



(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2020.04.29

(21) Номер заявки
201791499

(22) Дата подачи заявки
2016.01.29

(51) Int. Cl. *H04N 19/61* (2014.01)
G06F 9/30 (2006.01)
H04N 19/593 (2014.01)
H04N 19/126 (2014.01)

(54) ДЕКОДИРОВАНИЕ ВИДЕОДАННЫХ ДЛЯ ПИКСЕЛЕЙ ВНЕ ПАЛИТРЫ В БЛОКАХ, КОДИРУЕМЫХ В РЕЖИМЕ НА ОСНОВЕ ПАЛИТРЫ

(31) 62/110,519; 15/009,609

(32) 2015.01.31; 2016.01.28

(33) US

(43) 2017.11.30

(86) PCT/US2016/015672

(87) WO 2016/123492 2016.08.04

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

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

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

(56) Anonymous: "TComPrediction.cpp in tags/HM-16.2+SCM-3.0/source/Lib/TLibCommon - JCT-VC HEVC", 15 November 2014 (2014-11-15), XP055263033, Retrieved from the Internet: URL: <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.2%2BSCM-3.0/source/Lib/TLibCommon/TComPrediction.cpp> [retrieved on 2016-04-05], sentence 1515 - sentence 1564, sentence 957 - sentence 960, sentence 971 - sentence 974, sentence 986 - sentence 989, sentence 1009 - sentence 1011, sentence 1022 - sentence 1025

Anonymous: "#560 (Intra transform skipping semantic issue with high bit-depth) - JCT-VC HEVC", 2 July 2012 (2012-07-02), XP055263328, Retrieved from the Internet: URL: <https://hevc.hhi.fraunhofer.de/trac/hevc/ticket/560> [retrieved on 2016-04-06], the whole document

Anonymous: "Shifts and Rotates", 1 January 2007 (2007-01-01), XP055263894, Retrieved from the Internet: URL: <http://www.flexhex.com/docs/help/editing/shifts.phtml> [retrieved on 2016-04-08], the whole document

Anonymous: "TComRom.h in tags/HM-16.2+SCM-3.0/source/Lib/TLibCommon - JCT-VC HEVC", 11 November 2014 (2014-11-11), XP055263036, Retrieved from the Internet: URL: <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.2%2BSCM-3.0/source/Lib/TLibCommon/TComRom.h> [retrieved on 2016-04-05], sentence 96

Anonymous: "TComRom.cpp in tags/HM-16.2+SCM-3.0/source/Lib/TLibCommon - JCT-VC HEVC", 11 November 2014 (2014-11-11), XP055263040, Retrieved from the Internet: URL: <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.2%2BSCM-3.0/source/Lib/TLibCommon/TComRom.cpp> [retrieved on 2016-04-05], sentence 376 - sentence 384

"Proof of dates of files in HM-16.2+SCM-3.0" In: "Proof of dates of files in HM-16.2+SCM-3.0", 6 April 2016 (2016-04-06), XP055264052,

ZOU F. ET AL.: "CE1 Related: On escape pixel coding for palette mode", 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-T0112-v3, 10 February 2015 (2015-02-10), XP030117256, * sections 3 and 8.5 *

XIU X. ET AL.: "Non-CE1: On escape color coding for palette coding mode", 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-TSG.16); URL: [HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/](http://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/), no. JCTVC-T0118, 31 January 2015 (2015-01-31), XP030117264, * section 2.3 *

(57) Способ и устройство для декодирования видеоданных, посредством которых определяют, что значение для параметра сдвига вправо для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что значение для параметра сдвига вправо меньше нуля, устанавливают значение для параметра сдвига влево равным положительному значению, имеющему абсолютное значение, равное абсолютному значению параметра сдвига вправо; и выполняют обратное квантование пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

Эта заявка испрашивает приоритет временной заявки на патент США № 62/110519, поданной 31 января 2015 г., которая тем самым полностью включена в настоящий документ по ссылке.

