блок индексов палитры в одномерный массив посредством сканирования индексов палитры с использованием горизонтального растрового порядка сканирования. Подобным образом, видеодекодер может восстановить блок индексов палитры с использованием горизонтального растрового порядка сканирования. Таким образом, это раскрытие сущности изобретения может относиться к предшествующему отсчету как к отсчету, который предшествует отсчету, кодируемому в данный момент, в блоке в порядке сканирования. Следует понимать, что сканирования, отличные от горизонтального растрового сканирования, такие как вертикальный растровый порядок сканирования, могут также быть применимыми. Вышеупомянутый пример, а также другие примеры, приведенные в этом раскрытии сущности изобретения, предназначены для обеспечения общего описания основанного на палитре кодирования видео.

Фиг. 1 является блок-схемой, показывающей иллюстративную систему 10 кодирования видео, которая может использовать технологии этого раскрытия сущности изобретения. При использовании здесь, термин "видеокодер" относится, в общем, как к видеокодерам, так и к видеодекодерам. В этом раскрытии сущности изобретения, термины "кодирование видео" или "кодирование" могут относиться, в общем, к кодированию видео или к декодированию видео. Видеокодер 20 и видеодекодер 30 системы 10 кодирования видео представляют собой примеры устройств, которые могут быть выполнены с возможностью реализации технологий для основанного на палитре кодирования видео и энтропийного кодирования (например, CABAC) согласно различным примерам, описанным в этом раскрытии сущности изобретения. Например, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью выборочного кодирования различных блоков видеоданных, таких как CU или PU в кодировании HEVC, с использованием либо основанного на палитре кодирования, либо основанного не на палитре кодирования. Режимы основанного не на палитре кодирования могут относиться к различным режимам интер-предиктивного временного кодирования или режимам интра-предиктивного пространственного кодирования, таким как различные режимы кодирования, заданные стандартом HEVC.

Как показано на фиг. 1, система 10 кодирования видео включает в себя исходное устройство 12 и целевое устройство 14. Исходное устройство 12 генерирует кодированные видеоданные. Соответственно исходное устройство 12 может называться устройством кодирования видео или аппаратом кодирования видео. Целевое устройство 14 может декодировать кодированные видеоданные, генерируемые исходным устройством 12.

Соответственно целевое устройство 14 может называться устройством декодирования видео или аппаратом декодирования видео. Исходное устройство 12 и целевое устройство 14 могут быть примерами устройств кодирования видео или аппаратов кодирования видео.

Исходное устройство 12 и целевое устройство 14 могут содержать широкий диапазон устройств, в том числе настольные компьютеры, мобильные вычислительные устройства, компьютеры-ноутбуки (например, переносные компьютеры), планшетные компьютеры, телевизионные приставки, мобильные телефоны, такие как так называемые смартфоны, телевизионные устройства, камеры, устройства отображения, устройства воспроизведения цифрового мультимедиа, консоли видеоигр, автомобильные компьютеры и т.п.

Целевое устройство 14 может принимать кодированные видеоданные от исходного устройства 12 через канал 16. Канал 16 может содержать один или несколько носителей или устройств, способных перемещать кодированные видеоданные от исходного устройства 12 к целевому устройству 14. В одном примере канал 16 может содержать одну или несколько сред передачи данных, которые позволяют исходному устройству 12 передавать кодированные видеоданные прямо к целевому устройству 14 в реальном времени. В этом примере исходное устройство 12 может модулировать кодированные видеоданные согласно стандарту связи, такому как протокол беспроводной связи, и может передавать модулированные видеоданные к целевому устройству 14. Упомянутые одна или несколько сред передачи данных могут включать в себя беспроводные и/или проводные среды передачи данных, такие как спектр радиочастот (`radio frequency - RF`) или одна или несколько физических линий передачи данных. Упомянутые одна или несколько сред передачи данных могут образовывать часть основанной на пакетах сети, такой как локальная сеть, широкомасштабная сеть, или глобальная сеть (например, Интернет). Упомянутые одна или несколько сред передачи данных могут включать в себя маршрутизаторы, коммутаторы, базовые станции, или другое оборудование, которое обеспечивает передачу данных от исходного устройства 12 к целевому устройству 14.

В другом примере канал 16 может включать в себя носитель данных, который запоминает кодированные видеоданные, генерируемые исходным устройством 12. В этом примере целевое устройство 14

может осуществлять доступ к носителю данных посредством, например, осуществления доступа к диску или доступа к карте. Носитель данных может включать в себя множество локально-доступных устройств хранения данных, таких как диски Blu-ray, DVD, CD-ROM, флэш-память или другие устройства хранения данных, подходящие для хранения кодированных видеоданных.

В дополнительном примере канал 16 может включать в себя файловый сервер или другое промежуточное устройство хранения данных, которое запоминает кодированные видеоданные, генерируемые исходным устройством 12. В этом примере целевое устройство 14 может осуществлять доступ к кодированным видеоданным, хранимым в файловом сервере или другом промежуточном устройстве хранения данных посредством потокового режима или загрузки. Файловый сервер может быть сервером такого типа, который способен хранить кодированные видеоданные и передавать кодированные видеоданные к целевому устройству 14. Иллюстративные файловые серверы включают в себя веб-серверы (например, для веб-сайта), серверы протокола передачи файлов (file transfer protocol - FTP), подключенные к сети запоминающие устройства (network attached storage - NAS) и локальные дисковые накопители.

Целевое устройство 14 может осуществлять доступ к кодированным видеоданным посредством стандартного информационного соединения, такого как Интернет-соединение. Иллюстративные 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 являются подходящими для обеспечения доступа к кодированным видеоданным, хранимым на файловом сервере. Передача кодированных видеоданных от файлового сервера может быть потоковой передачей, передачей с помощью загрузки или их комбинацией.

Исходное устройство 12 и целевое устройство 14 могут быть выполнены с возможностью выполнения основанного на палитре кодирования и энтропийного кодирования (например, CABAC), совместимых с этим раскрытием сущности изобретения. Технологии этого раскрытия сущности изобретения для основанного на палитре кодирования или CABAC, однако, не ограничены беспроводными применениями или системами. Эти технологии могут быть применены к кодированию видео для поддержки множества мультимедийных применений, таких как эфирное телевизионное вещание, кабельные телевизионные передачи, спутниковые телевизионные передачи, передачи потокового видео, например, через Интернет, кодирование видеоданных для хранения на устройстве хранения данных, декодирование видеоданных на устройстве хранения данных, или другие применения. В некоторых примерах система 10 кодирования видео может быть выполнена с возможностью поддержки односторонней или двусторонней передачи видео для поддержки применений, таких как обеспечение потокового видео, воспроизведение видео, трансляция видео и/или видеотелефония.

Система 10 кодирования видео, показанная на фиг. 1, является просто примером, и технологии этого раскрытия сущности изобретения могут применяться к системам кодирования видео (например, кодированию видео или декодированию видео), которые необязательно включают в себя какую-либо передачу данных между устройствами кодирования и декодирования. В других примерах данные извлекаются из локальной памяти, передаются в потоковом режиме по сети и т.п. Устройство кодирования видео может кодировать и сохранять данные в памяти, и/или устройство декодирования видео может извлекать и декодировать данные из памяти. Во многих примерах кодирование и декодирование выполняются устройствами, которые не передают данные друг другу, а просто кодируют данные в памяти и/или извлекают и декодируют данные из памяти.

В примере фиг. 1 исходное устройство 12 включает в себя источник 18 видео, видеокодер 20 и интерфейс 22 вывода. В некоторых примерах интерфейс 22 вывода может включать в себя модулятор/демодулятор (модем) и/или передатчик. Источник 18 видео 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 видео.

Видеокодер 20 может кодировать видеоданные из источника 18 видео. В некоторых примерах исходное устройство 12 прямо передает кодированные видеоданные к целевому устройству 14 через интерфейс 22 вывода. В других примерах кодированные видеоданные могут быть также сохранены на носителе данных или файловом сервере для более позднего доступа посредством целевого устройства 14 для декодирования и/или воспроизведения.

В примере фиг. 1 целевое устройство 14 включает в себя интерфейс 28 ввода, видеodeкодер 30 и устройство 32 отображения. В некоторых примерах интерфейс 28 ввода включает в себя приемник и/или модем. Интерфейс 28 ввода может принимать кодированные видеоданные через канал 16. Устройство 32 отображения может быть встроено в целевое устройство 14 или может быть внешним по отношению к нему. В общем, устройство 32 отображения отображает декодированные видеоданные. Устройство 32 отображения может содержать множество устройств отображения, таких как жидкокристаллический дисплей (liquid crystal display - LCD), плазменный дисплей, дисплей на органических светодиодах (organic light emitting diode - OLED) или другой тип устройства отображения.

Это раскрытие сущности изобретения может, в общем, относиться к видеокодеру 20, "сигнализирующему" или "передающему" некоторую информацию к другому устройству, такому как видеodeкодер

30. Термин "сигнализирующий" или "передающий" может, в общем, относиться к передаче синтаксических элементов и/или других данных, используемых для декодирования сжатых видеоданных. Такая передача может происходить в реальном времени или близко к реальному масштабу времени. Альтернативно, такая передача может происходить в течение интервала времени, например, может происходить при сохранении на машиночитаемом носителе данных в кодированном битовом потоке во время кодирования синтаксических элементов, которые затем могут быть извлечены устройством декодирования в любой момент времени после сохранения на этом носителе. Таким образом, в то время как видеodeкодер 30 может называться "принимающим" некоторую информацию, прием информации не обязательно происходит в реальном времени или близко к реальному масштабу времени и информация может быть извлечена из носителя в некоторый момент времени после сохранения.

Каждый из видеокодера 20 и видеodeкодера 30 может быть реализован в виде любой из множества подходящих схем, таких как один или несколько микропроцессоров, цифровых сигнальных процессоров (digital signal processor - DSP), специализированных интегральных схем (application-specific integrated circuit ASIC), матриц программируемых логических вентилей (field-programmable gate array - FPGA), дискретных логических схем, аппаратных средств или любых их комбинаций. Если технологии реализованы частично в программном средстве, то устройство может хранить команды для программного средства в подходящем, долговременном машиночитаемом носителе данных и может исполнять команды в аппаратном средстве с использованием одного или нескольких процессоров для реализации технологий этого раскрытия сущности изобретения. Можно считать, что любое из вышеупомянутого (в том числе аппаратные средства, программные средства, комбинация аппаратных и программных средств и т.д.) является одним или несколькими процессорами. Каждый из видеокодера 20 и видеodeкодера 30 может быть включен в один или несколько кодеров и декодеров, любой из которых может быть встроен в виде части объединенного кодера/декодера (кодека) в соответствующее устройство.

В некоторых примерах видеокодер 20 и видеodeкодер 30 функционируют согласно стандарту сжатия видео, такому как стандарт HEVC, упомянутый выше, и описаны в стандарте HEVC. Дополнительно к базовому стандарту HEVC предпринимаются постоянные попытки по созданию масштабируемых расширений HEVC для кодирования видео, многовидового кодирования видео и трехмерного кодирования. Дополнительно, основанные на палитре режимы кодирования, например, описанные в этом раскрытии сущности изобретения, могут быть обеспечены для расширения стандарта HEVC. В некоторых примерах технологии, описанные в этом раскрытии сущности изобретения для основанного на палитре кодирования, могут быть применены к кодерам и декодерам, выполненным с возможностью функционирования согласно другим стандартам кодирования видео. Соответственно применение основанного на палитре режима кодирования для кодирования единиц кодирования (CU) или единиц предсказания (PU) в HEVC-кодеке описано в качестве примера.

В HEVC и других стандартах кодирования видео видеопоследовательность обычно включает в себя последовательность изображений. Изображения могут также называться "кадрами". Изображение может включать в себя три массива отсчетов, обозначаемых S_L , S_{Cb} и S_{Cr} . S_L является двумерным массивом (т.е. блоком) отсчетов яркости. S_{Cb} является двумерным массивом отсчетов Cb -цветности. S_{Cr} является двумерным массивом отсчетов Cr -цветности. Отсчеты цветности могут также называться здесь отсчетами "хроматичности". В других примерах изображение может быть монохромным и может включать в себя только массив отсчетов яркости.

Для генерирования кодированного представления изображения видеокодер 20 может генерировать набор древовидных единиц кодирования (coding tree unit - CTU). Каждая из CTU может быть древовидным блоком кодирования отсчетов яркости, двумя соответствующими древовидным блоками кодирования отсчетов цветности и синтаксическими структурами, используемыми для кодирования отсчетов древовидных блоков кодирования. Древовидный блок кодирования может быть $N \times N$ -блоком отсчетов. CTU может также называться "древовидным блоком" или "наибольшей единицей кодирования" (largest coding unit - LCU). CTU HEVC могут быть, в широком смысле, аналогичными макроблокам других стандартов, таких как H.264/AVC. Однако CTU не обязательно ограничена конкретным размером и может включать в себя одну или несколько единиц кодирования (CU). Слайс может включать в себя целое число CTU, упорядоченных последовательно в растровом сканировании. Кодированный слайс может содержать заголовки слайса и данные слайса. Заголовок слайса может быть синтаксической структурой, которая включает в себя синтаксические элементы, которые обеспечивают информацию о слайсе. Данные слайса могут включать в себя кодированные CTU слайса.

Это раскрытие сущности изобретения может использовать термин "единица видео" или "видео-блок" или "блок" для именования одного или нескольких блоков отсчетов и синтаксических структур, используемых для кодирования отсчетов одного или нескольких блоков отсчетов. Иллюстративные типы единиц видео или блоков могут включать в себя CTU, CU, PU, единицы преобразования (transform unit - TU), макроблоки, части макроблоков и т.д. В некоторых контекстах описание PU может быть взаимозаменяемым с описанием макроблоков или частей макроблоков.

Для генерирования кодированной CTU видеокодер 20 может рекурсивно выполнять разбиение посредством дерева квадрантов на древовидные блоки кодирования CTU для разделения древовидных бло-

ков кодирования на блоки кодирования, отсюда название "древовидные блоки кодирования". Блок кодирования является $N \times N$ -блоком отсчетов. CU может быть блоком кодирования отсчетов яркости и двумя соответствующими блоками кодирования отсчетов цветности изображения, которое имеет массив отсчетов яркости, массив Cb-отсчетов и массив Cr-отсчетов, и синтаксические структуры, используемые для кодирования отсчетов блоков кодирования. Видеокодер 20 может разбить блок кодирования CU на один или несколько блоков предсказания. Блок предсказания может быть прямоугольным (т.е. квадратным или неквадратным) блоком отсчетов, в отношении которых применяется одно и то же предсказание. Единица предсказания (PU) CU может быть блоком предсказания отсчетов яркости, двумя соответствующими блоками предсказания отсчетов цветности изображения и синтаксическими структурами, используемыми для предсказания отсчетов блока предсказания. Видеокодер 20 может генерировать предиктивные блоки яркости, Cb и Cr для блоков предсказания яркости, Cb и Cr каждой PU CU.

Видеокодер 20 может использовать интра-предсказание или интер-предсказание для генерирования предиктивных блоков для PU. Если видеокодер 20 использует интра-предсказание для генерирования предиктивных блоков PU, то видеокодер 20 может генерировать предиктивные блоки PU на основе декодированных отсчетов изображения, связанного с PU.

Если видеокодер 20 использует интер-предсказание для генерирования предиктивных блоков PU, то видеокодер 20 может генерировать предиктивные блоки PU на основе декодированных отсчетов одного или нескольких изображений, отличных от изображения, связанного с PU. Видеокодер 20 может использовать одинарное предсказание или двойное предсказание для генерирования предиктивных блоков PU. Когда видеокодер 20 использует одинарное предсказание для генерирования предиктивных блоков PU, PU может иметь единственный вектор движения (motion vector - MV). Когда видеокодер 20 использует двойное предсказание для генерирования предиктивных блоков PU, PU может иметь два MV.

После генерирования видеокодером 20 предиктивных блоков (например, предиктивных блоков яркости, Cb и Cr) для одной или нескольких PU CU видеокодер 20 может сгенерировать остаточные блоки для CU. Каждый отсчет в остаточном блоке CU может указывать на различие между отсчетом в предиктивном блоке PU CU и соответствующим отсчетом в блоке кодирования CU. Например, видеокодер 20 может сгенерировать остаточный блок яркости для CU. Каждый отсчет в остаточном блоке яркости CU указывает на различие между отсчетом яркости в одном из предиктивных блоков яркости CU и соответствующим отсчетом в исходном блоке кодирования яркости CU. Дополнительно, видеокодер 20 может сгенерировать остаточный блок Cb для CU. Каждый отсчет в остаточном блоке Cb CU может указывать на различие между отсчетом Cb в одном из предиктивных блоков Cb CU и соответствующим отсчетом в исходном блоке кодирования Cb CU. Видеокодер 20 может также сгенерировать остаточный блок Cr для CU. Каждый отсчет в остаточном блоке Cr CU может указывать на различие между отсчетом Cr в одном из предиктивных блоков Cr CU и соответствующим отсчетом в исходном блоке кодирования Cr CU.

Кроме того, видеокодер 20 может использовать разбиение посредством дерева квадрантов для декомпозирования остаточных блоков (например, остаточных блоков яркости, Cb и Cr) CU на один или несколько блоков преобразования (например, блоков преобразования яркости, Cb и Cr). Блок преобразования может быть прямоугольным блоком отсчетов, к которому применяется одно и то же преобразование. Единица преобразования (transform unit - TU) CU может быть блоком преобразования отсчетов яркости, двумя соответствующими блоками преобразования отсчетов цветности и синтаксическими структурами, используемыми для преобразования отсчетов блока преобразования. Таким образом, каждая TU CU может быть связана с блоком преобразования яркости, блоком преобразования Cb и блоком преобразования Cr. Блок преобразования яркости, связанный с TU, может быть подблоком остаточного блока яркости CU. Блок преобразования Cb может быть подблоком остаточного блока Cb CU. Блок преобразования Cr может быть подблоком остаточного блока Cr CU.

Видеокодер 20 может применять одно или несколько преобразований к блоку преобразования для генерирования блока коэффициентов для TU. Блок коэффициентов может быть двумерным массивом коэффициентов преобразования. Коэффициент преобразования может быть скалярной величиной. Например, видеокодер 20 может применять одно или несколько преобразований к блоку преобразования яркости TU для генерирования блока коэффициентов яркости для TU. Видеокодер 20 может применять одно или несколько преобразований к блоку преобразования Cb TU для генерирования блока коэффициентов Cb для TU. Видеокодер 20 может применять одно или несколько преобразований к блоку преобразования Cr TU для генерирования блока коэффициентов Cr для TU.

После генерирования блока коэффициентов (например, блока коэффициентов яркости, блока коэффициентов Cb или блока коэффициентов Cr) видеокодер 20 может квантовать блок коэффициентов. Квантование, в общем, относится к процессу, в котором коэффициенты преобразования квантуются для возможного уменьшения количества данных, используемых для представления коэффициентов преобразования, что обеспечивает дополнительное сжатие. После квантования видеокодером 20 блока коэффициентов видеокодер 20 может энтропийно кодировать синтаксические элементы, указывающие на квантованные коэффициенты преобразования. Например, видеокодер 20 может выполнить контекстно-адаптивное двоичное арифметическое кодирование (CABAC) в отношении синтаксических элементов, указывающих на квантованные коэффициенты преобразования.

В отношении САВАС в качестве примера видеокодер 20 и видеodeкодер 30 могут выбрать вероятностную модель (также называемую контекстной моделью) для кодирования символов, связанных блоком видеоданных, на основе контекста. Например, контекстная модель (Ctx) может быть индексом или сдвигом, который применяют для выбора одного из множества разных контекстов, каждый из которых может соответствовать конкретной вероятностной модели. Соответственно другую вероятностную модель обычно определяют для каждого контекста. После кодирования или декодирования бина, вероятностную модель дополнительно обновляют на основе значения бина, чтобы она отражала самые последние вероятностные оценки для бина. Например, вероятностная модель может быть сохранена в виде состояния в конечном автомате. Каждое конкретное состояние может соответствовать конкретному значению вероятности. Следующее состояние, которое соответствует обновлению вероятностной модели, может зависеть от значения текущего бина (например, бина, кодируемого в данный момент). Соответственно на выбор вероятностной модели могут влиять значения ранее кодированных бинов, поскольку эти значения указывают, по меньшей мере частично, на вероятность того, что бин имеет данное значение. Процесс кодирования контекста, описанный выше, может, в общем, называться контекстно-адаптивным режимом кодирования.

Таким образом, видеокодер 20 может кодировать целевой символ с использованием вероятностной модели. Подобным образом, видеodeкодер 30 может синтаксически разобрать целевой символ с использованием вероятностной модели. В некоторых примерах видеокодер 20 может кодировать синтаксические элементы с использованием комбинации контекстно-адаптивного и не-контекстно-адаптивного кодирования. Например, видеокодер 20 может контекстно кодировать бины посредством выбора вероятностной модели или "контекстной модели", которая функционирует в отношении контекста для кодирования некоторых бинов. Напротив, для других бинов видеокодер 20 может кодировать в обходном режиме бины посредством обхода или исключения обычного процесса арифметического кодирования при кодировании бинов. В таких примерах видеокодер 20 может использовать фиксированную вероятностную модель для кодирования бинов в обходном режиме. А именно, кодированные в обходном режиме бины не включают в себя обновления контекста или вероятности.