Область техники

Это раскрытие относится к видеокодированию и, в частности, к кодированию видеоданных с использованием режима на основе палитры.

Уровень техники

Возможности цифрового видео могут быть включены в широкий диапазон устройств, в том числе в цифровые телевизоры, системы цифрового прямого вещания, беспроводные системы вещания, карманные персональные компьютеры (PDA), ноутбуки или настольные компьютеры, планшетные компьютеры, устройства для чтения электронных книг, цифровые фотоаппараты, устройства цифровой записи, проигрыватели цифровых данных, видеоигровые устройства, игровые приставки, сотовые или спутниковые радио-телефоны, так называемые "смартфоны", устройства организации видеоконференций, устройства потоковой передачи видеoinформации и т.п. Цифровые видеоустройства реализуют методики видеокодирования, например, описанные в стандартах MPEG 2, MPEG 4, ITU-T H.263, ITU-T H.264/MPEG-4 Часть 10, усовершенствованное видеокодирование (AVC), ITU-T H.265, также известное как высокоэффективное видеокодирование (HEVC), и в расширениях таких стандартов. Видеоустройства могут передавать, принимать, кодировать, декодировать и/или сохранять информацию цифрового видео более эффективно посредством реализации таких методик видеокодирования.

Методики видеокодирования включают в себя пространственное предсказание (внутри изображения) и/или временное предсказание (между изображениями), чтобы сократить или удалить избыточность, свойственную видеопоследовательностям. Для основанного на блоках видеокодирования видеослой (например, видеоизображение или участок видеоизображения) может быть разбит на видеоблоки, которые могут также упоминаться как элементы кодового дерева (CTU), элементы кодирования (CU) и/или узлы кодирования. Видеоблоки в интракодированном (I) слое изображения закодированы с использованием пространственного предсказания относительно опорных отсчетов в соседних блоках в том же самом изображении. Видеоблоки в интер-кодированном (P или B) слое изображения могут использовать пространственное предсказание относительно опорных отсчетов в соседних блоках того же самого изображения или временное предсказание относительно опорных отсчетов в других опорных изображениях. Изображения могут упоминаться как кадры, и опорные изображения могут упоминаться как опорные кадры.

Пространственное или временное предсказание дает в результате блок предсказания для блока, который должен быть закодирован. Разностные данные представляют различия в пикселях между первоначальным блоком, который должен быть закодирован, и блоком предсказания. Интер-кодированный блок кодируется в соответствии с вектором движения, который указывает блок опорных отсчетов, формирующих блок предсказания, и разностных данных, указывающих различие между закодированным блоком и блоком предсказания. Интра-кодированный блок кодируется в соответствии с режимом интракодирования и разностными данными. Для дополнительного сжатия разностные данные могут быть преобразованы из области пикселей в область преобразования, что приводит к разностным коэффициентам преобразования, которые затем могут быть квантованы. Квантованные коэффициенты преобразования, первоначально расположенные в двумерном массиве, могут быть просканированы, чтобы получить одномерный вектор коэффициентов преобразования, и может быть применено энтропийное кодирование для достижения еще большего сжатия.

Сущность изобретения

В целом это раскрытие описывает методики, относящиеся к квантованию значений для пикселей, кодируемых как пиксель вне палитры, блоков, кодируемых в режиме на основе палитры, видеоданных. В частности, в некоторых случаях традиционные методики кодирования пикселей вне палитры привели к определенным ситуациям, в которых двоичное значение должно было быть сдвинуто направо на отрицательную величину. Битовая операция сдвига на отрицательное значение обычно представляет собой неопределенную операцию, которая может вызвать ошибку. Методики этого раскрытия могут использоваться, чтобы исправить такие ошибки, когда они возникают. В частности, методики этого раскрытия могут использоваться, чтобы заменить значение смещения с отрицательного на положительное и выполнить битовую операцию сдвига влево вместо битовой операции сдвига вправо.

В одном примере способ включает в себя определение, что значение для параметра сдвига вправо для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что исходное значение для параметра сдвига вправо меньше нуля, установку значения для параметра сдвига влево равным положительному значению, имеющему абсолютное значение, равное абсолютному значению исходного значения; и выполнение обратного квантования пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

В другом примере устройство для декодирования видеоданных включает в себя память, выполненную с возможностью хранить видеоданные, и видеодекoder, выполненный с возможностью определять, что значение для параметра сдвига вправо для пикселя, кодируемого как пиксель вне палитры, блока,

кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что значение для параметра сдвига вправо меньше нуля, устанавливать значение для параметра сдвига влево равным положительному значению, имеющему абсолютное значение, равное абсолютному значению параметра сдвига вправо; и выполнять обратное квантование пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

В другом примере устройство для декодирования видеоданных включает в себя средство для определения, что значение для параметра сдвига вправо для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля, средство для установки на основе того, что значение для параметра сдвига вправо меньше нуля, значения для параметра сдвига влево равным положительному значению, имеющему абсолютное значение, равное абсолютному значению параметра сдвига вправо, и средство для выполнения обратного квантования пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

В другом примере машиночитаемый запоминающий носитель закодирован с помощью команд, которые при их исполнении заставляют программируемый процессор определять, что значение для параметра сдвига вправо для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что значение для параметра сдвига вправо меньше нуля, устанавливать значение для параметра сдвига влево равным положительному значению, имеющему абсолютное значение, равное абсолютному значению параметра сдвига вправо, и выполнять обратное квантование пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

В другом примере способ кодирования видеоданных включает в себя определение, что значение для параметра сдвига влево для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что значение для параметра сдвига влево меньше нуля, установку значения для параметра сдвига вправо равным положительному значению, имеющему абсолютное значение, равное абсолютному значению параметра сдвига влево; и выполнение квантования пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

Подробности одного или более примеров изложены на прилагаемых чертежах и в приведенном ниже описании. Другие признаки, объекты и преимущества будут очевидны из описания, чертежей и формулы изобретения.

Краткое описание чертежей

Фиг. 1 - блок-схема, демонстрирующая иллюстративную систему кодирования и декодирования видеоинформации, которая может использовать методики этого раскрытия для кодирования пикселей вне палитры в блоках, кодируемых в режиме на основе палитры.

Фиг. 2 - блок-схема, демонстрирующая пример видеокодера, который может реализовать методики для кодирования пикселей вне палитры в блоках, кодируемых в режиме на основе палитры, в соответствии с методиками этого раскрытия.

Фиг. 3 - блок-схема, иллюстрирующая пример видеодекодера, который может реализовать методики для декодирования пикселей вне палитры в блоках, кодируемых в режиме на основе палитры, в соответствии с методиками этого раскрытия.

Фиг. 4 - блок-схема последовательности этапов, демонстрирующая иллюстративный способ кодирования блока видеоданных в соответствии с методиками этого раскрытия.

Фиг. 5 - блок-схема последовательности этапов, демонстрирующая иллюстративный способ декодирования блока видеоданных в соответствии с методиками этого раскрытия.

Фиг. 6 - блок-схема последовательности этапов, демонстрирующая иллюстративную методику, посредством которой блок декодирования на основе палитры может выполнить обратное квантование квантованного значения пикселя вне палитры закодированного в режиме на основе палитры блока видеоданных в соответствии с методиками этого раскрытия.

Подробное описание

В целом эта заявка описывает методики для поддержки кодирования видеоинформации, особенно содержания экрана с помощью кодирования в режиме на основе палитры. В частности, эти методики относятся к кодированию пикселя вне палитры для кодирования в режиме на основе палитры (также называемого "кодированием на основе палитры").