Видеокодер 20 может выдавать битовый поток, который включает в себя энтропийно кодированные синтаксические элементы. Битовый поток может также включать в себя синтаксические элементы, которые не являются энтропийно кодированными. Битовый поток может включать в себя последовательность битов, которая образует представление кодированных изображений и соответствующих данных. Битовый поток может содержать последовательность единиц уровня сетевой абстракции (network abstraction layer - NAL). Каждая из единиц NAL включает в себя заголовок единицы NAL и заключает в себе полезную нагрузку последовательности необработанных байтов (raw byte sequence payload - RBSP). Заголовок единицы NAL может включать в себя синтаксический элемент, который указывает на код типа единицы NAL. Код типа единицы NAL, заданный заголовком единицы NAL, указывает на тип единицы NAL. RBSP может быть синтаксической структурой, содержащей целое число байтов, которые заключены в пределах единицы NAL. В некоторых примерах RBSP включает в себя нулевые биты.

Разные типы единиц NAL могут заключать в себе разные типы RBSP. Например, первый тип единицы NAL может заключать в себе RBSP для набора параметров изображения (picture parameter set - PPS), второй тип единицы NAL может заключать в себе RBSP для кодированного слайса, третий тип единицы NAL может заключать в себе RBSP для информации для дополнительной оптимизации (supplemental enhancement information - SEI), и т.д. Единицы NAL, которые заключают в себе RBSP для данных кодирования видео (в противоположность RBSP для наборов параметров и SEI-сообщений), могут называться единицами NAL уровня кодирования видео (video coding layer - VCL).

Видеodeкодер 30 может принимать битовый поток, генерируемый видеокодером 20. Дополнительно, видеodeкодер 30 может синтаксически разбирать битовый поток для декодирования синтаксических элементов из битового потока. Видеodeкодер 30 может восстанавливать изображения видеоданных на основе, по меньшей мере частично, синтаксических элементов, декодированных из битового потока. Процесс восстановления видеоданных может быть, в общем, обратным процессу, выполняемому видеокодером 20. Например, видеodeкодер 30 может использовать MV PU для определения предиктивных блоков для интер-предсказанных PU текущей CU. Подобным образом, видеodeкодер 30 может генерировать интра-предсказанные блоки для PU текущей CU. Дополнительно, видеodeкодер 30 может обратно квантовать блоки коэффициентов преобразования, связанные с TU текущей CU. Видеodeкодер 30 может выполнить обратные преобразования в отношении блоков коэффициентов преобразования для восстановления блоков преобразования, связанных с TU текущей CU. Видеodeкодер 30 может восстанавливать блоки кодирования текущей CU посредством добавления отсчетов предиктивных блоков для PU текущей CU к соответствующим остаточным значениям, получаемым на основании обратного квантования и обратного преобразования блоков преобразования TU текущей CU. Посредством восстановления блоков кодирования для каждой CU изображения, видеodeкодер 30 может восстановить изображение.

В некоторых примерах видеокодер 20 и видеodeкодер 30 могут быть выполнены с возможностью выполнения основанного на палитре кодирования. Например, в основанном на палитре кодировании, вместо реализации технологий интра-предиктивного или интер-предиктивного кодирования, описанных

выше, видеокодер 20 и видеодекодер 30 могут кодировать так называемую палитру в виде таблицы цветов или пиксельных значений, представляющих видеоданные конкретной области (например, данного блока). Таким образом, вместо кодирования фактических пиксельных значений или их остаточных данных для текущего блока видеоданных, видеокодер может кодировать индексные значения для одного или нескольких пиксельных значений текущего блока, причем индексные значения указывают на записи в палитре, которые используются для представления пиксельных значений текущего блока.

Например, видеокодер 20 может кодировать блок видеоданных посредством определения палитры для блока, размещения записи в палитре для представления значения каждого пиксела, и кодирования палитры и индексных значений для пикселов, связывающих пиксельное значение с палитрой. Видеодекодер 30 может получить, из кодированного битового потока, палитру для блока, а также индексные значения для пикселов блока. Видеодекодер 30 может согласовать индексные значения отдельных пикселов с записями палитры для восстановления пиксельных значений блока. В примерах, где индексное значение, связанное с отдельным пикселом, не согласуется с ни с одним из индексных значений соответствующей палитры для блока, видеодекодер 30 может идентифицировать такой пиксел как переходной пиксел для целей основанного на палитре кодирования.

Как описано более подробно ниже, основная идея основанного на палитре кодирования состоит в том, что для данного блока видеоданных, подлежащих кодированию, видеокодер 20 может получить палитру, которая включает в себя наиболее преобладающие пиксельные значения в текущем блоке. Например, палитра может относиться к числу пиксельных значений, которые, как определено или как предполагается, являются преобладающими и/или характерными для текущей CU. Видеокодер 20 может сначала передать размер и элементы палитры к видеодекодеру 30. Дополнительно, видеокодер 20 может кодировать пиксельные значения в данном блоке согласно некоторому порядку сканирования. Для каждого пиксела, включенного в данный блок, видеокодер 20 может сигнализировать индексное значение, которое отображает пиксельное значение в соответствующей записи в палитре. Если пиксельное значение не включено в палитру (т.е. не существует никакой записи палитры, которая задавала бы конкретное пиксельное значение кодированного в режиме палитры блока), то тогда такой пиксел определяют как "переходной пиксел". Согласно основанному на палитре кодированию видеокодер 20 может кодировать и сигнализировать индексное значение, которое зарезервировано для переходного пиксела. В некоторых примерах видеокодер 20 может также кодировать и сигнализировать пиксельное значение (или его квантованную версию) для переходного пиксела, включенного в данный блок. Например, видеодекодер 30 может быть выполнен с возможностью определения того, согласуется ли пиксельное значение с записью палитры или, иначе говоря, является ли оно близким к ней на основе меры расхождения (например, MSE, SAD и т.п.).

После приема кодированного битового потока видео, сигнализированного видеокодером 20, видеодекодер 30 может сначала определить палитру на основе информации, принятой от видеокодера 20. Видеодекодер 30 может затем отобразить принятые индексные значения, связанные с пиксельными местоположениями в данном блоке, в записях палитры для восстановления пиксельных значений данного блока. В некоторых примерах видеодекодер 30 может определить, что пиксел кодированного в режиме палитры блока является переходным пикселом, например, посредством определения того, что пиксел является кодированным в режиме палитры с индексным значением, зарезервированным для переходных пикселов. В примерах, где видеодекодер 30 идентифицирует переходной пиксел в кодированном в режиме палитры блоке, видеодекодер 30 может принять пиксельное значение (или его квантованную версию) для переходного пиксела, включенного в данный блок. Видеодекодер 30 может восстановить кодированный в режиме палитры блок посредством отображения отдельных пиксельных значений в соответствующих записях палитры и посредством использования пиксельного значения (или его квантованной версии) для восстановления любых переходных пикселов, включенных в кодированный в режиме палитры блок.

Как указано выше, в примере режима кодирования палитры палитра может включать в себя записи, нумерованные по индексу. Каждая запись может представлять значения или интенсивности цветовых компонентов (например, в цветовых пространствах, таких как YCbCr, RGB, YUV, CMYK, или других форматах), которые могут быть использованы в качестве предиктора для блока или в качестве конечных восстановленных отсчетов блока. Как описано в стандартном поданном документе JCTVC-Q0094 (Wei Pu и др., "AHG10: Suggested Software for Palette Coding based on RExt6.0", JCTVC-Q0094, Valencia, ES, 27 марта - 4 апреля 2014 г.), палитра может включать в себя записи, которые копируются из палитры предиктора. Палитра предиктора может включать в себя записи палитры из блоков, кодированных ранее с использованием режима палитры или других восстановленных отсчетов. Для каждой записи в палитре предиктора, отправляют двоичный флаг для указания на то, копируется ли эта запись в текущую палитру (указывается flag=1 (флаг=1)). Он называется двоичным вектором предсказания палитры. Дополнительно, текущая палитра может содержать (например, состоять из) новые записи, сигнализируемые явным образом. Число новых записей может также сигнализироваться.

В качестве другого примера в режиме палитры палитра может включать в себя записи, нумерованные по индексу, представляющему значения цветовых компонентов, которые могут быть использованы в качестве предикторов для отсчетов блока или в качестве конечных восстановленных отсчетов блока. Та-

кая запись в палитре может содержать, например, один компонент яркости (например, значение яркости), два компонента цветности (например, два значения цветности) или три цветовых компонента (например, RGB, YUV и т.д.). Ранее декодированные записи палитры могут храниться в списке. Этот список может быть использован для предсказания записей палитры в текущем режиме палитры CU, например. Двоичный вектор предсказания может сигнализироваться в битовом потоке для указания на то, какие записи в списке повторно используются в текущей палитре. В некоторых примерах кодирование длин серий может быть использовано для сжатия двоичного предиктора палитры. Например, значение длины серии может быть кодировано с использованием экспоненциального кода Голомба (Exp-Golomb) 0-го порядка.

В этом раскрытии сущности изобретения будет предполагаться, что каждая запись палитры задает значения для всех цветовых компонентов отсчета. Однако идеи этого раскрытия сущности изобретения применимы к использованию отдельной палитры и/или отдельной записи палитры для каждого цветового компонента. Также предполагается, что отсчеты в блоке обрабатывают с использованием горизонтального растрового порядка сканирования. Однако другие сканирования, такие как вертикальный растровый порядок сканирования, также являются применимыми. Как упомянуто выше, палитра может содержать предсказанные записи палитры, например предсказанные на основании палитры (палитр), используемой для кодирования предшествующего блока (блоков), и новые записи, которые могут быть конкретными для текущего блока и сигнализируются явным образом. Кодер и декодер могут знать число предсказанных и новых записей палитры, и их сумма может указывать на общий размер палитры в блоке.

Как предложено в примере JCTVC-Q0094, приведенном выше, каждый отсчет в блоке, кодированный с использованием палитры, может относиться к одному из трех режимов, указанных ниже:

переходной режим. В этом режиме значение отсчета не включают в палитру в качестве записи палитры, а квантованное значение отсчета сигнализируют явным образом для всех цветовых компонентов. Это является подобным сигнализации новых записей палитры, хотя для новых записей палитры, значения цветовых компонентов не квантуют;

режим CopyAbove (КопироватьСверху) (также называемый режимом CopyFromTop (КопироватьСверху)). В этом режиме индекс записи палитры для текущего отсчета копируют из отсчета, расположенного прямо над текущим отсчетом в блоке отсчетов. В других примерах для режима копирования сверху блок видеоданных может быть перемещен таким образом, чтобы отсчет над этим блоком был фактически отсчетом слева от этого блока;

режим значений (также называемый индексным режимом). В этом режиме значение индекса записи палитры сигнализируют явным образом.

Как описано здесь, индекс записи палитры может называться индексом палитры или просто индексом. Эти термины могут быть использованы взаимозаменяемо для описания технологий этого раскрытия сущности изобретения. Дополнительно, как описано более подробно ниже, индекс палитры может иметь одно или несколько соответствующих значений цвета или интенсивности. Например, индекс палитры может иметь единственное соответствующее значение цвета или интенсивности, связанное с единственным компонентом цвета или интенсивности пикселя (например, Красным компонентом RGB-данных, Y-компонентом YUV-данных, и т.п.). В другом примере индекс палитры может иметь множественные соответствующие значения цвета или интенсивности. В некоторых примерах основанное на палитре кодирование видео может быть применено для кодирования монохромного видео. Соответственно "значение цвета" может, в общем, относиться к любому цветовому компоненту или не-цветовому компоненту, используемому для генерирования пиксельного значения.

Значение серии может указывать на серию значений индексов палитры, которые кодируются с использованием одного и того же режима кодирования палитры. Например, в отношении режима значений видеокодер (например, видеокодер 20 или видеокодер 30) может кодировать индексное значение и значение серии, которое указывает число последовательных последующих отсчетов в порядке сканирования, которые имеют одно и то же индексное значение и которые кодируются с использованием индекса палитры. В отношении режима CopyAbove видеокодер может кодировать указание на то, что индексное значение для текущего значения отсчета является тем же самым, что и значение индекса верхнего соседнего отсчета (например, отсчета, который расположен над отсчетом, кодируемым в данный момент, в блоке), и значение серии, которое указывает число последовательных последующих отсчетов в порядке сканирования, которые также копируют индексное значение из верхнего соседнего отсчета.

Соответственно в приведенных выше примерах серия значений индексов палитры относится к серии значений палитры, имеющих одно и то же значение, или к серии индексных значений, которые копируются из верхних соседних отсчетов.

Следовательно, серия может задавать для данного режима число последующих отсчетов, которые относятся к одному и тому же режиму. В некоторых примерах сигнализация индексного значения и значения серии может быть подобной кодированию длин серий. В одном примере в целях иллюстрации, строкой последовательных индексных значений палитры индексного блока, соответствующего блоку видеоданных, может быть 0, 2, 2, 2, 2, 5. Каждое индексное значение соответствует отсчету в блоке ви-

деоданных. В этом примере видеокодер может кодировать второй отсчет (например, первое индексное значение палитры, равное "2") с использованием режима значений. После кодирования индексного значения, равного 2, видеокодер может кодировать серию, равную 3, которая указывает на то, что три последующих отсчета также имеют то же самое индексное значение, равное 2. Подобным образом, кодирование серии из четырех индексов палитры после кодирования индекса с использованием режима *Сору-Above* может указывать на то, что всего пять индексов палитры копируют из соответствующих индексных значений палитры в строке, находящейся над положением отсчета, кодируемого в данный момент.

С использованием палитры видеокодер 20 и/или видеodeкодер 30 могут быть выполнены с возможностью кодирования блока отсчетов (например, блока видеоданных) в индексный блок, причем индексный блок является блоком, включающим в себя индексные значения, которые отображаются в одной или нескольких записях палитры и в некоторых примерах в одном или нескольких значениях переходных пикселей. Видеокодер 20 может быть выполнен с возможностью энтропийного кодирования индексного блока для сжатия индексного блока. Подобным образом, видеodeкодер 30 может быть выполнен с возможностью энтропийного декодирования кодированного индексного блока для генерирования индексного блока, из которого видеodeкодер 30 может сгенерировать блок отсчетов (например, блок видеоданных, кодированных видеокодером 20). Например, основанное на длинах серий энтропийное кодирование может быть использовано для сжатия и распаковки индексного блока. В некоторых примерах видеокодер 20 и видеodeкодер 30 могут быть выполнены с возможностью соответственно энтропийного кодирования и декодирования индексного блока с использованием САВАС.

Для применения САВАС-кодирования к информации (например, синтаксическому элементу, индексному блоку, например к индексным значениям индексного блока, или другой информации) видеокодер (например, видеокодер 20 и видеodeкодер 30) может выполнить преобразование в двоичную форму в отношении информации. Преобразование в двоичную форму относится к процессу преобразования информации в последовательность из одного или нескольких битов. Каждая последовательность из одного или нескольких битов может называться "бином". Преобразование в двоичную форму является процессом без потерь и может включать в себя одну или комбинацию из следующих технологий кодирования: кодирование с фиксированной длиной, унарное кодирование, усеченное унарное кодирование, усеченное кодирование Райса (Rice), кодирование Голомба (Golomb), экспоненциальное кодирование Голомба, кодирование Голомба-Райса, любая форма кодирования Голомба, любая форма кодирования Райса, и любая форма энтропийного кодирования. Например, преобразование в двоичную форму может включать в себя представление целого значения, равного 5, в виде 00000101 с использованием технологии 8-битовой фиксированной длины, или в виде 11110 с использованием технологии унарного кодирования.

После преобразования в двоичную форму видеокодер может идентифицировать контекст кодирования. Контекст кодирования может идентифицировать вероятности того, что бины кодирования имеют конкретные значения. Например, контекст кодирования может указывать на вероятность, равную 0,7, кодирования 0-значного бина, и на вероятность, равную 0,3, кодирования 1-значного бина. После идентификации контекста кодирования, видеокодер может арифметически кодировать то, что бин основан на данном контексте, что известно как кодирование в контекстном режиме. Бины, кодированные с использованием кодирования в контекстном режиме САВАС, могут называться "контекстными бинами".

Дополнительно, вместо выполнения кодирования в контекстном режиме в отношении всех бинов, видеокодер (например, видеокодер 20 и видеodeкодер 30) может кодировать некоторые бины с использованием обходного кодирования САВАС (например, кодирование в обходном режиме). Кодирование в обходном режиме относится к процессу арифметического кодирования бина без использования адаптивного контекста (например, контекста кодирования). А именно, механизм обходного кодирования не выбирает контексты и может предполагать вероятность, равную 0,5, для обоих символов (0 и 1). Хотя кодирование в обходном режиме может не быть таким эффективным по полосе пропускания, как кодирование в контекстном режиме, выполнение кодирования в обходном режиме в отношении бина может быть менее затратным по вычислительным ресурсам, чем выполнение кодирования в контекстном режиме в отношении бина. Дополнительно, выполнение кодирования в обходном режиме может обеспечить большую степень параллелизма и большую производительность. Бины, кодированные с использованием кодирования в обходном режиме, могут называться "обходными бинами".

Видеокодер 20 и видеodeкодер 30 могут быть сконфигурированы с использованием САВАС-кодера (например, САВАС-кодера и САВАС-декодера соответственно). САВАС-кодер может включать в себя механизм кодирования в контекстном режиме для выполнения кодирования в контекстном режиме САВАС и механизм кодирования в обходном режиме для выполнения кодирования в обходном режиме. Если бин кодируют в контекстном режиме, то для кодирования этого бина используют механизм кодирования в контекстном режиме. Механизм кодирования в контекстном режиме может нуждаться более чем в двух циклах обработки для кодирования единственного бина. Однако при использовании соответствующей конвейерной конструкции, механизм кодирования в контекстном режиме может нуждаться только в $n+M$ циклах для кодирования n бинов, где M является затратами для запуска конвейера. M является обычно большим 0.

При запуске процесса кодирования САВАС (т.е. при каждом переключении из обходного режима в

контекстный режим и наоборот) вводятся затраты на конвейер. Если бин кодируют в обходном режиме, то для кодирования этого бина используют механизм кодирования в обходном режиме. Можно предполагать, что механизм кодирования в обходном режиме будет нуждаться только в одном цикле для кодирования n -битовой информации, где n может быть большим одного. Таким образом, общее число циклов для кодирования набора обходных бинов и контекстных бинов может быть уменьшено, если все обходные бины в пределах набора кодируются вместе и все контекстные бины в пределах набора кодируются вместе. Конкретно, кодирование обходных бинов вместе перед или после перехода к кодированию в контекстном режиме может сэкономить затраты, требуемые для повторного запуска механизма кодирования в контекстном режиме. Например, видеокодер 20 и декодер 30 могут быть выполнены с возможностью переключения между обходным режимом и контекстным режимом соответственно при кодировании или декодировании блока видеоданных с использованием режима палитры. В другом примере видеокодер 20 и декодер 30 могут быть выполнены с возможностью уменьшения числа раз, которое процесс кодирования или декодирования переключается между обходным режимом и контекстным режимом при кодировании или декодировании блока видеоданных с использованием режима палитры.

Технологии, описанные в этом раскрытии сущности изобретения, могут включать в себя технологии для различных комбинаций одного или нескольких из сигнализации основанных на палитре режимов кодирования видео, передачи палитр, получения палитр, сигнализации порядка сканирования, получения порядка сканирования и передачи основанных на палитре карт кодирования видео и других синтаксических элементов. Например, технологии этого раскрытия сущности изобретения могут быть направлены на энтропийное кодирование информации палитры. В некоторых примерах технологии этого раскрытия сущности изобретения могут быть, среди прочего, использованы для увеличения эффективности кодирования и уменьшения неэффективности кодирования, связанных с основанным на палитре кодированием видео. Соответственно, как описано более подробно ниже, технологии этого раскрытия сущности изобретения могут, в некоторых примерах, улучшить эффективность и увеличить скорость передачи в битах при кодировании видеоданных с использованием режима палитры.

Как описано выше, в текущей конструкции режима палитры при кодировании контента экрана синтаксические элементы `palette_index_idc` и `palette_escape_val` кодируют в обходном режиме CABAC и чередуют с другими синтаксическими элементами (например, `palette_run_msb_id_plus1`, которые кодируются в контекстном режиме CABAC. Однако может быть предпочтительным группирование вместе кодированной в обходном режиме информации (например, синтаксических элементов), что может улучшить эффективность кодирования и/или уменьшить сложность кодера.

Синтаксический элемент `palette_index_idc` может быть указанием индекса массива, представленного `currentPaletteEntries` (текущими Записями Палитры), определенным, например, в JCTVC-S1005. Значение `palette_index_idc` может находиться в диапазоне от 0 до (`adjustedIndexMax` (настроенный Максимальный-Индекс)-1), включительно. Синтаксический элемент `palette_escape_val` может задавать квантованное переходное кодированное значение отсчета для компонента, определенное, например, в JCTVC-S1005. `palette_run_msb_id_plus1` минус 1 может задавать индекс наиболее значимого бита в двоичном представлении `paletteRun` (Серия Палитры), определенном, например, в JCTVC-S1005.

Переменная `paletteRun` может задавать число последовательных местоположений минус 1 с тем же самым индексом палитры, что и положение в верхней строке, когда `palette_run_type_flag` (флаг типа серии палитры) равен `COPY_ABOVE_MODE` (РЕЖИМ КОПИРОВАНИЯ СВЕРХУ), или задает число последовательных местоположений минус 1 с тем же самым индексом палитры, когда `palette_run_type_flag` равен `COPY_INDEX_MODE` (РЕЖИМ КОПИРОВАНИЯ ИНДЕКСА), как определено, например, в JCTVC-S1005. Дополнительные подробности относительно `palette_index_idc`, `palette_escape_val`, `palette_run_msb_id_plus1`, `currentPaletteEntries`, `adjustedIndexMax` и `paletteRun` могут быть найдены в JCTVC-S1005.