В традиционном видеокодировании предполагается, что изображения имеют непрерывный тон и являются пространственно сглаженными. На основе этих предположений были разработаны различные инструменты, такие как поблочное преобразование, фильтрация и т.д., и такие инструменты показали хорошие результаты для видео с естественным содержанием. Однако в таких применениях, как удаленный рабочий стол, совместная работа и беспроводной дисплей, сформированное компьютером содержание экрана (например, текст или компьютерная графика) может являться преобладающим содержанием для сжатия. Этот тип информационного содержания как правило имеет прерывистый тон, и для него характерны острые линии и высококонтрастные границы объектов. Предположение о непрерывном тоне и сглаженности больше не может применяться к содержанию экрана, и, таким образом, традиционные ме-

тодики видеокодирования могут не являться эффективными методами сжатия видеоданных, включающих в себя содержание экрана.

На основе характеристик видеоинформации содержания экрана было введено в использование кодирование на основе палитры для повышения эффективности кодирования содержания экрана (SCC), и оно сначала было предложено в литературе Guo et al., "Palette Mode for Screen Content Coding," JCT-VC of ITU-T SG 13 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 13th Meeting, JCTVC-M0323, Incheon, KR. 18-26 April 2013. Более конкретно, кодирование на основе палитры вводит таблицу поиска, т.е., цветовую палитру, для сжатия повторяющихся значений пикселей на основе того, что в кодировании SCC цвета в одном элементе кодирования (CU) обычно концентрируются на нескольких пиковых значениях. Если имеется палитра для заданного элемента CU, пиксели в элементе CU отображаются на индекс палитры. На второй стадии предложен эффективный метод длин серий с копированием слева для эффективного сжатия повторяющийся шаблон блока индекса. Позже, в литературе Guo et al., "Non-RCE3: Modified Palette Mode for Screen Content Coding," JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-N0249, 14th Meeting, Vienna, AT, 25 July - 2 Aug. 2013 индексный режим кодирования палитры был обобщен для копирования слева и копирования сверху с кодированием длин серий. Следует отметить, для кодирования на основе палитры не вызывается процесс преобразования, чтобы избежать размытия резких краев, которое оказывает очень большое негативное влияние на визуальное качество содержания экрана.

Палитра представляет собой структуру данных, которая хранит пары (индекс, значение пикселя). Видеокодер может определить палитру, например, на основе гистограммы значений пикселей в текущем элементе CU. Например, пиковые значения в гистограмме могут быть добавлены в палитру, в то время как значения пикселей с низкой частотой не обязательно должны быть включены в палитру. Размер палитры может быть ограничен диапазоном от 0 до `max_palette_size`, равному 31.

Для кодирования SCC блоки элемента CU в одном слое могут совместно использовать многие преобладающие цвета. Таким образом, возможно предсказать палитру текущего блока с использованием предыдущих палитр элементов CU режима на основе палитры (в порядке декодирования элементов CU) в качестве опорных. Более конкретно, может быть сигнализирован двоичный вектор 0-1, чтобы указать, используются ли повторно текущей палитрой значения пикселей в опорной палитре. В качестве примера в приведенных ниже табл. 1 и 2 предполагается, что опорная палитра имеет 6 элементов. С текущей палитрой сигнализирован вектор (1, 0, 1, 1, 1, 1), который указывает, что v_0 , v_2 , v_3 , v_4 и v_5 повторно использованы в текущей палитре, в то время как v_1 повторно не использован. Если текущая палитра содержит цвета, которые не предсказуемы на основе опорной палитры, кодируется количество не предсказанных цветов, и затем эти значения пикселей (например, значения яркости или цветности) сигнализируются непосредственно. Например, в табл. 1 и 2 значения u_0 и u_1 непосредственно сигнализированы в битовом потоке.

Таблица 1. Опорная палитра

Индекс	Значение пикселя
0	V_0
1	V_1
2	V_2
3	V_3
4	V_4
5	V_5

Таблица 2. Текущая палитра

Флаг предсказания	Индекс	Значение пикселя
1	0	V_0
0		
1	1	V_2
1	2	V_3
1	3	V_4
	4	V_5
	5	U_0
	6	U_1

Для блока, закодированного с режимом на основе палитры, палитра может быть предсказана на основе записей палитры предыдущих закодированных на основе палитры блоков, может быть явно сигнализирована как новые записи, или палитра предыдущего закодированного блока может быть полностью повторно использована. Последний случай называют совместным использованием палитры, и сигнализируется флаг `palette_share_flag`, чтобы указать, что вся палитра предыдущего блока повторно использована без модификации, как есть. Примеры опорной палитры и текущей палитры показаны в приведенных выше табл. 1 и 2. В частности, табл. 1 демонстрирует иллюстративную опорную палитру и табл. 2 демонстрирует иллюстративную текущую палитру, которая может быть предсказана на основе опорной палитры

табл. 1.

В текущем эталонном программном обеспечении SCM3.0 двумя первичными аспектами кодирования на основе палитры с нормативной точки зрения являются кодирование палитры и кодирование индекса палитры для каждого отсчета в блоке, закодированном в режиме на основе палитры. Кодирование индексов палитры выполняется с использованием двух первичных режимов, "индексного" режима и режима "копирования сверху". Они сообщаются посредством кодирования флага `palette_mode`. "Индексный" режим также используется для указания отсчетов вне палитры, т.е., отсчетов, которые не принадлежат палитре. В текущих проектных решениях режим "копирования сверху" невозможен для первой строки палитры. Кроме того, за режимом "копия сверху" не может следовать другой режим "копия" сверху. В этих случаях подразумевается "индексный" режим.

Более конкретно, для режима на основе палитры пиксели в элементе CU закодированы в порядке горизонтального/вертикального прохода сканирования "змейкой" следующим образом:

1. "Индексный" режим:

В этом режиме сначала сообщается один индекс палитры. Если индекс равен размеру палитры, это указывает, что отсчет является отсчетом вне палитры. В этом случае сообщается значение отсчета или квантованное значение отсчетов для каждого компонента (например, яркости и цветности). Например, если размер палитры равен 4, для отсчетов, не являющихся отсчетами вне палитры, индексы палитры находятся в диапазоне [0, 3]. В этом случае значение индекса 4 показывает отсчет вне палитры. Если индекс указывает отсчет, не являющийся отсчетом вне палитры, сообщается длина серии, которая определяет количество последующих отсчетов в порядке сканирования, которые имеют одинаковый индекс, посредством неотрицательного значения $n-1$, указывающего длину серии, что означает, что следующие n пикселей, включая текущий, имеют такой же индекс пикселя, как первый сообщенный.

2. Режим серии "копирования сверху" (CA):

В этом режиме передается только неотрицательное значение $m-1$ длины серии, чтобы указать, что для следующих m пикселей, включая текущий, индексы палитры совпадают со своими непосредственными соседями сверху соответственно. Следует отметить, что этот режим отличается от "индексного" режима в том смысле, что индексы палитры могут быть разными в рамках режима серии "копирования сверху".

В текущих проектных решениях режим на основе палитры сообщается на уровне элемента CU, но возможно сообщить его на уровне элемента PU. Флаг `palette_esc_val_present_flag` также может быть сообщен, чтобы указать наличие отсчетов вне палитры в текущем блоке.

В режиме на основе палитры сканирование пикселей в блоке может иметь два типа: вертикальное сканирование с возвратом или горизонтальное сканирование с возвратом ("змейкой"). Шаблон сканирования, используемый в блоке, выявляется в соответствии с флагом `palette_transpose_flag`, сообщается для каждого блока.

Во время кодирования индекса палитры может быть применен процесс корректировки индексов палитры. Начиная со второго пикселя в блоке, он состоит в проверке режима на основе палитры предыдущего закодированного пикселя. Сначала размер палитры уменьшается на 1, и если режим слева равен режиму серии, то индекс палитры, который должен быть закодирован, уменьшается на 1, если индекс больше, чем индекс палитры слева, или если режим слева является режимом копирования, то индекс палитры, который должен быть закодирован, уменьшается на 1, если индекс больше, чем индекс палитры сверху. Описание представлено на стороне кодирования, и соответствующий процесс также может быть выполнен в обратном порядке на стороне декодера.