В некоторых примерах это раскрытие сущности изобретения описывает способ группирования всех синтаксических элементов `palette_index_idc` в передней части секции кодирования индексных блоков палитры для улучшения производительности CABAC. Например, видеокодер 20 может быть выполнен с возможностью кодирования всех синтаксических элементов `palette_index_idc` в передней части секции кодирования индексных блоков палитры. Например, видеокодер 20 может быть выполнен с возможностью кодирования всех синтаксических элементов `palette_index_idc` перед кодированием синтаксических элементов, подлежащих кодированию в контекстном режиме. Подобным образом, декодер 30 может быть выполнен с возможностью декодирования всех синтаксических элементов `palette_index_idc` в передней части секции кодирования индексных блоков палитры. Например, декодер 30 может быть выполнен с возможностью декодирования всех синтаксических элементов `palette_index_idc` перед декодированием синтаксических элементов, кодированных в контекстном режиме.

В качестве другого примера видеокодер 20 может быть выполнен с возможностью кодирования в обходном режиме всех синтаксических элементов `palette_index_idc` в передней части секции кодирования индексных блоков палитры таким образом, чтобы все синтаксические элементы `palette_index_idc` были кодированы перед кодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом `CopyAbove` или индексным режимом) и/или длиной серии (например, `pal-`

ette_run_msb_id_plus1). Подобным образом, видеodecoder 30 может быть выполнен с возможностью декодирования всех синтаксических элементов palette_index_idc для блока в передней части секции кодирования индексных блоков палитры блока таким образом, чтобы все синтаксические элементы palette_index_idc были декодированы перед декодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1).

Синтаксический элемент (элементы), связанный с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1).

В качестве другого примера видеodecoder 20 может быть выполнен с возможностью кодирования всех синтаксических элементов palette_index_idc перед контекстным кодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1). Подобным образом, видеodecoder 30 может быть выполнен с возможностью кодирования всех синтаксических элементов palette_index_idc перед контекстным декодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1).

В качестве другого примера видеodecoder 20 может быть выполнен с возможностью кодирования всех синтаксических элементов palette_index_idc в пределах секции кодирования блоков палитры перед кодированием синтаксических элементов, подлежащих кодированию в контекстном режиме. Подобным образом, видеodecoder 30 может быть выполнен с возможностью декодирования всех синтаксических элементов palette_index_idc в пределах секции кодирования блоков палитры перед декодированием кодированных в контекстном режиме синтаксических элементов. В качестве другого примера видеodecoder 20 может быть выполнен с возможностью кодирования всех синтаксических элементов palette_index_idc в пределах секции кодирования блоков палитры перед контекстным кодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1). Подобным образом, видеodecoder 30 может быть выполнен с возможностью декодирования всех синтаксических элементов palette_index_idc в пределах секции кодирования блоков палитры перед контекстным декодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1).

В общем, видеodecoder 20 и видеodecoder 30 могут быть выполнены с возможностью не чередовать кодирование или декодирование palette_index_idc в обходном режиме с синтаксическими элементами, которые подлежат кодированию или декодированию с использованием контекстного режима соответственно. Например, видеodecoder 20 и видеodecoder 30 могут быть выполнены с возможностью не чередовать кодирование или декодирование palette_index_idc в обходном режиме с синтаксическим элементом (элементами), связанным с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1), который подлежит кодированию или декодированию с использованием контекстного режима соответственно. В качестве другого примера видеodecoder 20 может быть выполнен с возможностью обходного кодирования всех экземпляров синтаксического элемента palette_index_idc перед контекстным кодированием синтаксического элемента, который требует контекстного режима. Подобным образом, видеodecoder 30 может быть выполнен с возможностью обходного декодирования всех экземпляров синтаксического элемента palette_index_idc перед контекстным декодированием синтаксического элемента, который требует контекстного режима. В качестве другого примера видеodecoder 20 может быть выполнен с возможностью обходного кодирования всех экземпляров синтаксического элемента palette_index_idc перед контекстным кодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1). Подобным образом, видеodecoder 30 может быть выполнен с возможностью обходного декодирования всех экземпляров синтаксического элемента palette_index_idc перед контекстным декодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом CopyAbove или индексным режимом) и/или длиной серии (например, palette_run_msb_id_plus1).

Видеodecoder 20 и видеodecoder 30 могут также соответственно кодировать и декодировать значение, представляющее число событий palette_index_idc. Видеodecoder 20 и видеodecoder 30 могут использовать значение, представляющее число событий palette_index_idc, соответственно для кодирования или декодирования каждого из синтаксических элементов palette_index_idc. Технологии, описанные в этом раскрытии сущности изобретения, могут также устранять избыточность связанных с длиной серии палитры синтаксических элементов и устранять избыточность palette_run_type_flag и palette_index_idc.

В некоторых примерах это раскрытие сущности изобретения описывает способ группирования всех синтаксических элементов palette_escape_val в передней части секции кодирования индексных блоков палитры блока (например, PU или CU) для улучшения производительности CABAC. Например, видеodecoder 20 может быть выполнен с возможностью кодирования всех синтаксических элементов palette_escape_val в передней части секции кодирования индексных блоков палитры блока. Например,

видеокодер 20 может быть выполнен с возможностью кодирования в обходном режиме всех синтаксических элементов `palette_escape_val` в передней части секции кодирования индексных блоков палитры таким образом, чтобы все синтаксические элементы `palette_escape_val` были кодированы перед кодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом `CopyAbove` или индексным режимом) и/или длиной серии (например, `palette_run_msb_id_plus1`). Подобным образом, видеодекодер 30 может быть выполнен с возможностью декодирования всех синтаксических элементов `palette_escape_val` для блока в передней части секции кодирования индексных блоков палитры блока таким образом, чтобы все синтаксические элементы `palette_escape_val` были декодированы перед декодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом `CopyAbove` или индексным режимом) и/или длиной серии (например, `palette_run_msb_id_plus1`). В качестве другого примера видеокодер 20 может быть выполнен с возможностью кодирования всех синтаксических элементов `palette_escape_val` перед кодированием синтаксических элементов, подлежащих кодированию в контекстном режиме. Например, видеокодер 20 может быть выполнен с возможностью кодирования всех синтаксических элементов `palette_escape_val` перед контекстным кодированием синтаксического элемента (элементов), связанного с типом серии палитры (например, режимом `CopyAbove` или индексным режимом) и/или длиной серии (например, `palette_run_msb_id_plus1`). Подобным образом, видеодекодер 30 может быть выполнен с возможностью декодирования всех синтаксических элементов `palette_escape_val` для блока в передней части секции кодирования индексных блоков палитры блока. Например, видеодекодер 30 может быть выполнен с возможностью декодирования всех синтаксических элементов `palette_escape_val` перед декодированием кодированных в контекстном режиме синтаксических элементов в блоке.

В качестве другого примера видеокодер 20 может быть выполнен с возможностью кодирования всех синтаксических элементов `palette_escape_val` в пределах секции кодирования блоков палитры блока перед кодированием синтаксических элементов, подлежащих кодированию в контекстном режиме. Подобным образом, видеодекодер 30 может быть выполнен с возможностью декодирования всех синтаксических элементов `palette_escape_val` в пределах секции кодирования блоков палитры блока перед декодированием кодированных в контекстном режиме синтаксических элементов блока.

В общем, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью не чередовать кодирование или декодирование `palette_escape_val` для блока (например, PU или CU) в обходном режиме с синтаксическими элементами, которые подлежат кодированию или декодированию с использованием контекстного режима для блока соответственно. Например, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью не чередовать кодирование или декодирование `palette_escape_val` в обходном режиме с синтаксическим элементом (элементами), связанным с типом серии палитры (например, режимом `CopyAbove` или индексным режимом) и/или длиной серии (например, `palette_run_msb_id_plus1`), который подлежит кодированию или декодированию с использованием контекстного режима соответственно. В качестве другого примера видеокодер 20 может быть выполнен с возможностью обходного кодирования всех экземпляров синтаксического элемента `palette_escape_val` для блока перед контекстным кодированием синтаксического элемента, который требует контекстного режима. Подобным образом, видеодекодер 30 может быть выполнен с возможностью обходного декодирования всех экземпляров синтаксического элемента `palette_escape_val` блока (например, PU или CU) перед контекстным декодированием синтаксического элемента, который требует контекстного режима блока.

Видеокодер 20 и видеодекодер 30 могут также соответственно кодировать и декодировать значение, представляющее число событий `palette_escape_val` для блока. Видеокодер 20 и видеодекодер 30 могут использовать значение, представляющее число событий `palette_escape_val`, соответственно для кодирования или декодирования каждого из синтаксических элементов `palette_escape_val` для блока. Технологии, описанные в этом раскрытии сущности изобретения, могут уменьшить динамический диапазон `palette_index_idc` для блока, что может привести к улучшению эффективности кодирования.

Технологии, аспекты и/или примеры, описанные здесь, могут быть использованы в сочетании друг с другом в любой комбинации или отдельно друг от друга. Например, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью выполнения любой или любой подходящей комбинации одного или нескольких из технологий, аспектов и/или примеров, описанных здесь.

В некоторых примерах для улучшения производительности САВАС видеокодер (например, видеокодер 20) может быть выполнен с возможностью группирования всех событий синтаксического элемента `palette_index_idc`, как описано выше. Например, видеокодер (например, видеокодер 20) может быть выполнен с возможностью группирования всех событий синтаксического элемента `palette_index_idc` в текущем блоке (например, PU или CU) в передней части секции кодирования индексов для текущего блока. Подобным образом, видеодекодер (например, видеодекодер 30) может быть выполнен с возможностью декодирования всех синтаксических элементов `palette_index_idc`, как описано выше. Фиг. 7 показывает один пример, где видеокодер 20 может быть выполнен с возможностью группирования всех событий синтаксического элемента `palette_index_idc` в текущем блоке (например, CU), например в передней части блока кодирования индексов, в соответствии с R. Joshi и J. Xu, "High efficient video coding (HEVC)

screen content coding: Draft 2", JCTVC-S1005, секция 7.3.3.8. Этот аспект раскрытия сущности изобретения называется аспектом 1. Конкретно, фиг. 7 показывает пример видеокодера 20, перемещающего экземпляр синтаксического элемента `palette_index_idc` в переднюю часть блока кодирования индексов (которая может также называться секцией кодирования блоков палитры или передней частью блока кодирования индексов). Посредством перемещения показанного примера синтаксического элемента `palette_index_idc` видеокодер 20 может быть выполнен с возможностью улучшения производительности САВАС посредством кодирования всех экземпляров синтаксического элемента `palette_index_idc` с использованием обходного режима и переключения снова в контекстный режим для кодирования информации палитры, встречающейся после кодирования в обходном режиме всех экземпляров синтаксического элемента `palette_index_idc` в блоке кодирования индексов.

Согласно раскрытию JCTVC-S1005 один пример `palette_index_idc` может быть кодирован в обходном режиме, затем один экземпляр синтаксического элемента, связанного с типом серии палитры, и один пример `palette_run_msb_id_plus1` могут быть кодированы в контекстном режиме, и процесс может повторяться, пока не будет выполнено условие ($\text{scanPos} < \text{nCbS} * \text{nCbS}$), что означает, что видеокодер может переключаться туда и обратно между кодированием в обходном режиме и кодированием в контекстном режиме, поскольку синтаксические элементы, подлежащие кодированию с использованием обходного режима, не сгруппированы вместе. Это показано на фиг. 7 логической блок-схемой, отображающей условное событие непосредственно под циклом с проверкой условия "`while(scanPos < nCbS * nCbS)`" (т.е. логическая блок-схема, отображающая условное событие, исключает информацию, показывающую, что синтаксический элемент, связанный с типом серии палитры, кодирован с использованием контекстного режима), причем рамка, окружающая условный оператор с логически следующим синтаксическим элементом `palette_index_idc`, находится под циклом с проверкой условия "`while(scanPos < nCbS * nCbS)`", и последующим псевдокодом. Однако, как описано выше, фиг. 7 также описывает аспект 1 этого раскрытия сущности изобретения, который является группированием (которое может также называться перемещением) одного или нескольких экземпляров синтаксического элемента `palette_index_idc`, например, в переднюю часть блока кодирования индексов. Посредством перемещения одного или нескольких синтаксических элементов (например, или другой информации палитры), подлежащих кодированию с использованием обходного режима, видеокодер (например, видеокодер 20) может увеличить производительность энтропийного кодирования посредством уменьшения числа раз, которое видеокодер или декодер должен переключать между кодированием в обходном режиме и кодированием в контекстном режиме. Подобным образом, посредством перемещения одного или нескольких синтаксических элементов таким образом производительность декодера (например, декодера 30) может увеличиться, поскольку число раз, которое декодер должен переключаться между декодированием в обходном режиме и декодированием в контекстном режиме, уменьшается. В некоторых примерах технологий, описанных в этом раскрытии сущности изобретения, все примеры синтаксического элемента `palette_index_idc` могут быть кодированы в обходном режиме перед тем, как пример `palette_run_msb_id_plus1` сможет быть кодирован в контекстном режиме.

В некоторых примерах видеокодер 20 может быть выполнен с возможностью сигнализации числа событий (например, примеров) синтаксического элемента `palette_index_idc` с использованием синтаксического элемента, именуемого, например, `num_palette_index` (число_индексов_палитры). Например, видеокодер 20 может сигнализировать значение для `num_palette_index` в битовый поток, причем это значение указывает число событий синтаксического элемента `palette_index_idc`. В некоторых примерах видеокодер 20 может быть выполнен с возможностью не сигнализировать индексное значение в виде `palette_index_idc`. В таких примерах декодер 30 может быть выполнен с возможностью получения индексного значения. Например, событие `palette_index_idc` может подсчитываться в `num_palette_index`, который может быть равен числу раз, чей тип серии (например, `COPY_INDEX_MODE` (КОПИРОВАНИЕ_В_ИНДЕКСНОМ_РЕЖИМЕ)) встречается в конкретном блоке. Даже когда получают тип серии (например, `COPY_INDEX_MODE`) или получают `palette_index_idc`, это все же подсчитывается в `num_palette_index`. При использовании здесь ссылка на число синтаксически разобранных индексов, декодированных или подлежащих декодированию, может в некоторых примерах относиться к числу `COPY_INDEX_MODE`, независимо от того, получен ли режим или индекс. Декодер 30 может быть выполнен с возможностью определения числа событий (например, примеров) синтаксического элемента `palette_index_idc` посредством, например, декодирования кодированного значения, соответствующего синтаксическому элементу `num_palette_index`, из битового потока. Этот аспект раскрытия сущности изобретения называется аспектом 2. Видеокодер 20 и декодер 30 могут быть выполнены с возможностью реализации аспекта 1 с аспектом 2 или без аспекта 2. Что касается синтаксиса, аспект 2 может быть согласно некоторым примерам определен в виде:

indices_idc_coding() {	
num_palette_index	ae(v)
for (i = 0; i < num_palette_index; i++)	
palette_index_idc	ae(v)
}	

В некоторых примерах видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью реализации (например, посредством разблокирования) аспектов 1 и 2 только тогда, когда переменная indexMax (МаксимальноеЧислоИндексов) является большей чем 1. Этот аспект раскрытия сущности изобретения называется аспектом 3. Переменная indexMax может задавать число различных значений, которые имеет индекс палитры, для текущей единицы кодирования. В некоторых примерах indexMax может относиться к значению, равному (размеру палитры+palette_escape_val_present_flag (флаг наличия переходных значений палитры)).

В некоторых примерах аспекты 1 и 2 могут быть заблокированы, когда: (а) не существует никаких переходных пикселей (т.е. palette_escape_val_present_flag== 0) в текущем блоке и размер палитры является меньшим, чем 3; или (б) может существовать по меньшей мере один переходной пиксел (т.е. palette_escape_val_present_flag== 1) в текущем блоке, и размер палитры равен 0. В других примерах, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью реализации (например, посредством разблокирования) аспектов 1 и 2 только тогда, когда переменная indexMax является большей чем 2. Подобным образом, в примерах, где indexMax является равной (размеру палитры+palette_escape_val_present_flag), аспекты 1 и 2 могут быть разблокированы (например, реализованы), когда indexMax является большей чем 1. Например, если размер палитры равен 0 и palette_escape_val_present_flag равен 1, то все пиксели в блоке являются переходными пикселями; и, по существу, индексы уже известны. В качестве другого примера, если palette_escape_val_present_flag равен 0, и размер палитры равен 1, снова, то каждый пиксел имеет индекс, равный 0; и, по существу, никакая сигнализация индексов не требуется.

В некоторых примерах видеокодер 20 может быть выполнен с возможностью реализации аспектов 1 и 2 таким образом, чтобы последнее событие (например, пример) синтаксического элемента palette_run_type_flag[xC][yC] (флаг типа серии палитры[xC][yC]) сигнализировалось видеокодером 20 в передней части секции кодирования индексных блоков палитры. Этот аспект раскрытия сущности изобретения называется аспектом 4. Конкретно, синтаксическая таблица может быть обновлена согласно некоторому примеру посредством добавления нового синтаксического элемента palette_last_run_type_flag (флаг последнего события типа серии палитры) следующим образом:

indices_idc_coding() {	
num_palette_index	ae(v)
for (i = 0; i < num_palette_index; i++)	
palette_index_idc	ae(v)
palette_last_run_type_flag	ae(v)
}	

Видеодекодер 30 может быть выполнен с возможностью определения последнего события (например, примера) синтаксического элемента palette_run_type_flag[xC][yC], например, посредством декодирования кодированного синтаксического элемента palette_last_run_type_flag из битового потока. Синтаксический элемент palette_last_run_type_flag может быть кодирован в обходном режиме или кодирован в контекстном режиме, например, в CABAC. В примерах, где синтаксический элемент palette_last_run_type_flag кодирован в контекстном режиме, синтаксический элемент palette_last_run_type_flag может совместно использовать один и тот же контекст (контексты) с palette_run_type_flag[xC][yC], или синтаксический элемент palette last run type flag может иметь свой собственный контекст (контексты), который является независимым от контекста (контекстов) palette_run_type_flag[xC][yC].

В некоторых примерах видеодекодер 30 может быть выполнен с возможностью декодирования синтаксического элемента palette_index_idc таким образом, чтобы процесс настройки динамического диапазона был заблокирован для первого события (например, примера) синтаксического элемента palette_index_idc. Этот аспект раскрытия сущности изобретения называется аспектом 5. Конкретно, используется процесс, очень похожий на процедуру получения переменной adjustedIndexMax (настроенноеМаксимальноеЧислоИндексов), заданную в JCTVC-S1005, секция 7.4.9.6. В целях сравнения JCTVC-S1005 описывает, что переменная adjustedIndexMax может быть получена следующим образом:

```
adjustedIndexMax=indexMax
if( scanPos > 0 )
adjustedIndexMax -=1
```

Однако согласно аспекту 5 этого раскрытия сущности изобретения переменная `adjustedIndexMax` может быть получена, как описано ниже. Например, для каждого блока инициализируют переменную `isFirstIndex` (ПервыйИндекс), равную 1, перед синтаксическим разбором. В некоторых примерах переменная `adjustedIndexMax` может быть получена следующим образом:

```
adjustedIndexMax=indexMax
palette_index_idc
if( isFirstIndex ) {
    adjustedIndexMax -=isFirstIndex
    isFirstIndex=0
}
```

В некоторых примерах видеodecoder 30 может быть выполнен с возможностью проверки одного или нескольких условий перед синтаксическим разбором и декодированием `paletteRun` (СерияПалитры). Этот аспект раскрытия сущности изобретения называется аспектом 6. Переменная `paletteRun` может задавать число последовательных местоположений минус 1 с одним и тем же индексом палитры в виде положения в верхней строке, когда `palette_run_type_flag` равен `COPY_ABOVE_MODE`, или задавать число последовательных местоположений минус 1 с одним и тем же индексом палитры, когда `palette_run_type_flag` равен `COPY_INDEX_MODE`, как описано в JCTVC-S1005, например.

Со ссылкой на одно или несколько условий, которые видеodecoder 30 может проверить, если видеodecoder 30 определяет, что одно или несколько условий удовлетворены, то видеodecoder 30 может быть выполнен с возможностью обхода процесса синтаксического разбора и декодирования для синтаксических элементов, связанных с текущей `paletteRun` (т.е. `palette_run_msb_id_plus1` и `palette_run_refinement_bits` (биты уточнения серии палитры)). В таком примере видеodecoder 30 может быть выполнен с возможностью получения в явном виде текущей `paletteRun` при переходе к концу текущего блока, т.е. равной `maxPaletteRun`. Список из одного или нескольких условий, связанных с аспектом 6, включает в себя: (i) число синтаксически разобранных/декодированных синтаксических элементов `palette_index_idc` равно `num_palette_index`; или, альтернативно, может быть определена переменная `paletteIndicesLeft` (ОставленныеИндексыПалитры), которая равна `num_palette_index` минус число принятых индексов, и, с использованием такого определения, это условие может быть указано в виде `paletteIndicesLeft` является равной нулю; и/или (ii) текущий `palette_run_type_flag[xC][yC]` типа серии палитры равен последнему `palette_last_run_type_flag` типа серии палитры.

В некоторых примерах, если условия (i) и (ii), приведенные выше для аспекта 6, не удовлетворяются одновременно, то видеodecoder 20 может быть выполнен с возможностью кодирования длины серии палитры в битовом потоке. Этот аспект раскрытия сущности изобретения называется аспектом 7. В других примерах, если условия (i) и (ii), приведенные выше для аспекта 6, не удовлетворяются одновременно, то видеodecoder 20 может быть выполнен с возможностью кодирования длины серии палитры в битовом потоке. Согласно текущему проекту спецификации JCTVC-S1005 в качестве входных данных требуется параметр, задающий максимально доступную длину серии, причем этот параметр равен `maxPaletteRun=nCbS*nCbS - scanPos - 1`. Согласно этому раскрытию сущности изобретения, однако, видеodecoder 20 может быть выполнен с возможностью уменьшения параметра, задающего максимальную доступную длину серии, до `maxPaletteRun=nCbS*nCbS - scanPos - 1 - paletteIndicesLeft`, для улучшения эффективности кодирования. При использовании здесь `nCbS` задает размер текущего блока.

В некоторых примерах на видеodecoder 20 может быть наложено нормативное ограничение, требующее, чтобы он никогда не сигнализировал палитру с неиспользуемыми записями, если блок не находится в режиме совместного использования палитры (т.е. `palette_share_flag[x0][y0]==0`) (флаг совместного использования палитры[x0][y0]==0). Этот аспект раскрытия сущности изобретения называется аспектом 8.

В некоторых примерах для режима палитры, не использующего режим совместного использования палитры, видеodecoder 30 может быть выполнен с возможностью обхода декодирования текущего события (например, примера) синтаксического элемента `palette_index_idc`, когда удовлетворено одно или несколько следующих условий: условие 1, где `num_palette_index` равно `indexMax`, и условие 2, где `paletteIndicesLeft==1`. В таких примерах видеodecoder 20 может быть выполнен с возможностью явного получения значения текущего события синтаксического элемента `palette_index_idc` в виде индекса, который находится в палитре, но еще только должен появиться в карте индексов во время процесса декодирования (например, не появлялся в карте индексов вплоть до этого момента в процессе декодирования). Этот аспект раскрытия сущности изобретения называется аспектом 9.

Видеodecoder 30 может быть выполнен с возможностью получения значения для текущего события синтаксического элемента `palette_index_idc`, как описано выше для аспекта 9, поскольку условие 1 требует, чтобы каждый индекс между 0 и (`indexMax - 1`) включительно был сигнализирован и был сигнализирован только один раз. Таким образом, после сигнализации первых (`indexMax - 1`) индексных значений видеodecoder 30 может быть выполнен с возможностью получения последнего индексного значения в виде числа между 0 и (`indexMax - 1`), которое еще только должно появиться во время процесса декодирования для текущей карты индексов.

В некоторых примерах видеodeкодер 30 может быть выполнен с возможностью обхода декодирования текущего события (например, примера) синтаксического элемента `palette_run_type_flag[xC][yC]`, когда удовлетворено одно или оба следующих условия: условие 1, где `paletteIndicesLeft` равно 0, и условие 2, где текущий пиксел находится в последнем положении блока в порядке сканирования. В таких примерах видеodeкодер 30 может быть выполнен с возможностью явного получения значения для текущего события синтаксического элемента `palette_run_type_flag[xC][yC]`. Например, когда удовлетворено условие 1, видеodeкодер 30 `palette_run_type_flag[xC][yC]` может быть выполнен с возможностью получения значения для текущего события синтаксического элемента `palette_run_type_flag[xC][yC]` в виде `COPY_ABOVE_MODE`. В качестве другого примера, когда удовлетворено условие 1, видеodeкодер 30 `palette_run_type_flag[xC][yC]` может быть выполнен с возможностью получения значения для текущего события синтаксического элемента `palette_run_type_flag[xC][yC]` в виде `COPY_INDEX_MODE`, если `paletteIndicesLeft > 0`, и в виде `COPY_ABOVE_MODE` (РЕЖИМ КОПИРОВАНИЯ СВЕРХУ), если `paletteIndicesLeft=0`. Этот аспект раскрытия сущности изобретения называется аспектом 10.

Как описано здесь, видеodeкодер 20 и видеodeкодер 30 могут быть выполнены с возможностью определения того, когда условие удовлетворено. Например, в отношении аспекта 10 видеodeкодер 30 может быть выполнен с возможностью определения того, удовлетворено ли условие 1. Подобным образом, видеodeкодер 30 может быть выполнен с возможностью определения того, удовлетворено ли условие 2. В ответ на определение того, что удовлетворено условие 1 или условие 2, видеodeкодер 30 может быть выполнен с возможностью получения значения для текущего события синтаксического элемента `palette_run_type_flag[xC][yC]`, как описано выше.

В некоторых примерах видеodeкодер 20 и видеodeкодер 30 могут быть выполнены с возможностью соответственно кодирования или декодирования синтаксического элемента `num_palette_index` с использованием любого семейства кодов Голомба. Например, видеodeкодер 20 и видеodeкодер 30 могут быть выполнены с возможностью соответственно кодирования или декодирования синтаксического элемента `num_palette_index` с использованием, например, кода Голомба-Райса, экспоненциального кода Голомба, усеченного кода Райса, унарного кода или сочленения кода Голомба-Райса и экспоненциального кода Голомба. Этот аспект раскрытия сущности изобретения называется аспектом 11.

В других примерах видеodeкодер 20 и видеodeкодер 30 могут быть выполнены с возможностью соответственно кодирования или декодирования синтаксического элемента `num_palette_index` с использованием любой усеченной версии любого семейства кодов Голомба. Например, видеodeкодер 20 и видеodeкодер 30 могут быть выполнены с возможностью соответственно кодирования или декодирования синтаксического элемента `num_palette_index` с использованием, например, усеченного кода Голомба-Райса, усеченного экспоненциального кода Голомба, усеченного Усеченного кода Райса, усеченного унарного кода или сочленения усеченного кода Райса и экспоненциального кода Голомба, такого как код используемый для кодирования синтаксических элементов `coeff_abs_level_remaining`. Этот аспект раскрытия сущности изобретения называется аспектом 12.

В некоторых примерах любые параметры Голомба, связанные с аспектами 11 или 12, зависят от размера `CU`, `indexMax`, размера палитры, и/или `palette_escape_val_present_flag`. Такая зависимость может быть выражена в виде уравнений или справочной таблицы. В некоторых примерах видеodeкодер 20 может быть выполнен с возможностью сигнализации справочной таблицы или параметров в уравнениях таким образом, чтобы они были приняты видеodeкодером 30, например, в заголовке `SPS/PPS/слайса`. Альтернативно или дополнительно, параметры могут быть адаптивно обновлены на поблочной основе. Этот аспект раскрытия сущности изобретения называется аспектом 13. В некоторых примерах параметр Голомба `cRiceParam` может зависеть от `indexMax`, размера палитры и/или `palette_escape_val_present_flag`. Параметр Голомба `cRiceParam` может изменяться от блока к блоку.

В некоторых примерах видеodeкодер 20 может быть выполнен с возможностью предиктивного кодирования `num_palette_index` посредством сигнализации разности между значением `num_palette_index` и значением сдвига, которое может быть выражено синтаксическим элементом, именуемым, например, `numPaletteIndexCoded` (числоКодированныхИндексовПалитры). Этот аспект раскрытия сущности изобретения называется аспектом 14. Например, видеodeкодер 20 может быть выполнен с возможностью предиктивного кодирования `num_palette_index` посредством сигнализации значения для `numPaletteIndexCoded`, где `numPaletteIndexCoded=num_palette_index - IndexOffsetValue` (ЗначениеСдвига-Индексов). Подобным образом, видеodeкодер 30 может быть выполнен с возможностью предиктивного декодирования `num_palette_index` посредством, например, определения значения для `numPaletteIndexCoded` из битового потока. Поскольку `numPaletteIndexCoded=num_palette_index - IndexOffsetValue`, видеodeкодер 30 может быть выполнен с возможностью определения значения `num_palette_index` на основе определенного значения `numPaletteIndexCoded` и значения `IndexOffsetValue`.

В некоторых примерах переменная `IndexOffsetValue` может быть постоянной. Например, `IndexOffsetValue` может равняться постоянному значению `X` для режима совместного использования палитры или может равняться постоянному значению `Y` для не-режима совместного использования палитры, где `X` и `Y` являются целыми. В некоторых примерах `X` и `Y` могут быть одинаковыми (например, `X` равно `Y` и равно, например, 1). В других примерах `X` и `Y` могут быть разными (например, `X` не равно `Y`). Например,

IndexOffsetValue может равняться 9, когда используется режим совместного использования палитры, и IndexOffsetValue может равняться 33, когда используется не-режим совместного использования. В некоторых примерах переменная IndexOffsetValue может зависеть от синтаксического элемента palette_share_flag[x0][y0]. В других примерах переменная IndexOffsetValue может зависеть от переменной indexMax. Например, IndexOffsetValue может равняться indexMax. В некоторых примерах видеокoder 20 может быть выполнен с возможностью сигнализации IndexOffsetValue в заголовке SPS/PPS/Слайса. Альтернативно или дополнительно, переменная IndexOffsetValue может быть адаптивно обновлена поблоч-но, что означает, что значение, соответствующее переменной IndexOffsetValue, может быть адаптивно обновлено поблоч-но.

В некоторых примерах видеокoder 20 и видеокoder 30 могут быть выполнены с возможностью соответственно кодирования или декодирования numPaletteIndexCoded, которая может быть кодирована с использованием любого семейства кодов Голомба или любого усеченного семейства кодов Голомба, например сочленения кода Голомба-Райса и экспоненциального кода Голомба. Например, когда IndexOffsetValue равно 1, numPaletteIndexCoded равно num_palette_index - 1.

В некоторых примерах видеокoder 20 и видеокoder 30 могут быть выполнены с возможностью соответственно кодирования или декодирования numPaletteIndexCoded с использованием любого семейства кодов Голомба. Например, видеокoder 20 и видеокoder 30 могут быть выполнены с возможностью соответственно кодирования или декодирования numPaletteIndexCoded с использованием, например, кода Голомба-Райса, экспоненциального кода Голомба, усеченного кода Райса, унарного кода, или сочленения кода Голомба-Райса и экспоненциального кода Голомба.

В других примерах, видеокoder 20 и видеокoder 30 могут быть выполнены с возможностью соответственно кодирования или декодирования numPaletteIndexCoded с использованием только усеченной версии семейства кодов Голомба. Например, видеокoder 20 и видеокoder 30 могут быть выполнены с возможностью соответственно кодирования или декодирования numPaletteIndexCoded с использованием, например, усеченного кода Голомба-Райса, усеченного экспоненциального кода Голомба, усеченного Усеченного кода Райса, усеченного унарного кода, или сочленения усеченного кода Райса и экспоненциального кода Голомба, такого как код, используемый для кодирования синтаксических элементов coeff abs level remaining.

Для кодирования numPaletteIndexCoded видеокoder 20 может быть выполнен с возможностью определения знака numPaletteIndexCoded. Видеокoder 20 может быть выполнен с возможностью определения знака флага, указывающего на то, является ли значение numPaletteIndexCoded отрицательным или нет (например, является ли определенный знак положительным или отрицательным). Этот аспект раскрытия сущности изобретения называется аспектом 15. В некоторых примерах видеокoder 20 может быть выполнен с возможностью сигнализации флага и затем сигнализации значения numPaletteIndexCoded. В других примерах видеокoder 20 может быть выполнен с возможностью сигнализации значения numPaletteIndexCoded и затем сигнализации флага. Видеокoder 20 может быть выполнен с возможностью кодирования флага с использованием обходного режима или контекстного режима. При контекстном кодировании контексты могут зависеть от размера CU, indexMax, размера палитры и/или palette_escape_val_present_flag.

Как описано выше, видеокoder 20 может быть выполнен с возможностью определения знака numPaletteIndexCoded согласно некоторым примерам. Если определенный знак numPaletteIndexCoded является отрицательным, то видеокoder 20 может быть выполнен с возможностью кодирования значения (1 - numPaletteIndexCoded) в битовом потоке. Если определенный знак numPaletteIndexCoded является положительным, то видеокoder 20 может быть выполнен с возможностью кодирования значения numPaletteIndexCoded в битовом потоке. Видеокoder 20 может быть выполнен с возможностью кодирования значения (1 - numPaletteIndexCoded) или значения numPaletteIndexCoded с использованием разных параметров кода Голомба, в зависимости, например, от знака numPaletteIndexCoded, размера CU, indexMax, размера палитры и/или palette_escape_val_present_flag.

В некоторых примерах видеокoder 20 может быть выполнен с возможностью представления отрицательных частей numPaletteIndexCoded с использованием операции отображения, которая может быть дополнительной или альтернативной к аспекту 15. Этот аспект раскрытия сущности изобретения называется аспектом 16. Например, интервал отображения может быть введен и определен как переменная mapInterval (ИнтервалОтображения). Видеокoder 20 может быть выполнен с возможностью, с использованием переменной mapInterval, отображения отрицательных значений numPaletteIndexCoded в равным образом разнесенных положительных значениях, равных: $\text{mapInterval} \times (-\text{numPaletteIndexCoded}) - 1$. Соответствующее положительное значение numPaletteIndexCoded может быть сдвинуто соответственно для размещения положений, занимаемых отображенными отрицательными значениями.

Например, если mapInterval=2, и numPaletteIndexCoded выбирается из {-3, -2, -1, 0, 1, 2, 3}, то тогда отображение может быть показано в виде табл. I, приведенной ниже. В этом примере видеокoder 20 может быть выполнен с возможностью кодирования значений numPaletteIndexCode с использованием отображенных значений в табл. I. Например, видеокoder 20 может быть выполнен с возможностью энтропийного кодирования отображенных значений в двоичной форме.

Таблица I

Пример отображения кодовых слов

numPaletteIndexCoded	отображенное значение
-3	5
-2	3
-1	1
0	0
1	2
2	4
3	6

В некоторых примерах видеокодер 20 может быть выполнен с возможностью представления отрицательных частей numPaletteIndexCoded с использованием операции отображения, описанной в отношении аспекта 16. Видеокодер 20 может быть также выполнен с возможностью устранения одной или нескольких избыточностей, которые могут присутствовать при реализации аспекта 16. Этот аспект раскрытия сущности изобретения называется аспектом 17. Например, отрицательные значения numPaletteIndexCoded могут находиться в диапазоне $A=\{-1, -2, \dots, -\text{IndexOffsetValue}+1\}$. В качестве другого примера отрицательные значения numPaletteIndexCoded могут находиться в диапазоне $A=\{-1, -2, \dots, -\text{IndexOffsetValue}+1, \text{IndexOffsetValue}\}$. В любом из этих примеров отображенное значение должно резервировать только $(\text{IndexOffsetValue} - 1)$ или IndexOffsetValue положений для отрицательных значений numPaletteIndexCoded. Например, если $\text{mapInterval}=2$ и numPaletteIndexCoded выбирается из $\{-3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8\}$, то отображение показано в табл. II, приведенной ниже. В этом примере видеокодер 20 может быть выполнен с возможностью кодирования значений numPaletteIndexCode с использованием отображенных значений numPaletteIndexCode в табл. II. Например, видеокодер 20 может быть выполнен с возможностью энтропийного кодирования отображенных значений в двоичной форме.

Таблица II

Пример отображения кодовых слов

numPaletteIndexCoded	отображенное значение
-3	5
-2	3
-1	1
0	0
1	2
2	4
3	6
4	7
5	8
6	9
7	10
8	11

Как показано в табл. II, приведенной выше, видеокодер 20 может быть выполнен с возможностью кодирования отображенных значений, соответствующих значениям numPaletteIndexCode, таким образом, чтобы отрицательные и положительные значения numPaletteIndexCode не чередовались после некоторого значения. Например, в примере табл. II, приведенной выше, не существует чередования положительных и отрицательных значений numPaletteIndexCoded посредством отображенных значений, начиная со значения 3 numPaletteIndexCoded (т.е. положительные значения 3-8 numPaletteIndexCoded отображаются в отображенные значения 6-11).

Как описано выше, видеокодер 20 может быть также выполнен с возможностью устранения одной или нескольких избыточностей, которые могут присутствовать, при реализации аспекта 16. Другой пример избыточности, отличный от примера избыточности, описанного выше, включает в себя: поскольку num_palette_index является ограниченным сверху общим числом пикселей в текущем блоке, numPaletteIndexCoded также является ограниченным сверху. Таким образом, после выделения положений для всех возможностей положительного кодового слова отрицательные значения могут быть отображены в следующих положениях без чередования. Например, если $\text{mapInterval}=2$ и numPaletteIndexCoded выбирается из $\{-5, -4, -3, -2, -1, 0, 1, 2, 3\}$, то отображение показано в табл. III, приведенной ниже. В этом примере видеокодер 20 может быть выполнен с возможностью кодирования значений numPaletteIndexCode с использованием отображенных значений в табл. III. Например, видеокодер 20 может быть выполнен с

возможностью энтропийного кодирования отображенных значений в двоичной форме.

Таблица III

Пример отображения кодовых слов

numPaletteIndexCoded	отображенное значение
-5	8
-4	7
-3	5
-2	3
-1	1
0	0
1	2
2	4
3	6

Как показано в табл. III, приведенной выше, видеокодер 20 может быть выполнен с возможностью кодирования отображенных значений, соответствующих значениям numPaletteIndexCode, таким образом, чтобы отрицательные и положительные значения numPaletteIndexCode не чередовались после некоторого значения. Например, в примере табл. III, приведенной выше, не существует чередования положительных и отрицательных значений numPaletteIndexCoded посредством отображенных значений, начиная со значения 4 numPaletteIndexCoded (т.е. отрицательные значения -4 и -5 numPaletteIndexCoded отображаются в отображенные значения 7 и 8).

В некоторых примерах видеокодер 20 может быть выполнен с возможностью, дополнительно, нарушения связи между индексом палитры и серией палитры. Этот аспект раскрытия сущности изобретения называется аспектом 18. Например, вместо обеспечения контекстов кодирования серии палитры в зависимости от синтаксически разобранных или декодированных индексов видеокодер 20 может быть выполнен с возможностью обеспечения контекстов кодирования серии палитры в зависимости от предыдущей длины серии палитры или в зависимости от palette_run_msb_id_plus1, indexMax, и/или размера CU предыдущей серии.

В некоторых примерах для дополнительного группирования обходных бинов видеокодер 20 может быть выполнен с возможностью сигнализации числа переходных индексов в блоке палитры, а также переходных значений перед сигнализацией типа серии палитры (т.е. palette_run_type_flag[xC][yC]), как указано ниже. Этот аспект раскрытия сущности изобретения называется аспектом 19. Части с курсивным начертанием показывают изменения относительно предыдущей версии (версий) JCT-VC S1005, а части с полужирным начертанием, а также "ae(v)" в правом столбце указывают на сигнализацию синтаксического элемента.

...	
if(currentPaletteSize != 0)	
palette_escape_val_present_flag	ae(v)
if(palette_escape_val_present_flag (indexMax > 0))	
escape_idc_coding()	
if(palette_escape_val_present_flag) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded)	
{	
cu_qp_delta_palette_abs	ae(v)
if(cu_qp_delta_palette_abs)	
cu_qp_delta_palette_sign_flag	ae(v)
}	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_palette_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_palette_offset_idx	ae(v)
}	
}	
if(indexMax > 0)	
palette_transpose_flag	ae(v)
scanPos=0	
while(scanPos < nCbS * nCbS) {	
...	

В примере, приведенном выше, `escape_idc_coding()` (кодирование_переходных_индексов()) состоит из сигнализации числа переходных индексов и переходных значений, соответствующих каждому переходному индексу. Число переходных индексов в блоке палитры может не сигнализироваться, если `palette_escape_val_present_flag` равен 0 или если `indexMax` равен 0. В первом случае делают вывод, что число переходных индексов равно 0 и никакие переходные значения не сигнализируются. В последнем случае, когда `indexMax` равен 0, делают вывод, что число переходных индексов равно размеру блока, когда `palette_escape_val_present_flag` равен 1, и сигнализируют переходные значения, или делают вывод, что число переходных индексов равно 0, когда `palette_escape_val_present_flag` равен 0. `equals 0`.

В некоторых примерах видеокодер 20 может быть выполнен с возможностью сигнализации числа переходных индексов с использованием семейства кодов Голомба. Этот аспект раскрытия сущности изобретения называется аспектом 20. Например, видеокодер 20 может быть выполнен с возможностью сигнализации данного числа переходных индексов с использованием, например, кода Голомба-Райса, экспоненциального кода Голомба, усеченного кода Райса, унарного кода, или сочленения кода Голомба-Райса и экспоненциального кода Голомба. Усеченные версии вышеупомянутых кодов могут быть использованы с максимальным множеством, равным размеру блока.

В некоторых примерах предлагается требовать нормативного ограничения в отношении `palette_escape_val_present_flag`, состоящего в том, что когда `palette_escape_val_present_flag` равен 0, не существует никаких переходных пикселей в текущем блоке. Этот аспект раскрытия сущности изобретения называется аспектом 21. Когда `palette_escape_val_present_flag` равен 1, существует по меньшей мере один переходной пиксел в текущем блоке. С использованием этого ограничения, при `escape_idc_coding()` может быть кодировано число переходных индексов минус 1, вместо числа переходных индексов, для улучшения эффективности кодирования. В этом случае, максимальное значение для усеченного семейства кодов Голомба может быть настроено на $(blockSize-1)$ соответственно.