В SCM-3.0 была применена следующая оптимизация синтаксиса:

Если размер палитры равен 0, выявлены все пиксели вне палитры и не сообщаются флаг наличия пикселей вне палитры, режим на основе палитры, индекс палитры, длина серии палитры и флаг шаблона сканирования палитры и подразумевается, что флаг наличия пикселей вне палитры равен 1, подразумевается, что режим на основе палитры равен режиму INDEX, индекс палитры установлен равным ESCAPE, значение длины серии палитры устанавливается равным размеру блока и флаг шаблона сканирования палитры устанавливается равным 0.

Если размер палитры равен 1 и пиксели вне палитры не используются в блоке, то не сообщаются режим на основе палитры, длины серии палитры или флага шаблона сканирования палитры, и выявляется, что режим на основе палитры равен режиму INDEX, индекс палитры устанавливается равным 0, значение длины серии палитры устанавливается равным размеру блока и флаг шаблона сканирования палитры устанавливается равным 0.

Это раскрытие описывает методики, относящиеся к кодированию на основе палитры, которое может быть особенно подходящим для кодирования сформированного на экране содержания. Например, в предположении, что конкретная область видеоданных имеет относительно небольшое количество цветов, устройство видеокодирования (например, видеокодер или видеodeкодер) может сформировать так называемую "палитру" для представления видеоданных конкретной области. Палитра может быть выражена как таблица цветов или значений пикселей, представляющих видеоданные конкретной области (например, данный блок). Например, палитра может включать в себя наиболее преобладающие значения

пикселей в данном блоке. В некоторых случаях наиболее преобладающие значения пикселей могут включать в себя одно или более значений пикселей, которые наиболее часто встречаются в блоке. Кроме того, в некоторых случаях устройство видеокодирования может применять пороговое значение, чтобы определить, должно ли значение пикселя быть включено как одно из наиболее преобладающих значений пикселей в блоке. В соответствии с различными аспектами кодирования на основе палитры устройство видеокодирования может кодировать индексные значения, указывающие одно или более значений пикселей текущего блока, вместо того, чтобы кодировать фактические значения пикселей или их разности для текущего блока видеоданных. В контексте кодирования на основе палитры индексные значения указывают соответствующие записи в палитре, которые используются для представления индивидуальных значений пикселей текущего блока.

Например, видеокодер может закодировать блок видеоданных, определяя палитру для блока (например, явное кодирование палитры, предсказание палитры или их комбинация), определяя положение записи в палитре для представления одного или более значений пикселей и кодируя блок с индексными значениями, которые указывают записи в палитре, использованные для представления значений пикселей блока. В некоторых примерах видеокодер может сообщать палитру и/или индексные значения в закодированном битовом потоке. В свою очередь видеодекoder может получить из закодированного битового потока палитру для блока, а также индексные значения для индивидуальных пикселей блока. Видеодекoder может соотнести индексные значения пикселей с записями палитры, чтобы воссоздать различные значения пикселей блока.

В частности, пиксели блока, закодированного с использованием режима на основе палитры, могут быть закодированы с использованием "индексного" режима, в котором пиксель кодируется с использованием ссылки на палитру, или режима "копирования сверху", в котором пиксель кодируется с использованием ссылки на соседний пиксель сверху. Третий вариант состоит в том, чтобы закодировать пиксель как пиксель вне палитры. В этом случае значение пикселя (или квантованное значение для пикселя) сообщается непосредственно.