В некоторых примерах, когда число переходных индексов сигнализируют перед кодированием блока карты индексов, и когда все переходные индексы уже кодированы, тогда `indexMax` может быть уменьшено на 1. Кроме того, если `indexMax` становится равным 1, то кодирование индекса, серии и режима прекращается, поскольку индексы для всех оставшихся отсчетов могут быть получены. Этот аспект раскрытия сущности изобретения называется аспектом 22. В качестве одного примера аспекта 22 предположим, что размер палитры равен 1, и `palette_escape_val_present_flag` равен 1. Обычно возможны-

ми индексными значениями являются 0 и 1, причем 1 используется для переходного пиксела (пикселей). В соответствии с аспектом 22 видеокодер 20 может быть выполнен с возможностью сигнализации числа переходных значений/отсчетов. Тогда, когда индексы сигнализируются и встречается последнее переходное значение/отсчет, как видеокодер 20, так и/или видеодекодер 30 могут быть выполнены с возможностью получения (например, определения) того, что больше не существует переходных значений/отсчетов. По существу, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью определения того, что единственным индексным значением, которое может встретиться от последнего переходного значения/отсчета до конца блока, является 0, что означает, что видеокодер 20 может быть выполнен с возможностью не сигнализировать режим, индексное значение, и/или значение серии от последнего переходного значения/отсчета до конца блока.

В некоторых примерах `escape_idc_coding()` используется в комбинации с `indices_idc_coding()` (кодирование_индексов()). Этот аспект раскрытия сущности изобретения называется аспектом 23. В одном примере число переходных индексов может сигнализироваться перед сигнализацией числа индексов. В этом случае только число не-переходных индексов должно сигнализироваться в `escape_idc_coding()`. В одном примере число переходных индексов может сигнализироваться после сигнализации числа индексов. В этом случае максимальное значение для усеченного семейства кодов Голомба может быть настроено на `num_palette_index` соответственно.

Видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью функционирования согласно технологиям, описанным в этом раскрытии сущности изобретения. В общем, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью определения того, что текущий блок кодирован в режиме палитры, кодирования в обходном режиме множества экземпляров первого синтаксического элемента для восстановления текущего блока, и после кодирования в обходном режиме множества экземпляров первого синтаксического элемента, декодирования в контекстном режиме множества примеров второго синтаксического элемента для восстановления текущего блока.

Фиг. 2 является блок-схемой, показывающей иллюстративный видеокодер 20, который может реализовать технологии этого раскрытия сущности изобретения. Фиг. 2 обеспечена в целях объяснения и не должна рассматриваться как ограничение технологий, в широком смысле поясненных и описанных в этом раскрытии сущности изобретения. В целях объяснения это раскрытие сущности изобретения описывает видеокодер 20 в контексте кодирования HEVC и, например, SCC-расширения HEVC. Однако технологии этого раскрытия сущности изобретения могут быть применимыми к другим стандартам или способам кодирования.

Видеокодер 20 представляет собой пример устройства, которое может быть выполнено с возможностью реализации технологий для основанного на палитре кодирования и энтропийного кодирования (например, САВАС) согласно различным примерам, описанным в этом раскрытии сущности изобретения.

В примере фиг. 2 видеокодер 20 включает в себя блок 100 кодирования блоков, память 101 видеоданных, блок 102 генерирования остаточных данных, блок 104 обработки преобразований, блок 106 квантования, блок 108 обратного квантования, блок 110 обработки обратных преобразований, блок 112 восстановления, блок 114 фильтрации, буфер 116 декодированных изображений и блок 118 энтропийного кодирования. Блок 100 кодирования блоков включает в себя блок 120 обработки интер-предсказаний и блок 126 обработки интра-предсказаний. Блок 120 обработки интер-предсказаний включает в себя блок оценки движения и блок компенсации движения (не показаны). Видеокодер 20 также включает в себя блок 122 основанного на палитре кодирования, выполненный с возможностью выполнения различных аспектов технологий основанного на палитре кодирования, описанных в этом раскрытии сущности изобретения. В других примерах видеокодер 20 может включать в себя дополнительные функциональные компоненты, меньшее число функциональных компонентов, или другие функциональные компоненты.

Память 101 видеоданных может хранить видеоданные, подлежащие кодированию компонентами видеокодера 20. Видеоданные, хранимые в памяти 101 видеоданных, могут быть получены, например, из источника 18 видео. Буфер 116 декодированных изображений может быть памятью опорных изображений, которая хранит опорные видеоданные для использования в кодировании видеоданных видеокодером 20, например, в режимах интра- или интер-кодирования. Память 101 видеоданных и буфер 116 декодированных изображений могут быть образованы любым из множества запоминающих устройств, таких как динамическая память с произвольным доступом (dynamic random access memory - DRAM), в том числе синхронная DRAM (synchronous DRAM - SDRAM), магниторезистивная RAM (magnetoresistive RAM - MRAM), резистивная RAM (resistive RAM - RRAM), или другие типы запоминающих устройств. Память 101 видеоданных и буфер 116 декодированных изображений могут быть обеспечены одним и тем же запоминающим устройством или отдельными запоминающими устройствами. В различных примерах память 101 видеоданных может находиться в микросхеме вместе с другими компонентами видеокодера 20 или может быть отдельной от этих компонентов.

Видеокодер 20 может принимать видеоданные. Видеокодер 20 может кодировать каждую CTU в слайсе изображения видеоданных. Каждая из CTU может быть связана с равноразмерными древовидными блоками кодирования (coding tree block - CTB) яркости и соответствующими CTB изображения. В качестве части кодирования CTU блок 100 кодирования блоков может выполнить разбиение посредством

дерева квадрантов, для разделения СТВ СТU на все меньшие и меньшие блоки. Меньший блок может быть блоком кодирования CU. Например, блок 100 кодирования блоков может разбить СТВ, связанный с СТU, на четыре равноразмерных подблока, разбить один или несколько подблоков на четыре равноразмерных под-подблока и т.д.

Видеокодер 20 может кодировать CU СТU для генерирования кодированных представлений CU (т.е. кодированных CU). В качестве части кодирования CU блок 100 кодирования блоков может разбить блоки кодирования, связанные с CU, среди одной или нескольких PU CU. Таким образом, каждая PU может быть связана с блоком предсказания яркости и соответствующими блоками предсказания цветности. Видеокодер 20 и видеодекодер 30 могут поддерживать PU, имеющие различные размеры. Как указано выше, размер CU может относиться к размеру блока кодирования яркости CU, а размер PU может относиться к размеру блока предсказания яркости PU. Предположим, что размер конкретной CU равен $2N \times 2N$, видеокодер 20 и видеодекодер 30 могут поддерживать размеры PU, равные $2N \times 2N$ или $N \times N$ для интра-предсказания, и симметричные размеры PU, равные $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, или подобные, для интер-предсказания. Видеокодер 20 и видеодекодер 30 могут также поддерживать асимметричное разбиение для размеров PU, равных $2N \times nU$, $2N \times nD$, $nL \times 2N$ и $nR \times 2N$, для интер-предсказания.

Блок 120 обработки интер-предсказаний может генерировать предиктивные данные для PU посредством выполнения интер-предсказания в отношении каждой PU CU. Предиктивные данные для PU могут включать в себя предиктивные блоки PU и информацию о движении для PU. Блок 121 интер-предсказаний может выполнять разные операции для PU CU в зависимости от того, находится ли PU в I-слайсе, P-слайсе или B-слайсе. В I-слайсе все PU являются интра-предсказанными. Следовательно, если PU находится в I-слайсе, то блок 121 интер-предсказаний не выполняет никаких интерпредсказаний в отношении PU. Таким образом, для блоков, кодированных в I-режиме, предсказанный блок образуют с использованием пространственного предсказания на основании ранее кодированных соседних блоков в пределах того же самого кадра.

Если PU находится в P-слайсе, то блок оценки движения блока 120 обработки интер-предсказаний может искать опорные изображения в списке опорных изображений (например, "RefPicList0") для опорной области для PU. Опорная область для PU может быть областью, в пределах опорного изображения, которая содержит блоки отсчетов, которые наиболее точно соответствуют блокам отсчетов PU. Блок оценки движения блока 120 обработки интер-предсказаний может генерировать опорный индекс, который указывает на положение в RefPicList0 опорного изображения, содержащего опорную область для PU. Дополнительно, блок оценки движения может генерировать MV, который указывает на пространственное смещение между блоком кодирования PU и опорным местоположением, связанным с опорной областью. Например, MV может быть двумерным вектором, который обеспечивает сдвиг из координат в текущем декодированном изображении в координаты в опорном изображении. Блок оценки движения может выдавать опорный индекс и MV в виде информации о движении PU. Блок компенсации движения блока 120 обработки интер-предсказаний может генерировать предиктивные блоки PU на основе фактических или интерполированных отсчетов в опорном местоположении, указанном вектором движения PU.

Если PU находится в B-слайсе, то блок оценки движения может выполнить одинарное предсказание или двойное предсказание для PU. Для выполнения одинарного предсказания для PU блок оценки движения может искать опорные изображения в RefPicList0 или во втором списке опорных изображений ("RefPicList1") для опорной области для PU. Блок оценки движения может выдавать, в виде информации о движении PU, опорный индекс, который указывает на положение в RefPicList0 или RefPicList1 опорного изображения, которое содержит опорную область, MV, который указывает на пространственное смещение между предиктивным блоком PU и опорным местоположением, связанным с опорной областью, и один или несколько указателей направления предсказания, которые указывают на то, находится ли опорное изображение в RefPicList0 или в RefPicList1. Блок компенсации движения блока 120 обработки интер-предсказаний может генерировать предиктивные блоки PU на основе, по меньшей мере частично, фактических или интерполированных отсчетов в опорной области, указанной вектором движения PU.

Для выполнения двунаправленного интер-предсказания для PU блок оценки движения может искать опорные изображения в RefPicList0 для опорной области для PU и может также искать опорные изображения в RefPicList1 для другой опорной области для PU. Блок оценки движения может генерировать индексы опорных изображений, которые указывают на положения в RefPicList0 и RefPicList1 опорных изображений, которые содержат опорные области. Дополнительно, блок оценки движения может генерировать MV, которые указывают на пространственные смещения между опорным местоположением, связанным с опорными областями, и блоком отсчетов PU. Информация о движении PU может включать в себя опорные индексы и MV PU. Блок компенсации движения может генерировать предиктивные блоки PU на основе, по меньшей мере частично, фактических или интерполированных отсчетов в опорных областях, указанных векторами движения PU.

Согласно различным примерам этого раскрытия сущности изобретения видеокодер 20 может быть выполнен с возможностью выполнения основанного на палитре кодирования. В отношении инфраструктуры HEVC, в качестве примера, технологии основанного на палитре кодирования могут быть выполнены с возможностью использования на уровне CU. В других примерах, технологии основанного на палит-

ре кодирования видео могут быть выполнены с возможностью использования на уровне PU. В других примерах технологии основанного на палитре кодирования могут быть выполнены с возможностью использования на уровне под-единицы предсказания (под-PU) (например, под-блока единицы предсказания). Соответственно все раскрытые процессы, описанные здесь (во всем этом раскрытии сущности изобретения), в контексте уровня CU, могут, дополнительно или альтернативно, применяться к уровню PU или уровню под-PU. Однако эти основанные на HEVC примеры не должны рассматриваться как запрет или ограничение технологий основанного на палитре кодирования видео, описанных здесь, поскольку такие технологии могут быть применены для независимой работы или в виде части других существующих или еще подлежащих разработке систем/стандартов. В этих случаях единицами для кодирования палитры могут быть квадратные блоки, прямоугольные блоки или даже области непрямоугольной формы.

Блок 122 основанного на палитре кодирования, например, может выполнять основанное на палитре декодирование, когда выбран режим основанного на палитре кодирования, например, для CU или PU. Например, блок 122 основанного на палитре кодирования может быть выполнен с возможностью генерирования палитры, имеющей записи, указывающие на пиксельные значения, выбора пиксельных значений в палитре для представления пиксельных значений по меньшей мере некоторых положений блока видеоданных, и сигнализации информации, связывающей по меньшей мере некоторые из положений блока видеоданных с записями в палитре, соответствующими соответственно выбранным пиксельным значениям. Хотя описано, что различные функции выполняются блоком 122 основанного на палитре кодирования, некоторые или все такие функции могут быть выполнены другими блоками обработки, или комбинацией разных блоков обработки.

Согласно аспектам этого раскрытия сущности изобретения блок 122 основанного на палитре кодирования может быть выполнен с возможностью выполнения любой комбинации технологий для кодирования палитры, описанных здесь.

Блок 126 обработки интра-предсказаний может генерировать предиктивные данные для PU посредством выполнения интра-предсказания в отношении PU. Предиктивные данные для PU могут включать в себя предиктивные блоки для PU и различные синтаксические элементы. Блок 126 обработки интра-предсказаний может выполнять интра-предсказание в отношении PU в I-слайсах, P-слайсах или B-слайсах.

Для выполнения интра-предсказания в отношении PU блок 126 обработки интра-предсказаний может использовать множественные режимы интра-предсказания для генерирования множественных наборов предиктивных данных для PU. Блок 126 обработки интра-предсказаний может использовать отсчеты из блоков отсчетов соседних PU для генерирования предиктивного блока для PU. Соседние PU могут находиться выше, выше и справа, выше и слева или слева от PU, что предполагает порядок кодирования слева-направо, сверху-вниз для PU, CU и STU. Блок 126 обработки интра-предсказаний может использовать различные количества режимов интра-предсказания, например 33 направленных режима интра-предсказания. В некоторых примерах количество режимов интра-предсказания может зависеть от размера области, связанной с PU.

Блок 100 кодирования блоков может выбирать предиктивные данные для PU CU из предиктивных данных, генерируемых блоком 120 обработки интер-предсказаний для PU, или предиктивных данных, генерируемых блоком 126 обработки интра-предсказаний для PU. В некоторых примерах блок 100 кодирования блоков выбирает предиктивные данные для PU CU на основе мер расхождения/скорости передачи наборов предиктивных данных. Предиктивные блоки выбранных предиктивных данных могут называться здесь выбранными предиктивными блоками.

Блок 102 генерирования остаточных данных может генерировать на основе блока кодирования яркости, Cb и Cr CU, и выбранных предиктивных блоков яркости, Cb и Cr PU CU, остаточных блоков яркости, Cb и Cr CU. Например, блок 102 генерирования остаточных данных может генерировать остаточные блоки CU таким образом, чтобы каждый отсчет в остаточных блоках имел значение, равное разности между отсчетом в блоке кодирования CU и соответствующим отсчетом в соответствующем выбранном предиктивном блоке PU CU.

Блок 104 обработки преобразований может выполнять разбиение посредством дерева квадрантов, для разбиения остаточных блоков, связанных с CU, на блоки преобразования, связанные с TU CU. Таким образом, в некоторых примерах TU может быть связана с блоком преобразования яркости и двумя блоками преобразования цветности. Размеры и положения блоков преобразования яркости и цветности TU CU могут быть основаны или могут быть не основаны на размерах и положениях предиктивных блоков PU CU. Структура дерева квадрантов, известная как "остаточное дерево квадрантов" (residual quad-tree - RQT), может включать в себя узлы, связанные с каждой из областей. TU CU могут соответствовать листовым узлам RQT.

Блок 104 обработки преобразований может генерировать блоки коэффициентов преобразования для каждой TU CU посредством применения одного или нескольких преобразований к блокам преобразования TU. Блок 104 обработки преобразований может применять различные преобразования к блоку преобразования, связанному с TU. Например, блок 104 обработки преобразований может применять дис-

кретное косинусное преобразование (discrete cosine transform - DCT), направленное преобразование или концептуально подобное преобразование к блоку преобразования. В некоторых примерах блок 104 обработки преобразований не применяет преобразование к блоку преобразования. В таких примерах блок преобразования может быть обработан как блок коэффициентов преобразования.

Блок 106 квантования может квантовать коэффициенты преобразования в блоке коэффициентов. Процесс квантования может уменьшить битовую глубину, связанную 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 является большим, чем m . Блок 106 квантования может квантовать блок коэффициентов, связанный с TU CU, на основе значения параметра квантования (quantization parameter - QP), связанного с CU. Видеокодер 20 может настраивать степень квантования, применяемого к блокам коэффициентов, связанным с CU, посредством настройки значения QP, связанного с CU. Квантование может вводить потери информации, таким образом, квантованные коэффициенты преобразования могут иметь меньшую точность, чем исходные коэффициенты преобразования.

Блок 108 обратного квантования и блок 110 обработки обратных преобразований могут применять обратное квантование и обратные преобразования к блоку коэффициентов соответственно для восстановления остаточного блока из блока коэффициентов. Блок 112 восстановления может добавлять восстановленный остаточный блок к соответствующим отсчетам из одного или нескольких предиктивных блоков, генерируемых блоком 100 кодирования блоков, для создания восстановленного блока преобразования, связанного с TU. Посредством восстановления блоков преобразования для каждой TU CU таким образом, видеокодер 20 может восстановить блоки кодирования CU.

Блок 114 фильтрации может выполнять одну или несколько операций деблокирования для уменьшения артефактов блокирования в блоках кодирования, связанных с CU. Блок 114 фильтрации может выполнять другие операции фильтрации, в том числе фильтрацию с адаптивным сдвигом отсчетов (sample adaptive offset - SAO) и/или фильтрацию с адаптивным циклом (adaptive loop filtering - ALF). Буфер 116 декодированных изображений может хранить восстановленные блоки кодирования после выполнения блоком 114 фильтрации одной или нескольких операций деблокирования в отношении восстановленных блоков кодирования. Блок 120 обработки интер-предсказаний может использовать опорное изображение, которое содержит восстановленные блоки кодирования, для выполнения интер-предсказания в отношении PU других изображений. Дополнительно, блок 126 обработки интра-предсказаний может использовать восстановленные блоки кодирования в буфере 116 декодированных изображений для выполнения интра-предсказания в отношении других PU в том же самом изображении, что и CU.

Блок 118 энтропийного кодирования может принимать данные от других функциональных компонентов видеокодера 20. Например, блок 118 энтропийного кодирования может принимать блоки коэффициентов от блока 106 квантования и может принимать синтаксические элементы от блока 100 кодирования блоков. Блок 118 энтропийного кодирования может выполнять одну или несколько операций энтропийного кодирования в отношении данных для генерирования энтропийно кодированных данных. Например, блок 118 энтропийного кодирования может выполнять операцию контекстно-адаптивного кодирования, такую как операция CABAC, операцию контекстно-адаптивного кодирования с переменной длиной (context-adaptive variable length coding - CAVLC), операцию с перекодированием одной переменной длины в другую переменную длину (variable-to-variable - V2V), операцию синтаксического контекстно-адаптивного двоичного арифметического кодирования (syntax-based context-adaptive binary arithmetic coding - SBAC), операцию энтропийного кодирования с разбиением на интервалы вероятности (Probability Interval Partitioning Entropy - PIPE), операцию экспоненциального кодирования Голомба, или операцию энтропийного кодирования другого типа в отношении данных. Видеокодер 20 может выдавать битовый поток, который включает в себя энтропийно кодированные данные, генерируемые блоком 118 энтропийного кодирования. Например, битовый поток может включать в себя данные, которые представляют RQT для CU.

В некоторых примерах кодирование остаточных данных не выполняют с помощью кодирования палитры. Соответственно видеокодер 20 может не выполнять преобразование или квантование при кодировании с использованием режима кодирования палитры. Дополнительно, видеокодер 20 может энтропийно кодировать данные, генерируемые с использованием режима кодирования палитры, отдельно от остаточных данных.

Согласно одной или нескольким технологиям этого раскрытия сущности изобретения видеокодер 20 и, конкретно, блок 122 основанного на палитре кодирования может выполнять основанное на палитре кодирование видео предсказанных блоков видео. Как описано выше, палитра, генерируемая видеокодером 20, может быть явно закодированной и отправленной к видеодекодеру 30, может быть предсказанной на основании предыдущих записей палитры или может быть комбинацией этого.

Согласно одной или нескольким технологиям этого раскрытия сущности изобретения видеокодер 20 может быть выполнен с возможностью определения того, что текущий блок кодирован в режиме палитры, кодирования в обходном режиме множества экземпляров первого синтаксического элемента для восстановления текущего блока, и после кодирования в обходном режиме множества экземпляров перво-

го синтаксического элемента для восстановления текущего блока, кодирования в контекстном режиме множества примеров второго синтаксического элемента для восстановления текущего блока, например, с использованием процесса кодирования CABAC. Видеокодер 20 может быть выполнен с возможностью кодирования в обходном режиме любых двух примеров из множества экземпляров первого синтаксического элемента, например, с использованием обходного режима процесса кодирования CABAC, без чередования с кодированием в контекстном режиме любого примера из множества примеров второго синтаксического элемента. В одном примере первый синтаксический элемент содержит одно из синтаксического элемента `palette_index_idc` или синтаксического элемента `palette_escape_val` и второй синтаксический элемент содержит синтаксический элемент `palette_run_msb_id_plus1`. Видеокодер 20 может быть выполнен с возможностью обходного кодирования множества экземпляров первого синтаксического элемента в передней части секции кодирования индексных блоков для текущего блока.

Видеокодер 20 может быть выполнен с возможностью кодирования третьего синтаксического элемента, указывающего число экземпляров первого синтаксического элемента, причем кодирование в обходном режиме множества экземпляров первого синтаксического элемента содержит кодирование в обходном режиме множества экземпляров первого синтаксического элемента на основе третьего синтаксического элемента. Видеокодер 20 может кодировать третий синтаксический элемент с использованием одного из кода Голомба-Райса, экспоненциального кода Голомба, усеченного кода Райса, унарного кода, или сочленения кода Голомба-Райса и экспоненциального кода Голомба, или усеченной версии любого из предыдущих кодов.

Фиг. 3 является блок-схемой, показывающей иллюстративный видеodeкодер 30, который выполнен с возможностью реализации технологий этого раскрытия сущности изобретения. Фиг. 3 обеспечена в целях объяснения и не должна рассматриваться как ограничение технологий, в широком смысле поясненных и описанных в этом раскрытии сущности изобретения. В целях объяснения это раскрытие сущности изобретения описывает видеodeкодер 30 в контексте кодирования HEVC. Однако технологии этого раскрытия сущности изобретения могут быть применимыми к другим стандартам или способам кодирования.

Подробности кодирования палитры, описанного выше в отношении кодера 20, не повторяются здесь в отношении декодера 30, но следует понимать, что декодер 30 может выполнять процесс декодирования, обратный по отношению к любому процессу кодирования, описанному здесь в отношении кодера 20.

Видеodeкодер 30 представляет собой пример устройства, которое может быть выполнено с возможностью реализации технологий для основанного на палитре кодирования и энтропийного кодирования (например, CABAC) согласно различным примерам, описанным в этом раскрытии сущности изобретения.

В примере фиг. 3 видеodeкодер 30 включает в себя блок 150 энтропийного декодирования, память 151 видеоданных, блок 152 декодирования блоков, блок 154 обратного квантования, блок 156 обработки обратных преобразований, блок 158 восстановления, блок 160 фильтрации и буфер 162 декодированных изображений. Блок 152 декодирования блоков включает в себя блок 164 компенсации движения и блок 166 обработки интра-предсказаний. Видеodeкодер 30 также включает в себя блок 165 основанного на палитре декодирования, выполненный с возможностью выполнения различных аспектов технологий основанного на палитре кодирования, описанных в этом раскрытии сущности изобретения. В других примерах видеodeкодер 30 может включать в себя дополнительные функциональные компоненты, меньшее число функциональных компонентов или другие функциональные компоненты.

Память 151 видеоданных может хранить видеоданные, такие как кодированный битовый поток видео, подлежащие декодированию компонентами видеodeкодера 30. Видеоданные, хранимые в памяти 151 видеоданных, могут быть получены, например, из машиночитаемого носителя 16, например из локального источника видео, такого как камера, посредством проводной или беспроводной передачи видеоданных, или посредством осуществления доступа к физическим устройствам хранения данных. Память 151 видеоданных может образовывать буфер кодированных изображений (coded picture buffer - CPB), который хранит кодированные видеоданные из кодированного битового потока видео. Буфер 162 декодированных изображений может быть памятью опорных изображений, которая хранит опорные видеоданные для использования в декодировании видеоданных видеodeкодером 30, например, в режимах интра- или интер-кодирования. Память 151 видеоданных и буфер 162 декодированных изображений могут быть образованы любым из множества запоминающих устройств, таких как динамическая память с произвольным доступом (DRAM), в том числе синхронная DRAM (SDRAM), магниторезистивная RAM (MRAM), резистивная RAM (RRAM) или другие типы запоминающих устройств.

Память 151 видеоданных и буфер 162 декодированных изображений могут быть обеспечены одним и тем же запоминающим устройством или отдельными запоминающими устройствами. В различных примерах память 151 видеоданных может находиться в микросхеме вместе с другими компонентами видеodeкодера 30 или может быть внекристальной относительно этих компонентов.

Буфер кодированных изображений (CPB), который может быть обеспечен памятью 151 видеоданных, может принимать и хранить кодированные видеоданные (например, единицы NAL) битового пото-

ка. Блок 150 энтропийного декодирования может принимать кодированные видеоданные (например, единицы NAL) из CPB и синтаксически разбирать единицы NAL для декодирования синтаксических элементов. Блок 150 энтропийного декодирования может энтропийно декодировать энтропийно кодированные синтаксические элементы в единицах NAL. Блок 152 декодирования блоков, блок 154 обратного квантования, блок 156 обработки обратных преобразований, блок 158 восстановления, и блок 160 фильтрации могут генерировать декодированные видеоданные на основе синтаксических элементов, извлеченных из битового потока.

Видеодекодер 30 может быть выполнен с возможностью выполнения процесса, в общем, обратного по отношению к процессу видеокодера 20, описанному здесь. Подобным образом, видеокодер 20 может быть выполнен с возможностью выполнения процесса, в общем, обратного по отношению к процессу видеодекодера 30, описанному здесь. Например, раскрытие того, что видеодекодер 30 может быть выполнен с возможностью декодирования кодированного синтаксического элемента в битовом потоке подобным образом обязательно раскрывает, что видеокодер 20 может быть выполнен с возможностью кодирования синтаксического элемента в битовом потоке.

В качестве другого примера блок 150 энтропийного декодирования может быть выполнен с возможностью выполнения процесса, в общем, обратного процессу блока 118 энтропийного кодирования, описанного здесь. Согласно аспектам этого раскрытия сущности изобретения блок 150 энтропийного декодирования может быть выполнен с возможностью энтропийного декодирования любых кодовых слов, генерируемых блоком 118 энтропийного кодирования. Например, блок 150 энтропийного декодирования может быть выполнен с возможностью энтропийного декодирования постоянных и непостоянных значений усеченного экспоненциального кода Голомба k-го порядка (kth order truncated Exp-Golomb - TEGk), таких как двоичный вектор предсказания палитры и/или карта палитры для CU. В качестве другого примера блок 150 энтропийного декодирования может быть выполнен с возможностью энтропийного декодирования кодового слова экспоненциального кода Голомба k-го порядка (kth order Exp-Golomb - EGk), кодового слова усеченного экспоненциального кода Голомба k-го порядка (kth order truncated Exp-Golomb - TEGk), непостоянного кодового слова усеченного экспоненциального кода Голомба k-го порядка (TEGk) или любой их комбинации.

Единицы NAL битового потока могут включать в себя единицы NAL кодированных слайсов. В качестве части декодирования битового потока блок 150 энтропийного декодирования может извлекать и энтропийно декодировать синтаксические элементы из единиц NAL кодированных слайсов. 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, связанный с изображением, которое содержит слайс.

Дополнительно к декодированию синтаксических элементов из битового потока видеодекодер 30 может выполнять операцию восстановления в отношении неразбитой CU. Для выполнения операции восстановления в отношении неразбитой CU видеодекодер 30 может выполнить операцию восстановления в отношении каждой TU CU. Посредством выполнения операции восстановления для каждой TU CU, видеодекодер 30 может восстановить остаточные блоки CU.

В качестве части выполнения операции восстановления в отношении TU CU блок 154 обратного квантования может обратно квантовать, т.е. деквантовать, блоки коэффициентов, связанные с TU. Блок 154 обратного квантования может использовать значение QP, связанное с CU TU, для определения степени квантования и, подобным образом, степени обратного квантования, которую должен применять блок 154 обратного квантования. А именно, степенью сжатия, т.е. отношением числа битов, используемых для представления исходной последовательности, к числу битов, используемых для представления сжатой последовательности, можно управлять посредством настройки значения QP, используемого при квантовании коэффициентов преобразования. Степень сжатия может также зависеть от используемого способа энтропийного кодирования.

После квантования блоком 154 обратного квантования блока коэффициентов блок 156 обработки обратных преобразований может применить одно или несколько обратных преобразований к блоку коэффициентов для генерирования остаточного блока, связанного с TU. Например, блок 156 обработки обратных преобразований может применить обратное DCT, обратное целочисленное преобразование, обратное преобразование Карунена-Лозва (Karhunen-Loeve transform - KLT), обратное вращательное преобразование, обратное направленное преобразование или другое обратное преобразование к блоку коэффициентов.

Если PU кодирована с использованием интра-предсказания, то блок 166 обработки интра-предсказаний может выполнить интра-предсказание для генерирования предиктивных блоков для PU. Блок 166 обработки интра-предсказаний может использовать режим интра-предсказания для генерирования предиктивных блоков яркости, Cb и Cr для PU, на основе блоков предсказания пространственно соседних PU. Блок 166 обработки интра-предсказаний может определить режим интра-предсказания для PU на основе одного или нескольких синтаксических элементов, декодированных из битового потока.

Блок 152 декодирования блоков может создавать 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 интер-предсказания, то блок 150 энтропийного декодирования может извлечь информацию о движении для PU. Блок 164 компенсации движения может определить, на основе информации о движении PU, одну или несколько опорных областей для PU. Блок 164 компенсации движения может генерировать, на основе блоков отсчетов в одном или нескольких опорных блоках для PU, предиктивные блоки яркости, Cb и Cr для PU.

Блок 158 восстановления может использовать блоки преобразований яркости, Cb и Cr, связанные с TU CU, и предиктивные блоки яркости, Cb и Cr PU CU, т.е. либо данные интра-предсказания, либо данные интер-предсказания, если это применимо, для восстановления блоков кодирования яркости, Cb и Cr CU. Например, блок 158 восстановления может добавлять отсчеты блоков преобразования яркости, Cb и Cr к соответствующим отсчетам предиктивных блоков яркости, Cb и Cr для восстановления блоков кодирования яркости, Cb и Cr CU.

Блок 160 фильтрации может выполнять операцию деблокирования для уменьшения артефактов блокирования, связанных с блоками кодирования яркости, Cb и Cr CU. Видеодекодер 30 может хранить блоки кодирования яркости, Cb и Cr CU в буфере 162 декодированных изображений. Буфер 162 декодированных изображений может обеспечить опорные изображения для последующей компенсации движения, интра-предсказания и представления на устройстве отображения, таком как устройство 32 отображения фиг. 1. Например, видеодекодер 30 может выполнить на основе блоков яркости, Cb и Cr в буфере 162 декодированных изображений, операции интра-предсказания или интер-предсказания в отношении PU других CU.

Согласно различным примерам этого раскрытия сущности изобретения видеодекодер 30 может быть выполнен с возможностью выполнения основанного на палитре кодирования. Блок 165 основанного на палитре декодирования, например, может выполнить основанное на палитре декодирование, когда выбран режим основанного на палитре декодирования, например, для CU или PU. Например, блок 165 основанного на палитре декодирования может быть выполнен с возможностью генерирования палитры, имеющей записи, указывающие на пиксельные значения, приема информации, связывающей по меньшей мере некоторые пиксельные местоположения в блоке видеоданных с записями в палитре, выбора пиксельных значений в палитре на основе этой информации, и восстановления пиксельных значений блока на основе выбранных пиксельных значений в палитре. Хотя различные функции описаны как выполняемые блоком 165 основанного на палитре декодирования, некоторые или все такие функции могут быть выполнены другими блоками обработки, или комбинацией разных блоков обработки.

Блок 165 основанного на палитре декодирования может принимать информацию режима кодирования палитры и выполнять вышеупомянутые операции, когда информация режима кодирования палитры указывает на то, что режим кодирования палитры применяется к блоку. Когда информация режима кодирования палитры указывает на то, что режим кодирования палитры не применяется к блоку, или когда другая информация о режиме указывает на использование другого режима, блок 165 основанного на палитре декодирования декодирует блок видеоданных с использованием не-основанного на палитре режима кодирования, такого как интер-предиктивный или интра-предиктивный режим кодирования HEVC. Блок видеоданных может быть, например, CU или PU, сгенерированной согласно процессу кодирования HEVC. Режим основанного на палитре кодирования может содержать один или множество разных основанных на палитре режимов кодирования или это может быть единственный основанный на палитре режим кодирования.

Согласно аспектам этого раскрытия сущности изобретения блок 165 основанного на палитре декодирования может быть выполнен с возможностью выполнения любой комбинации технологий для кодирования палитры, описанных здесь. Подробности кодирования палитры, описанные выше в отношении видеокодера 20, не повторяются здесь в отношении видеодекодера 30, но следует понимать, что декодер 30 может выполнять основанный на палитре процесс декодирования, обратный по отношению к любому основанному на палитре процессу кодирования, описанному здесь в отношении кодера 20.

Видеодекодер 30 может быть выполнен с возможностью определения того, что текущий блок кодирован в режиме палитры, декодирования в обходном режиме множества экземпляров первого синтаксического элемента для восстановления текущего блока, например, с использованием обходного режима процесса кодирования САВАС, и, после декодирования в обходном режиме множества экземпляров первого синтаксического элемента, декодирования в контекстном режиме множества примеров второго синтаксического элемента для восстановления текущего блока, например, с использованием процесса кодирования САВАС. Видеодекодер 30 может декодировать в обходном режиме любые два примера из множества экземпляров первого синтаксического элемента без чередования с декодированием в контекстном режиме любого примера из множества примеров второго синтаксического элемента. В некоторых примерах первый синтаксический элемент содержит одно из синтаксического элемента `palette_index_idc` или синтаксического элемента `palette_escape_val` и второй синтаксический элемент содержит синтаксический элемент `palette_run_msb_id_plus1`. Видеодекодер 30 может декодировать в обходном режиме множество экземпляров первого синтаксического элемента в передней части секции кодирования индексных блоков для текущего блока.

Видеодекoder 30 может декодировать третий синтаксический элемент, указывающий число экземпляров первого синтаксического элемента, причем декодирование в обходном режиме множества экземпляров первого синтаксического элемента содержит декодирование в обходном режиме множества экземпляров первого синтаксического элемента на основе третьего синтаксического элемента. Видеодекoder 30 может декодировать третий синтаксический элемент с использованием одного из кода Голомба-Райса, экспоненциального кода Голомба, усеченного кода Райса, унарного кода, или сочленения кода Голомба-Райса и экспоненциального кода Голомба, или усеченной версии любого из предыдущих кодов.

Фиг. 4 является концептуальной схемой, показывающей пример определения палитры для кодирования видеоданных, совместимый с технологиями этого раскрытия сущности изобретения. Пример фиг. 4 включает в себя изображение 178, имеющее первую единицу 180 кодирования (CU) PAL (палитры), которая связана с первыми палитрами 184, и вторую CU 188 PAL, которая связана с вторыми палитрами 192. Как описано более подробно ниже и согласно технологиям этого раскрытия сущности изобретения, вторые палитры 192 основаны на первых палитрах 184. Изображение 178 также включает в себя блок 196, кодированный с использованием режима кодирования с интра-предсказанием, и блок 200, который кодирован с использованием режима кодирования с интер-предсказанием.

Технологии фиг. 4 описаны в контексте видеокодера 20 (фиг. 1 и 2) и видеодекодера 30 (фиг. 1 и 3) и в отношении стандарта кодирования видео HEVC в целях объяснения. Однако следует понимать, что технологии этого раскрытия сущности изобретения этим не ограничены и могут быть применены другими процессорами и/или устройствами кодирования видео в других процессах и/или стандартах кодирования видео.

В общем, палитра относится к числу пиксельных значений, которые являются преобладающими и/или характерными для CU, кодируемой в данный момент, CU 188 в примере фиг. 4. Первые палитры 184 (которые могут также называться индексами 184) и вторые палитры 192 (которые могут также называться индексами 192) показаны как включающие в себя множественные палитры (которые могут также называться множественными индексами). В некоторых примерах согласно аспектам этого раскрытия сущности изобретения видеокодер (такой как видеокодер 20 или видеодекoder 30) может кодировать палитры (например, индексы) отдельно для каждого цветового компонента CU. Например, видеокодер 20 может кодировать палитру для компонента яркости (Y) CU, другую палитру для компонента цветности (U) CU и еще одну палитру для компонента цветности (V) 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.

В других примерах видеокодер 20 может кодировать единственную 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 в виде набора палитр, имеющих множественные отдельные палитры, является просто одним примером и не является ограничением.

В примере фиг. 4 первые палитры 184 включают в себя три записи 202-206, имеющие индексное значение 1 записи, индексное значение 2 записи и индексное значение 3 записи соответственно. Первые палитры 184 связывают индексные значения (например, значения, показанные в левом столбце первых палитр 184) с пиксельными значениями. Например, как показано на фиг. 4, одна из первых палитр 184 связывает индексные значения 1, 2 и 3 с пиксельными значениями A, B и C соответственно. Как описано здесь, вместо кодирования фактических пиксельных значений первой CU 180, видеокодер (такой как видеокодер 20 или видеодекoder 30) может использовать основанное на палитре кодирование для кодирования пикселей блока с использованием индексов 1-3 (которые могут быть также выражены в виде индексных значений 1-3). А именно, для каждого пиксельного положения первой CU 180 видеокодер 20 может кодировать индексное значение для пикселя, причем индексное значение связано с пиксельным значением в одной или нескольких первых палитрах 184. Видеодекoder 30 может получить индексные значения из битового потока и восстановить пиксельные значения с использованием индексных значений и одной или нескольких первых палитр 184. Таким образом, первые палитры 184 передаются видеокодером 20 в кодированном битовом потоке видеоданных для использования видеодекoderом 30 в основном на палитре декодирования.

В некоторых примерах видеокодер 20 и видеодекoder 30 могут определить вторые палитры 192 на основе первых палитр 184. Например, видеокодер 20 и/или видеодекoder 30 могут определить местоположения одного или нескольких блоков, на основании которых определяются предиктивные палитры, в этом примере первые палитры 184. В некоторых примерах, таких как пример, показанный на фиг. 4, видеокодер 20 и/или видеодекoder 30 могут определить местоположение ранее кодированной CU, такой как левая соседняя CU (первая CU 180) при определении предиктивной палитры для второй CU 188.

В примере фиг. 4 вторые палитры 192 включают в себя три записи 208-212, имеющие индексное значение 1 записи, индексное значение 2 записи и индексное значение 3 записи соответственно. Вторые палитры 192 связывают индексные значения (например, значения, показанные в левом столбце первых палитр 192) с пиксельными значениями. Например, как показано на фиг. 4, одна из вторых палитр 192

связывает индексные значения 1, 2 и 3 с пиксельными значениями А, В и D соответственно. В этом примере видеокодер 20 может кодировать один или несколько синтаксических элементов, указывающих на то, какие записи первых палитр 184 включены во вторые палитры 192. В примере фиг. 4 упомянутые один или несколько синтаксических элементов показаны в виде вектора 216. Вектор 216 имеет некоторое число соответствующих бинов (или битов), причем каждый бин указывает на то, используется ли предиктор палитры, связанный с этим бином, для предсказания записи текущей палитры. Например, вектор 216 указывает на то, что первые две записи первых палитр 184 (202 и 204) включены во вторые палитры 192 (значение "1" в векторе 216), в то время как третья запись первых палитр 184 не включена во вторые палитры 192 (значение "0" в векторе 216). В примере фиг. 4 вектор является булевым вектором.

В некоторых примерах видеокодер 20 и декодер 30 могут определить список предикторов палитры (который может также называться таблицей предикторов палитры) при выполнении предсказания палитры. Список предикторов палитры может включать в себя записи из палитр одного или нескольких соседних блоков, которые используются для предсказания одной или нескольких записей палитры для кодирования текущего блока. Видеокодер 20 и декодер 30 могут создавать этот список одним и тем же образом. Видеокодер 20 и декодер 30 могут кодировать данные (такие как вектор 216) для указания на то, какие записи из списка предикторов палитры должны быть включены в палитру для кодирования текущего блока.

Фиг. 5 является концептуальной схемой, показывающей пример определения индексов для палитры для блока пикселей, совместимый с технологиями этого раскрытия сущности изобретения. Например, фиг. 5 включает в себя индексный блок 240 (который может также называться картой 240 или картой 240 индексов), включающий в себя индексные значения (например, индексные значения 1, 2, и 3), которые связывают соответствующие положения пикселей, связанных с индексными значениями, с записью палитры 244.

В то время как индексный блок 240 показан в примере фиг. 5 как включающий в себя индексное значение для каждого пиксельного положения, следует понимать, что в других примерах не все пиксельные положения могут быть связаны с индексным значением, связывающим пиксельное значение с записью палитры 244. А именно, как отмечено выше, в некоторых примерах видеокодер 20 может кодировать (и декодер 30 может получить, из кодированного битового потока) указание на фактическое пиксельное значение (или его квантованную версию) для положения в индексном блоке 240, если пиксельное значение не включено в палитры 244.

В некоторых примерах видеокодер 20 и декодер 30 могут быть выполнены с возможностью кодирования дополнительной карты, указывающей на то, какие пиксельные положения связаны с какими индексными значениями. Например, предположим, что запись (i, j) в индексном блоке 240 соответствует положению (i, j) CU. Видеокодер 20 может кодировать один или несколько синтаксических элементов для каждой записи индексного блока (например, каждого пиксельного положения), указывающих на то, имеет ли запись соответствующее индексное значение. Например, видеокодер 20 может кодировать флаг, имеющий значение, равное одному, для указания на то, что пиксельное значение в местоположении (i, j) в CU является одним из значений в палитрах 244.

Видеокодер 20 может в таком примере также кодировать палитру (показанную в примере фиг. 5 как 244). В примерах, в которых палитры 244 включают в себя единственную запись и соответствующее пиксельное значение, видеокодер 20 может игнорировать сигнализацию этого пиксельного значения. Видеокодер 20 может кодировать, чтобы флаг имел значение, равное нулю, для указания на то, что пиксельное значение в местоположении (i, j) в CU не является одним из значений в палитрах 244. В этом примере видеокодер 20 может также кодировать указание на пиксельное значение для использования декодером 30 в восстановлении пиксельного значения. В некоторых примерах пиксельное значение может быть кодировано с потерями.

Значение пикселя в одном положении CU может обеспечить указание на значения одного или нескольких других пикселей в других положениях CU. Например, может существовать относительно высокая вероятность того, что соседние пиксельные положения CU будут иметь то же самое пиксельное значение или могут быть отображены в то же самое индексное значение (в случае кодирования с потерями, в котором более чем одно пиксельное значение может быть отображено в единственное индексное значение).