В текущем SCM3.0 по состоянию на время этого раскрытия обратное квантование используется для воссоздания квантованных пикселей вне палитры в режиме на основе палитры. Более конкретно, следующая процедура используется для воссоздания пикселей вне палитры:

1) Параметр qP квантования выводится в соответствии со значениями индекса ($cIdx$) другого цветового компонента следующим образом:

$$qP = (cIdx == 0) ? Qp'Y : ((cIdx == 1) ? Qp'Cb : Qp'Cr)$$

2) Отношение qP_{per} квантования и остаток qP_{rem} квантования выводятся следующим образом:

$$qP_{per} = qP / 6$$

$$qP_{rem} = qP \% 6$$

3) Параметр $invQuantRightShift$ сдвига вправо и параметр $addOffset$ смещения выводятся следующим образом:

$$\begin{aligned} invQuantRightShift &= 6 - qP_{per} \\ addOffset &= invQuantRightShift == 0 ? 0 : 1 << (invQuantRightShift - 1) \end{aligned}$$

4) Деквантованный пиксель вне палитры $deQuantEspValue$ выводится на основе декодированного посредством энтропийного декодирования $EspValue$ следующим образом:

$$\begin{aligned} deQuantEspValue &= (EspValue * invQuantScale[qP_{rem}] + addOffset \\ & \gg invQuantRightShift \end{aligned}$$

5) Значение $deQuantEspValue$ далее модифицируется, чтобы оно находилось в диапазоне глубины, следующим образом:

$$deQuantEspValue = clip3(0, 2^{bitDepth} - 1, deQuantEspValue)$$

Одна проблема, с которой можно столкнуться в текущих проектных решениях SCM3.0, состоит в том, что когда qP_{per} больше 6, значение $invQuantRightShift$ является отрицательным. Не существует определения сдвига вправо на отрицательное число. Таким образом, этот сценарий может вызвать разную интерпретацию обратного квантования для разных декодеров, и это не желательно для практического проектирования.

В соответствии с методиками этого раскрытия устройства видеокодирования (например, видеокодеры и видеодекодеры) могут быть выполнены с возможностью избежать описанную выше проблему. В частности, устройства видеокодирования могут быть выполнены с возможностью адресоваться к полному спектру значений параметра квантования (qP). Например, qP может находиться в диапазоне [0, 1, ..., 51] включительно. И разные процессы обратного квантования могут использоваться для воссоздания пикселей вне палитры для режима на основе палитры. Эти и другие методики описаны более подробно ниже.

Фиг. 1 является блок-схемой, показывающей иллюстративную систему 10 кодирования и декодирования видеoinформации, которая может использовать методики для кодирования пикселей вне палитры в блоках, кодируемых в режиме на основе палитры. Как показано на фиг. 1, система 10 включает в себя устройство 12 источника, обеспечивающее закодированные видеоданные, которые должны быть позднее

декодированы устройством 14 назначения. В частности, устройство 12 источника обеспечивает видеоданные для устройства 14 назначения через машиночитаемый носитель 16. Устройство 12 источника и устройство 14 назначения могут содержать любое из широкого диапазона устройств, включающего в себя настольные компьютеры, ноутбуки (или переносные компьютеры), планшетные компьютеры, телеприставки, переносные телефоны, такие как так называемые "интеллектуальные" телефоны (смартфоны), так называемые "интеллектуальные" клавиатуры, телевизоры, камеры, устройства отображения, проигрыватели цифровых данных, консоли видеоигр, устройство потоковой передачи видеoinформации и т.п. В некоторых случаях устройство 12 источника и устройство 14 назначения могут быть оборудованы беспроводной связью.