Соответственно видеокодер 20 может кодировать один или несколько синтаксических элементов, указывающих число последовательных пикселей или индексных значений, при данном порядке сканирования, которые имеют одно и то же пиксельное значение или индексное значение. Как отмечено выше, последовательность подобных пиксельных или индексных значений может называться здесь серией. В примере в целях иллюстрации, если два последовательных пикселя или индекса при данном порядке сканирования имеют разные значения, то серия равна нулю. Если два последовательных пикселя или индекса при данном порядке сканирования имеют одно и то же значение, но третий пиксел или индекс при данном порядке сканирования имеет другое значение, то серия равна одному. Для трех последовательных индексов или пикселей с одним и тем же значением серия равна двум и т.д. Видеокодер 20 может получить синтаксические элементы, указывающие на серию, из кодированного битового потока, и

использовать эти данные для определения числа последовательных местоположений, которые имеют одно и то же пиксельное или индексное значение.

В некоторых примерах согласно технологиям этого раскрытия сущности изобретения блок 118 энтропийного кодирования и блок 150 энтропийного декодирования могут быть выполнены с возможностью энтропийного кодирования индексного блока 240. Например, блок 118 энтропийного кодирования и блок 150 энтропийного декодирования могут быть выполнены с возможностью энтропийного кодирования длин серий (например, значений или кодов длин серий) и/или двоичного вектора предсказания палитры, относящегося к индексному блоку в режиме палитры.

Фиг. 6 является концептуальной схемой, показывающей пример определения максимальной длины серии в режиме копирования сверху, предполагая пример растрового порядка сканирования, совместимый с технологиями этого раскрытия сущности изобретения. В примере фиг. 6, если ни один из пикселей, окруженных пунктирными линиями 280, не кодирован как переходной пиксел, то максимально возможная длина серии равна 35 (т.е. числу незатененных пиксельных положений). Если один или несколько пикселей в пределах пунктирных линий 280 кодированы как переходные отсчеты, предполагая, что пиксел, обозначенный как переходной пиксел (пиксельное положение с "X"), является первым переходным пикселем в пределах пунктирных линий 280 в порядке сканирования, то тогда максимально возможная длина серии при кодировании с копированием сверху равна пяти.

В некоторых примерах видеodeкодер 30 может определить режим серии (например, режим палитры, в котором кодированы пикселы) только для пикселей в пределах пунктирных линий 280. Следовательно, в наихудшем случае видеodeкодер 30 выполняет определение для BlockWidth (ШиринаБлока)-1 пикселей. В некоторых примерах видеodeкодер 30 может быть выполнен с возможностью реализации некоторых ограничений в отношении максимального числа пикселей, для которых проверяется режим серии. Например, видеodeкодер 30 может проверять только пикселы в пределах пунктирных линий 280, если пикселы находятся в той же самой строке, что и текущий пиксел. Видеodeкодер 30 может сделать вывод о том, что другие пикселы в пределах пунктирных линий 280 не кодированы как переходные отсчеты. Пример фиг. 6 предполагает растровый порядок сканирования. Однако эти технологии могут быть применены к другим порядкам сканирования, таким как вертикальное, горизонтальное перемещение, и вертикальное перемещение.

Согласно примеру этого раскрытия сущности изобретения, если текущим режимом серии является "копировать сверху", то контексты длины серии для текущего пикселя могут зависеть от индексного значения индекса верхнего соседнего пикселя относительно текущего пикселя. В этом примере, если верхний соседний пиксел относительно текущего пикселя находится за пределами текущей CU, то видеodeкодер предполагает, что соответствующий индекс равен предопределенной постоянной k. В некоторых примерах $k=0$.

Во время энтропийного кодирования энтропийный кодер или декодер может помещать биты символа, подлежащего кодированию, в один или несколько бинов. Бины могут указывать, равно ли значение символа нулю. Энтропийный кодер или энтропийный декодер может использовать значения бинов для настройки процесса энтропийного кодирования. В некоторых примерах энтропийный кодер или декодер может также использовать бины для указания на то, являются ли значения большими конкретного значения, например большими нуля, большими одного и т.д.

В некоторых примерах, если текущим режимом является "копировать сверху", то первый бин кодового слова длины серии выбирает один из двух CABAC-контекстов-кандидатов на основе того, равен 0 или нет верхний соседний отсчет (например, пиксел) относительно текущего отсчета (например, пикселя).

В качестве другого примера, если текущим режимом является "копировать предыдущее", то первый бин кодового слова длины серии выбирает один из четырех CABAC-контекстов-кандидатов на основе того, равно ли индексное значение 0, 1, 2 или оно больше чем 2.

Фиг. 8 является блок-схемой последовательности операций, показывающей иллюстративный процесс для декодирования видеоданных, совместимый с технологиями этого раскрытия сущности изобретения. Процесс фиг. 8, в общем, описан как выполняемый видеodeкодером 30 в целях иллюстрации, хотя множество других процессоров может также выполнить процесс, показанный на фиг. 8. В некоторых примерах блок 152 декодирования блоков, блок 165 основанного на палитре декодирования и/или блок 150 энтропийного декодирования могут выполнить один или несколько процессов, показанных на фиг. 8.

В примере фиг. 8 видеodeкодер 30 может быть выполнен с возможностью приема (800) из кодированного битового потока видео, кодированного в режиме палитры блока видеоданных изображения. Видеodeкодер 30 может быть выполнен с возможностью приема (802) из кодированного битового потока видео кодированной информации режима палитры для кодированного в режиме палитры блока видеоданных. В некоторых примерах кодированная информация режима палитры может включать в себя множество экземпляров первого синтаксического элемента и множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента. Например, первый синтаксический элемент может включать в себя `palette_index_idc` или `palette_escape_val`, а множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, может включать в себя

синтаксический элемент `palette_run_msb_id_plus1`. В качестве другого примера первый синтаксический элемент может быть указанием индекса массива записей палитры или первый синтаксический элемент может задавать квантованное переходное кодированное значение отсчета для цветового компонента, соответствующего переходному отсчету. Множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, может включать в себя синтаксический элемент, который задает индекс наиболее значимого бита в двоичном представлении переменной, представляющей длину серии, и синтаксический элемент, который задает режим типа серии.

В качестве другого примера множественные синтаксические элементы, которые отличаются от упомянутого первого синтаксического элемента, могут быть любыми и всеми синтаксическими элементами, которые отличаются от упомянутого первого синтаксического элемента. Как описано здесь в отношении некоторых примеров, множественные синтаксические элементы, которые отличаются от упомянутого первого синтаксического элемента, могут быть также отличными от второго, третьего и/или четвертого синтаксических элементов. В таких примерах множественные синтаксические элементы, которые являются отличными от первого, второго, третьего и четвертого синтаксических элементов, могут быть любыми и всеми синтаксическими элементами, которые являются отличными от первого, второго, третьего и/или четвертого синтаксических элементов. В некоторых примерах множественные синтаксические элементы, которые отличаются от упомянутого первого синтаксического элемента, могут быть любыми и всеми синтаксическими элементами, которые декодируются не в обходном режиме и/или которые не подлежат декодированию в обходном режиме.

Видеодекодер 30 может быть выполнен с возможностью декодирования (804) с использованием обходного режима, например обходного режима процесса кодирования САВАС, множества экземпляров первого синтаксического элемента перед декодированием множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, с использованием контекстного режима. Видеодекодер 30 может быть выполнен с возможностью декодирования (806) с использованием контекстного режима, например обычного режима САВАС (а не обходного режима), множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, после декодирования множества экземпляров первого синтаксического элемента с использованием обходного режима. В некоторых примерах множество экземпляров первого синтаксического элемента включает в себя все экземпляры первого синтаксического элемента для кодированного в режиме палитры блока видеоданных. В таких примерах все экземпляры первого синтаксического элемента декодируют с использованием обходного режима перед декодированием любых последующих данных, таких как множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента. Другими словами, видеодекодер 30 может быть выполнен с возможностью декодирования с использованием контекстного режима, множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, после декодирования всех экземпляров первого синтаксического элемента для кодированного в режиме палитры блока видеоданных с использованием обходного режима.

Видеодекодер 30 может быть выполнен с возможностью декодирования (808) кодированного в режиме палитры блока видеоданных с использованием декодированного множества экземпляров первого синтаксического элемента и декодированного множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента. В некоторых примерах множество экземпляров первого синтаксического элемента группируют вместе таким образом, чтобы уменьшалось число переключений между обходным режимом и контекстным режимом, при декодировании кодированного в режиме палитры блока видеоданных.

В некоторых примерах кодированная информация режима палитры может включать в себя второй синтаксический элемент, указывающий число экземпляров первого синтаксического элемента (например, указывающий на то, сколько экземпляров первого синтаксического элемента существует для кодированного в режиме палитры блока видеоданных). Множественные синтаксические элементы, которые отличаются от упомянутого первого синтаксического элемента, могут быть также отличными от второго синтаксического элемента. В таких примерах видеодекодер 30 может быть выполнен с возможностью декодирования с использованием обходного режима, второго синтаксического элемента перед декодированием множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента. В некоторых примерах никакой экземпляр второго синтаксического элемента не размещается между какими-либо двумя экземплярами первого синтаксического элемента для кодированного в режиме палитры блока видеоданных. В некоторых примерах видеодекодер 30 может быть выполнен с возможностью определения после декодирования числа экземпляров первого синтаксического элемента, равного числу, указанному вторым синтаксическим элементом, того, что последующие данные в кодированном битовом потоке видео, следующие за числом экземпляров первого синтаксического элемента, соответствуют множеству синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента. В некоторых примерах видеодекодер 30 может быть выполнен с возможностью декодирования второго синтаксического элемента с использованием сочленения усеченного кода Райса и экспоненциального кода Голмба.

В некоторых примерах кодированная информация режима палитры может включать в себя третий синтаксический элемент и четвертый синтаксический элемент. В таких примерах видеодекoder 30 может быть выполнен с возможностью декодирования третьего синтаксического элемента для определения значения, соответствующего третьему синтаксическому элементу, указывающего на то, включает ли кодированный в режиме палитры блок видеоданных в себя переходной пиксел. Видеодекoder 30 может быть выполнен с возможностью декодирования четвертого синтаксического элемента для определения значения, соответствующего четвертому синтаксическому элементу, указывающего на размер палитры. Видеодекoder 30 может быть выполнен с возможностью декодирования на основе определенных значений, соответственно соответствующих третьему и четвертому синтаксическим элементам, множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента, с использованием контекстного режима, после декодирования множества экземпляров первого синтаксического элемента и второго синтаксического элемента с использованием обходного режима.

В некоторых примерах кодированная информация режима палитры может включать в себя другой синтаксический элемент и видеодекoder 30 может быть выполнен с возможностью декодирования этого другого синтаксического элемента для определения значения, соответствующего этому другому синтаксическому элементу, которое задает число отдельных значений, которые имеет индекс палитры, для кодированного в режиме палитры блока видеоданных. Видеодекoder 30 может быть выполнен с возможностью декодирования на основе определенного значения, соответствующего этому другому синтаксическому элементу, множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента, с использованием контекстного режима, после декодирования множества экземпляров первого синтаксического элемента и второго синтаксического элемента с использованием обходного режима.

В некоторых примерах кодированная информация режима палитры может включать в себя другой синтаксический элемент и видеодекoder 30 может быть выполнен с возможностью декодирования этого другого синтаксического элемента для определения значения, соответствующего этому другому синтаксическому элементу, указывающего на последний экземпляр синтаксического элемента `palette_run_type_flag[xC][yC]` для кодированного в режиме палитры блока видеоданных.

В некоторых примерах видеодекoder 30 может быть выполнен с возможностью определения того, что кодированный блок видеоданных имеет один или несколько переходных отсчетов. В таких примерах видеодекoder 30 может быть выполнен с возможностью декодирования последнего переходного отсчета в кодированном блоке видеоданных среди одного или нескольких переходных отсчетов. Видеодекoder 30 может быть выполнен с возможностью получения индексного значения, которое применяется к отсчетам кодированного блока видеоданных, следующим за последним переходным отсчетом. Видеодекoder 30 может быть выполнен с возможностью декодирования отсчетов кодированного блока видеоданных, следующих за последним переходным отсчетом, с использованием полученного индексного значения для каждого отсчета из отсчетов, следующих за последним переходным отсчетом.

В некоторых примерах видеодекoder 30 может быть выполнен с возможностью определения числа принятых индексов палитры. В таких примерах видеодекoder 30 может быть выполнен с возможностью определения числа оставленных индексов палитры на основе числа принятых индексов палитры и числа экземпляров первого синтаксического элемента. Видеодекoder 30 может быть выполнен с возможностью определения максимально возможного значения серии для кодированного блока видеоданных на основе числа принятых индексов палитры и числа экземпляров первого синтаксического элемента. В некоторых примерах видеодекoder 30 может быть выполнен с возможностью определения максимально возможного значения серии для кодированного блока видеоданных согласно $nCbS * nCbS - scanPos - 1 - paletteIndicesLeft$, где $nCbS$ задает размер кодированного блока видеоданных, $scanPos$ задает положение сканирования и $paletteIndicesLeft$ задает число оставленных индексов палитры.

Фиг. 9 является блок-схемой последовательности операций, показывающей иллюстративный процесс для кодирования видеоданных, совместимый с технологиями этого раскрытия сущности изобретения. Процесс фиг. 9, в общем, описан как выполняемый видеокодером 20 в целях иллюстрации, хотя множество других процессоров может также выполнить процесс, показанный на фиг. 9. В некоторых примерах блок 100 кодирования блоков, блок 122 основанного на палитре кодирования и/или блок 118 энтропийного кодирования могут выполнить один или несколько процессов, показанных на фиг. 9.

В примере фиг. 9 видеокодер 20 может быть выполнен с возможностью определения (900) того, что блок видеоданных подлежит кодированию в режиме палитры. Видеокодер 20 может быть выполнен с возможностью кодирования (902) блока видеоданных с использованием режима палитры в кодированный битовый поток. В некоторых примерах видеокодер 20 может быть выполнен с возможностью генерирования (904) информации режима палитры для блока видеоданных. Информация режима палитры может включать в себя множество экземпляров первого синтаксического элемента и множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента. Например, первый синтаксический элемент может включать в себя `palette_index_idc` или `palette_escape_val`, а множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического эле-

мента, может включать в себя синтаксический элемент `palette_run_msb_id_plus1`. В качестве другого примера первый синтаксический элемент может быть указанием индекса массива записей палитры или первый синтаксический элемент может задавать квантованное переходное кодированное значение отсчета для цветового компонента, соответствующего переходному отсчету. Множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, может включать в себя синтаксический элемент, который задает индекс наиболее значимого бита в двоичном представлении переменной, представляющей длину серии, и синтаксический элемент, который задает режим типа серии.

В качестве другого примера множественные синтаксические элементы, которые отличаются от упомянутого первого синтаксического элемента, могут быть любыми и всеми синтаксическими элементами, которые отличаются от упомянутого первого синтаксического элемента. Как описано здесь в отношении некоторых примеров, множественные синтаксические элементы, которые отличаются от упомянутого первого синтаксического элемента, могут быть также отличными от второго, третьего и/или четвертого синтаксических элементов. В таких примерах множественные синтаксические элементы, которые являются отличными от первого, второго, третьего и/или четвертого синтаксических элементов, могут быть любыми и всеми синтаксическими элементами, которые являются отличными от первого, второго, третьего и/или четвертого синтаксических элементов. В некоторых примерах множественные синтаксические элементы, которые отличаются от упомянутого первого синтаксического элемента, могут быть любыми и всеми синтаксическими элементами, которые не являются кодированными в обходном режиме и/или которые не подлежат кодированию в обходном режиме.

Видеокодер 20 может быть выполнен с возможностью кодирования (906) с использованием обходного режима, например обходного режима процесса кодирования САВАС, множества экземпляров первого синтаксического элемента в кодированный битовый поток, перед кодированием множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, в кодированный битовый поток с использованием контекстного режима. Видеокодер 20 может быть выполнен с возможностью кодирования (908) с использованием контекстного режима, например обычного контекстного режима САВАС, множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, в кодированный битовый поток, после кодирования множества экземпляров первого синтаксического элемента с использованием обходного режима в кодированный битовый поток. В некоторых примерах множество экземпляров первого синтаксического элемента группируют вместе таким образом, чтобы уменьшалось число переключений между обходным режимом и контекстным режимом, при кодировании кодированного в режиме палитры блока видеоданных.

В некоторых примерах множество экземпляров первого синтаксического элемента включает в себя все экземпляры первого синтаксического элемента для блока видеоданных. В таких примерах все экземпляры первого синтаксического элемента кодируют с использованием обходного режима перед кодированием любых последующих данных, таких как множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента. Другими словами, видеокодер 20 может быть выполнен с возможностью кодирования, с использованием контекстного режима, множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, после кодирования всех экземпляров первого синтаксического элемента для блока видеоданных с использованием обходного режима.

В некоторых примерах информация режима палитры может включать в себя второй синтаксический элемент, указывающий число экземпляров первого синтаксического элемента (например, указывающий на то, сколько экземпляров первого синтаксического элемента существует для блока видеоданных). Множественные синтаксические элементы, которые отличаются от упомянутого первого синтаксического элемента, могут быть также отличными от второго синтаксического элемента. В таких примерах видеокодер 20 может быть выполнен с возможностью кодирования, с использованием обходного режима, второго синтаксического элемента в кодированный битовый поток перед кодированием множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента. В некоторых примерах видеокодер 20 может быть выполнен с возможностью кодирования множества экземпляров первого синтаксического элемента таким образом, чтобы никакой экземпляр второго синтаксического элемента не размещался между какими-либо двумя экземплярами первого синтаксического элемента для кодированного в режиме палитры блока видеоданных в кодированном битовом потоке. В некоторых примерах видеокодер 20 может быть выполнен с возможностью кодирования второго синтаксического элемента в кодированный битовый поток после кодирования множества экземпляров первого синтаксического элемента в кодированный битовый поток. Например, видеокодер 20 может быть выполнен с возможностью кодирования сначала всех экземпляров первого синтаксического элемента и затем кодирования второго синтаксического элемента в кодированный битовый поток. В некоторых примерах видеокодер 20 может быть выполнен с возможностью кодирования второго синтаксического элемента с использованием сочленения усеченного кода Райса и экспоненциального кода Големба.

В некоторых примерах информация режима палитры может включать в себя третий синтаксический

элемент и четвертый синтаксический элемент. В таких примерах видеокодер 20 может быть выполнен с возможностью кодирования значения, соответствующего третьему синтаксическому элементу, указывающего на то, включает ли блок видеоданных в себя переходной пиксел, в кодированный битовый поток. Видеокодер 20 может быть выполнен с возможностью кодирования значения, соответствующего четвертому синтаксическому элементу, указывающего на размер палитры, в кодированный битовый поток. В некоторых примерах информация режима палитры может включать в себя другой синтаксический элемент и видеокодер 20 может быть выполнен с возможностью кодирования значения, соответствующего этому другому синтаксическому элементу, которое задает число отдельных значений, которые имеет индекс палитры, для блока видеоданных, в кодированный битовый поток.

В некоторых примерах кодированная информация режима палитры может включать в себя другой синтаксический элемент и видеокодер 20 может быть выполнен с возможностью кодирования значения, соответствующего этому другому синтаксическому элементу, указывающего на последний экземпляр синтаксического элемента `palette_run_type_flag [xC][yC]` для блока видеоданных.

В некоторых примерах видеокодер 20 может быть выполнен с возможностью кодирования последнего переходного отсчета в блоке видеоданных среди одного или нескольких переходных отсчетов. В таких примерах видеокодер 20 может быть выполнен с возможностью получения индексного значения, которое применяется к отсчетам блока видеоданных, следующим за последним переходным отсчетом. Видеокодер 20 может быть выполнен с возможностью кодирования отсчетов блока видеоданных, следующих за последним переходным отсчетом, с использованием полученного индексного значения для каждого отсчета из отсчетов, следующих за последним переходным отсчетом.

Следует понимать, что все технологии, описанные здесь, могут быть использованы отдельно или в комбинации. Например, видеокодер 20 и/или один или несколько его компонентов и видеокодер 30 и/или один или несколько его компонентов могут выполнять технологии, описанные в этом раскрытии сущности изобретения, в любой комбинации.

Следует понимать, что в зависимости от примера некоторые действия или события любой из технологий, описанных здесь, могут быть выполнены в другой последовательности, могут быть добавлены, объединены или совсем исключены (например, не все описанные действия или события необходимы для применения на практике данных технологий). Кроме того, в некоторых примерах действия или события могут быть выполнены одновременно, например, посредством многопоточной обработки, обработки прерываний или множественных процессоров, а не последовательно. Дополнительно, в то время как некоторые аспекты этого раскрытия сущности изобретения описаны как выполняемые единственным модулем или блоком для ясности, следует понимать, что технологии этого раскрытия сущности изобретения могут быть выполнены комбинацией блоков или модулей, связанных с видеокодером.

Некоторые аспекты этого раскрытия сущности изобретения были описаны в отношении разрабатываемого стандарта HEVC в целях иллюстрации. Однако технологии, описанные в этом раскрытии сущности изобретения, могут быть полезными для других процессов кодирования видео, включающих в себя другой стандарт, или патентуемых процессов кодирования видео, которые еще не разработаны.

Технологии, описанные выше, могут быть выполнены видеокодером 20 (фиг. 1 и 2) и/или видеокодером 30 (фиг. 1 и 3), которые оба могут, в общем, называться видеокодером. Подобным образом, кодирование видео может относиться к кодированию видео или декодированию видео, если это применимо.

Согласно этому раскрытию сущности изобретения термин "или" может быть интерпретирован как "и/или", если контекст не указывает на иное. Дополнительно, фразы, такие как "один или несколько" или "по меньшей мере один" и т.п. могли использоваться для некоторых признаков, раскрытых здесь, но не для других признаков; признаки, для которых такой язык не использовался, могут быть интерпретированы как имеющие такое подразумеваемое значение, если контекст не указывает на иное.

В то время как выше описаны конкретные комбинации различных аспектов технологий, эти комбинации обеспечены только для иллюстрации примеров технологий, описанных в этом раскрытии сущности изобретения. Соответственно технологии этого раскрытия сущности изобретения не должны ограничиваться этими комбинациями примеров и могут охватывать любую постижимую комбинацию различных аспектов технологий, описанных в этом раскрытии сущности изобретения.

В одном или нескольких примерах описанные функции могут быть реализованы в аппаратном средстве, программном средстве, аппаратно-программном средстве или любой их комбинации. При реализации в программном средстве функции могут храниться или передаваться в виде одной или нескольких команд или кода на машиночитаемом носителе и исполняться аппаратным блоком обработки. Машиночитаемые носители могут включать в себя машиночитаемые носители данных, которые соответствуют материальным носителям, таким как носители данных, или средам передачи данных, включающим в себя любой носитель, который обеспечивает передачу компьютерной программы из одного места в другое место, например, согласно протоколу связи. Таким образом, машиночитаемые носители, в общем, могут соответствовать (1) материальным машиночитаемым носителям данных, которые являются долговременными, или (2) среде передачи данных, такой как сигнал или несущая. Носители данных могут быть любыми доступными носителями, к которым может быть осуществлен доступ одним или несколь-

кими компьютерами или одним или несколькими процессорами для извлечения команд, кода и/или структур данных для реализации технологий, описанных в этом раскрытии сущности изобретения. Компьютерный программный продукт может включать в себя машиночитаемый носитель.

В качестве примера, а не ограничения, такие машиночитаемые носители данных могут содержать RAM, ROM, EEPROM, CD-ROM или другое оптическое дисковое запоминающее устройство, магнитное дисковое запоминающее устройство или другие магнитные запоминающие устройства, флэш-память или любой другой носитель, который может быть использован для хранения необходимого программного кода в форме команд или структур данных и к которому может быть осуществлен доступ посредством компьютера. Также любое соединение, собственно, называют машиночитаемым носителем. Например, если команды передаются с веб-сайта, сервера или другого удаленного источника с использованием коаксиального кабеля, волоконно-оптического кабеля, витой пары, цифровой абонентской линии (digital subscriber line - DSL) или беспроводных технологий, таких как инфракрасная, радио- и СВЧ-технология, то тогда коаксиальный кабель, волоконно-оптический кабель, витую пару, цифровую абонентскую линию (DSL) или беспроводные технологии, такие как инфракрасная, радио- и СВЧ-технология, включают в определения носителя. Следует понимать, однако, что машиночитаемые носители данных и носители данных не включают в себя соединения, несущие, сигналы или другие переходные носители, а вместо этого относятся к долговременным, материальным носителям данных. Диск, при использовании здесь, включает в себя компакт-диск (compact disc - CD), лазерный диск, оптический диск, универсальный цифровой диск (digital versatile disc - DVD), гибкий диск или диск Blu-ray, причем диски обычно воспроизводят данные магнитным способом или оптическим способом с использованием лазеров. Комбинации вышеупомянутого также должны быть включены в пределы объема машиночитаемых носителей.

Команды могут быть выполнены одним или несколькими процессорами, такими как один или несколько цифровых сигнальных процессоров (DSP), микропроцессоров общего назначения, специализированных интегральных схем (ASIC), матриц программируемых логических вентилей (FPGA) или других эквивалентных интегральных или дискретных логических схем. Соответственно термин "процессор", при использовании здесь, может относиться к любой из вышеупомянутых структур или любой другой структуре, подходящей для реализации технологий, описанных здесь. Дополнительно, в некоторых аспектах функциональность, описанная здесь, может быть обеспечена в пределах специальных аппаратных и/или программных модулей, выполненных с возможностью кодирования и декодирования, или может быть включена в объединенный кодек. Также данные технологии могут быть полностью реализованы в одной или нескольких схемах или логических элементах.

Технологии этого раскрытия сущности изобретения могут быть реализованы в широком множестве устройств или аппаратов, включающем в себя беспроводной телефон, интегральную схему (IC) или набор IC (например, комплект интегральных схем). Различные компоненты, модули или блоки описаны в этом раскрытии сущности изобретения, чтобы подчеркнуть функциональные аспекты устройств, выполненных с возможностью реализации описанных технологий, но не обязательно требующие реализации разными аппаратными блоками. Напротив, как описано выше, различные блоки могут быть объединены в аппаратном блоке кодека или обеспечены набором взаимодействующих аппаратных блоков, включающих в себя один или несколько процессоров, описанных выше, в сочетании с подходящим программным и/или аппаратно-программным средством.

Здесь были описаны различные примеры. Предполагается любая комбинация описанных систем, операций, функций или примеров. Эти и другие примеры находятся в пределах объема нижеследующей формулы изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ декодирования видеоданных с использованием основанного на палитре режима декодирования, причем способ содержит

прием, из кодированного битового потока видео, кодированного в режиме палитры блока видеоданных изображения;

прием, из кодированного битового потока видео, кодированной информации режима палитры для кодированного в режиме палитры блока видеоданных, причем кодированная информация режима палитры включает в себя множество экземпляров первого синтаксического элемента и множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, при этом первый синтаксический элемент является указанием индекса массива записей палитры или задает квантованное переходное кодированное значение отсчета для цветового компонента, соответствующего переходному отсчету, в котором множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, включает в себя синтаксический элемент, который задает индекс старшего значащего бита в двоичном представлении переменной, представляющей длину серии, и синтаксический элемент, который задает режим типа серии;

синтаксический разбор множества экземпляров первого синтаксического элемента из кодированного битового потока видео перед синтаксическим разбором множества синтаксических элементов, кото-

рые отличаются от упомянутого первого синтаксического элемента, из кодированного битового потока видео;

синтаксический разбор множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, из кодированного битового потока видео после синтаксического разбора множества экземпляров первого синтаксического элемента из кодированного битового потока видео;

декодирование с использованием обходного режима контекстно-адаптивного двоичного арифметического кодирования множества экземпляров первого синтаксического элемента;

декодирование с использованием контекстного режима контекстно-адаптивного двоичного арифметического кодирования множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента; и

декодирование кодированного в режиме палитры блока видеоданных с использованием декодированного множества экземпляров первого синтаксического элемента и декодированного множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента.

2. Способ по п.1, в котором множество экземпляров первого синтаксического элемента сгруппировано вместе таким образом, чтобы уменьшалось число переключений между обходным режимом контекстно-адаптивного двоичного арифметического кодирования и контекстным режимом контекстно-адаптивного двоичного арифметического кодирования при декодировании кодированного в режиме палитры блока видеоданных.

3. Способ по п.1, в котором кодированная информация режима палитры включает в себя второй синтаксический элемент, указывающий число экземпляров первого синтаксического элемента, причем множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, отличаются от второго синтаксического элемента, и причем способ дополнительно содержит

синтаксический разбор второго синтаксического элемента из кодированного битового потока видео перед синтаксическим разбором множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента, из кодированного битового потока видео;

декодирование с использованием обходного режима контекстно-адаптивного двоичного арифметического кодирования второго синтаксического элемента.

4. Способ по п.3, в котором никакой экземпляр второго синтаксического элемента не размещается между какими-либо двумя экземплярами первого синтаксического элемента для кодированного в режиме палитры блока видеоданных.

5. Способ по п.3, дополнительно содержащий определение после декодирования числа экземпляров первого синтаксического элемента, равного числу, указанному вторым синтаксическим элементом, того, что последующие данные в кодированном битовом потоке видео, следующие за этим числом экземпляров первого синтаксического элемента, соответствуют множеству синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента.

6. Способ по п.3, в котором кодированная информация режима палитры включает в себя третий синтаксический элемент и четвертый синтаксический элемент, причем способ дополнительно содержит

декодирование третьего синтаксического элемента для определения значения, соответствующего третьему синтаксическому элементу, указывающего на то, включает ли кодированный в режиме палитры блок видеоданных в себя переходной отсчет;

декодирование четвертого синтаксического элемента для определения значения, соответствующего четвертому синтаксическому элементу, указывающего на размер палитры; и

декодирование на основе определенных значений, соответственно соответствующих третьему и четвертому синтаксическим элементам, множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента, с использованием контекстного режима контекстно-адаптивного двоичного арифметического кодирования.

7. Способ по п.3, в котором кодированная информация режима палитры включает в себя третий синтаксический элемент, причем способ дополнительно содержит

декодирование третьего синтаксического элемента для определения значения, соответствующего третьему синтаксическому элементу, которое задает число отдельных значений, которые имеет индекс палитры, для кодированного в режиме палитры блока видеоданных; и

декодирование на основе определенного значения, соответствующего третьему синтаксическому элементу, множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента и второго синтаксического элемента, с использованием контекстного режима контекстно-адаптивного двоичного арифметического кодирования.

8. Способ по п.3, в котором кодированная информация режима палитры включает в себя третий синтаксический элемент, причем способ дополнительно содержит

декодирование третьего синтаксического элемента для определения значения, соответствующего третьему синтаксическому элементу, указывающего на последний экземпляр синтаксического элемента `palette_run_type_flag[xC][yC]` для кодированного в режиме палитры блока видеоданных.

9. Устройство для декодирования видеоданных с использованием основанного на палитре режима декодирования, причем устройство содержит

память, выполненную с возможностью хранения видеоданных; и
 видеodeкодер, имеющий связь с памятью, причем видеodeкодер выполнен с возможностью приема кодированного в режиме палитры блока видеоданных изображения из памяти;

приема кодированной информации режима палитры для кодированного в режиме палитры блока видеоданных, причем кодированная информация режима палитры включает в себя множество экземпляров первого синтаксического элемента и множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, при этом первый синтаксический элемент является указанием индекса массива записей палитры или задает квантованное переходное кодированное значение отсчета для цветового компонента, соответствующего переходному отсчету, и в котором множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, включает в себя синтаксический элемент, который задает индекс старшего значащего бита в двоичном представлении переменной, представляющей длину серии, и синтаксический элемент, который задает режим типа серии;

синтаксического разбора множества экземпляров первого синтаксического элемента из кодированной информации режима палитры перед синтаксическим разбором множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, из кодированной информации режима палитры;

синтаксического разбора множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, из кодированной информации режима палитры после синтаксического разбора множества экземпляров первого синтаксического элемента из кодированной информации режима палитры;

декодирования с использованием обходного режима контекстно-адаптивного двоичного арифметического кодирования множества экземпляров первого синтаксического элемента;

декодирования с использованием контекстного режима контекстно-адаптивного двоичного арифметического кодирования множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента; и

декодирования кодированного в режиме палитры блока видеоданных с использованием декодированного множества экземпляров первого синтаксического элемента и декодированного множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента.

10. Способ кодирования видеоданных с использованием основанного на палитре режима кодирования, причем способ содержит

определение того, что блок видеоданных подлежит кодированию в режиме палитры;

кодирование блока видеоданных с использованием режима палитры в кодированный битовый поток, причем кодирование блока видеоданных с использованием режима палитры содержит

генерирование информации режима палитры для блока видеоданных, причем информация режима палитры включает в себя множество экземпляров первого синтаксического элемента и множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, при этом первый синтаксический элемент является указанием индекса массива записей палитры или задает квантованное переходное кодированное значение отсчета для цветового компонента, соответствующего переходному отсчету, и в котором множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, включает в себя синтаксический элемент, который задает индекс старшего значащего бита в двоичном представлении переменной, представляющей длину серии, и синтаксический элемент, который задает режим типа серии;

кодирование с использованием обходного режима контекстно-адаптивного двоичного арифметического кодирования множества экземпляров первого синтаксического элемента в кодированный битовый поток перед кодированием множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, в кодированный битовый поток с использованием контекстного режима контекстно-адаптивного двоичного арифметического кодирования; и

кодирование с использованием контекстного режима контекстно-адаптивного двоичного арифметического кодирования множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, в кодированный битовый поток, после кодирования множества экземпляров первого синтаксического элемента с использованием обходного режима контекстно-адаптивного двоичного арифметического кодирования в кодированный битовый поток.

11. Способ по п.1 или 10, в котором множество экземпляров первого синтаксического элемента включает в себя все экземпляры первого синтаксического элемента для блока видеоданных.

12. Способ по п.1 или 10, в котором первый синтаксический элемент является `palette_index_idc` или `palette_escape_val` и в котором множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, включает в себя синтаксический элемент `palette_run_msb_id_plus1`.

13. Способ по п.10, в котором множество экземпляров первого синтаксического элемента сгруппировано вместе таким образом, чтобы уменьшалось число переключений между обходным режимом кон-

текстно-адаптивного двоичного арифметического кодирования и контекстным режимом контекстно-адаптивного двоичного арифметического кодирования при кодировании кодированного в режиме палитры блока видеоданных.

14. Устройство для кодирования видеоданных с использованием основанного на палитре режима кодирования, причем устройство содержит

память, выполненную с возможностью хранения видеоданных; и

видеокодер, имеющий связь с памятью, причем видеокодер выполнен с возможностью

определения того, что блок видеоданных, хранимый в памяти, подлежит кодированию в режиме палитры;

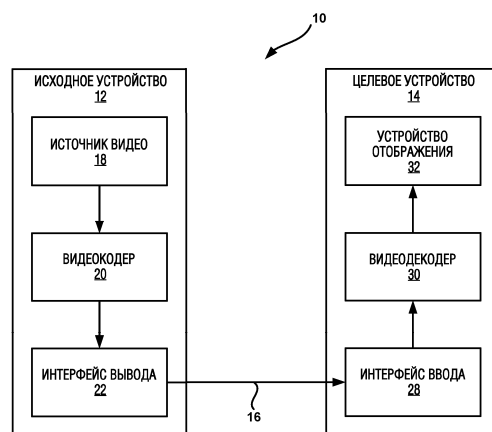
кодирования блока видеоданных с использованием режима палитры в кодированный битовый поток, причем то, что видеокодер выполнен с возможностью кодирования блока видеоданных с использованием режима палитры, содержит то, что видеокодер выполнен с возможностью

генерирования информации режима палитры для блока видеоданных, причем информация режима палитры включает в себя множество экземпляров первого синтаксического элемента и множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, при этом первый синтаксический элемент является указанием индекса массива записей палитры или задает квантованное переходное кодированное значение отсчета для цветового компонента, соответствующего переходному отсчету, и в котором множество синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, включает в себя синтаксический элемент, который задает индекс старшего значащего бита в двоичном представлении переменной, представляющей длину серии, и синтаксический элемент, который задает режим типа серии;

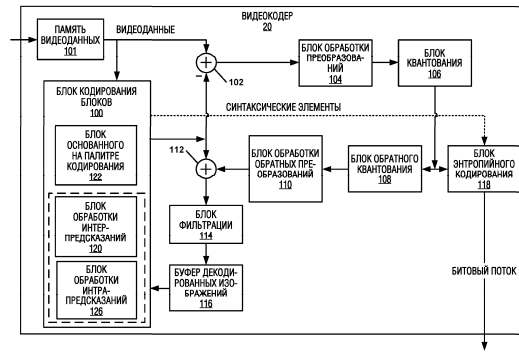
кодирования с использованием обходного режима контекстно-адаптивного двоичного арифметического кодирования множества экземпляров первого синтаксического элемента в кодированный битовый поток перед кодированием множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, в кодированный битовый поток с использованием контекстного режима контекстно-адаптивного двоичного арифметического кодирования; и

кодирования с использованием контекстного режима контекстно-адаптивного двоичного арифметического кодирования множества синтаксических элементов, которые отличаются от упомянутого первого синтаксического элемента, в кодированный битовый поток после кодирования множества экземпляров первого синтаксического элемента с использованием обходного режима контекстно-адаптивного двоичного арифметического кодирования в кодированный битовый поток.

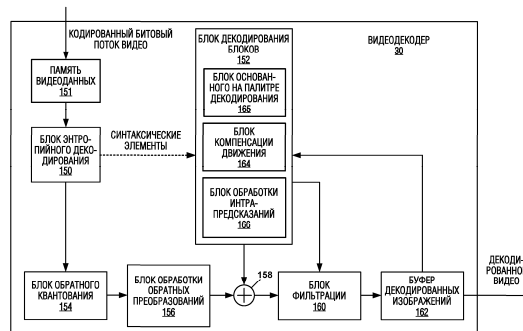
15. Долговременный машиночитаемый носитель данных, хранящий команды, которые при выполнении предписывают одному или нескольким процессорам выполнять способ по любому из пп.1-8 или 10-13.



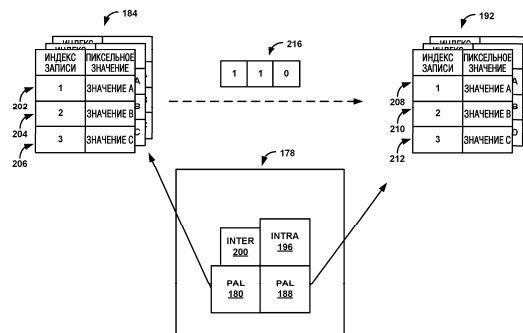
Фиг. 1



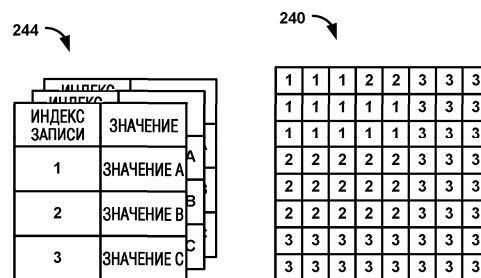
ФИГ. 2



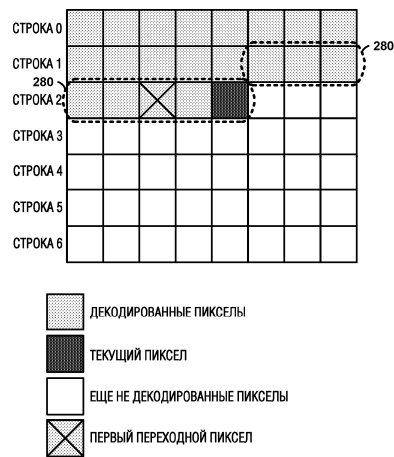
Фиг. 3



ФИГ. 4



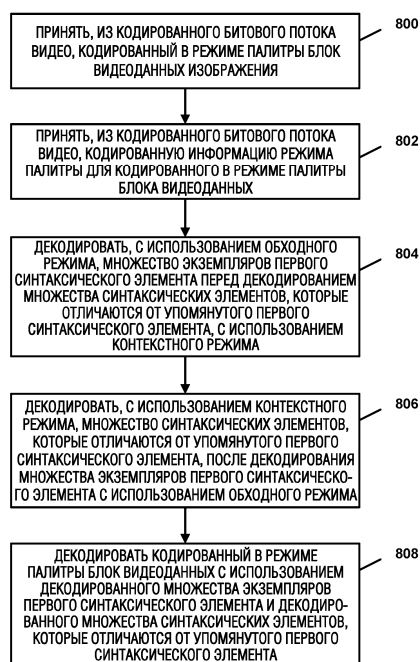
ФИГ. 5



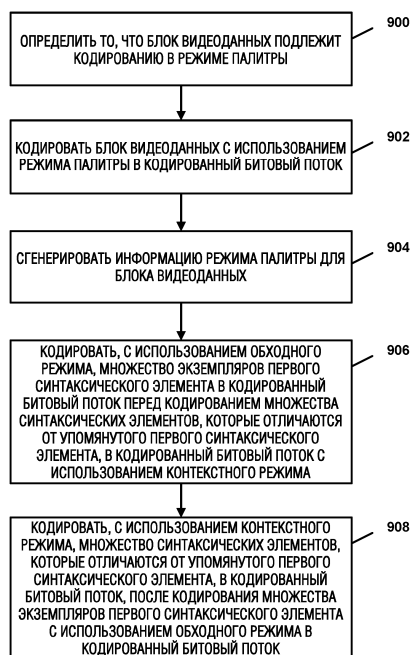
Фиг. 6

...	
if(currentPaletteSize != 0)	
palette_escape_val_present_flag	ae(v)
indices_idc_coding() (переместить сюда все события palette_index_idc)	
if(palette_escape_val_present_flag) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_palette_abs	ae(v)
if(cu_qp_delta_palette_abs)	
cu_qp_delta_palette_sign_flag	ae(v)
}	
if(cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_palette_offset_flag	ae(v)
if(cu_chroma_qp_palette_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_palette_offset_idx	ae(v)
}	
}	
if(indexMax > 0)	
palette_transpose_flag	ae(v)
scanPos = 0	
while(scanPos < nCbS * nCbS) {	
...	
if(palette_run_type_flag[xC][yC] == COPY_INDEX_MODE && adjustedIndexMax > 0)	
palette_index_idc	ae(v)
if(indexMax > 0) {	
maxPaletteRun = nCbS * nCbS - scanPos - 1	
if(maxPaletteRun > 0) {	
palette_run_msb_id_plus1	ae(v)
if(palette_run_msb_id_plus1 > 1)	
palette_run_refinement_bits	ae(v)
}	
} else	
...	

Фиг. 7



Фиг. 8



Фиг. 9