Устройство 14 назначения может принимать закодированные видеоданные, которые должны быть декодированы, через машиночитаемый носитель 16. Машиночитаемый носитель 16 может содержать носитель или устройство любого типа, способные переносить закодированные видеоданные от устройства 12 источника к устройству 14 назначения. В одном примере машиночитаемый носитель 16 может содержать средство связи для обеспечения возможности для устройства 12 источника передавать закодированные видеоданные непосредственно устройству 14 назначения в реальном времени. Закодированные видеоданные могут быть модулированы в соответствии со стандартом связи, таким как протокол беспроводной связи, и переданы устройству 14 назначения. Средство связи может содержать любое средство беспроводной или проводной связи, такое как радиочастотный (RF) спектр или одна или несколько физических линий передачи. Средство связи может являться частью пакетной сети, такой как локальная сеть, широкомасштабная сеть или глобальная сеть, например Интернет. Средство связи может включать в себя маршрутизаторы, переключатели, базовые станции или любое другое оборудование, которое может быть полезным для обеспечения возможности связи от устройства 12 источника до устройства 14 назначения.

В некоторых примерах закодированные данные могут быть выданы из выходного интерфейса 22 устройству хранения. Аналогичным образом к закодированным данным может быть получен доступ от устройства хранения посредством входного интерфейса. Устройство хранения может включать в себя любой из множества распределенных или локально доступных носителей данных, таких как накопитель на жестком диске, диски Blu-ray, цифровые универсальные диски (DVD), компакт-диски (CD-ROM), флэш-память, энергозависимая или энергонезависимая память или любые другие подходящие цифровые носители для хранения закодированных видеоданных. В дополнительном примере устройство хранения может соответствовать файловому серверу или другому промежуточному устройству хранения, которое может хранить закодированную видеoinформацию, сформированную устройством 12 источника. Устройство 14 назначения может осуществить доступ к сохраненным видеоданным от устройства хранения через потоковую передачу или загрузку. Файловый сервер может представлять собой сервер любого типа, способный хранить закодированные видеоданные и передавать эти закодированные видеоданные устройству 14 назначения. Иллюстративные файловые серверы включают в себя веб-сервер (например, для веб-сайта), сервер FTP, сетевые хранилища данных (NAS) или локальный дисковый накопитель. Устройство 14 назначения может осуществить доступ к закодированным видеоданным посредством любого стандартного соединения передачи данных, в том числе Интернет-соединения. Оно может включать в себя беспроводной канал (например, соединение Wi-Fi), проводное соединение (например, DSL, кабельный модем и т.д.) или их комбинацию, которая является подходящей для доступа к закодированным видеоданным, сохраненным на файловом сервере. Передача закодированных видеоданных от устройства хранения может представлять собой потоковую передачу, передачу загрузки или их комбинацию.

Методики этого раскрытия не обязательно ограничены приложениями или настройками в беспроводной связи. Методики могут быть применены к видеокодированию при поддержке любого из множества мультимедийных приложений, таких как эфирное телевидение, передачи кабельного телевидения, передачи спутникового телевидения, передачи потокового видео через Интернет, например, динамическая адаптивная потоковая передача по HTTP (DASH), цифровое видео, которое закодировано на носителе данных, декодирование цифрового видео, сохраненного на носителе данных, или других приложений. В некоторых примерах система 10 может быть выполнена с возможностью поддержки односторонней или двухсторонней передачи видеoinформации для поддержки таких приложений, как потоковая передача видеoinформации, воспроизведение видео, видеовещание и/или видеотелефония.

В примере на фиг. 1 устройство 12 источника включает в себя источник 18 видеoinформации, видеокодер 20 и выходной интерфейс 22. Устройство 14 назначения включает в себя входной интерфейс 28, видеодекодер 30 и устройство 32 отображения. В соответствии с этим раскрытием видеокодер 20 устройства 12 источника может быть выполнен с возможностью применять методики для кодирования пикселей вне палитры в блоках, кодируемых в режиме на основе палитры. В других примерах устройство источника и устройство назначения могут включать в себя другие компоненты или устройства. Например, устройство 12 источника может принимать видеоданные от внешнего источника 18 видеoinформации, такого как внешняя камера. Аналогичным образом, устройство 14 назначения может взаимодействовать через интерфейс с внешним устройством отображения вместо того, чтобы включать в себя интегрированное устройство отображения.

Проиллюстрированная система 10 на фиг. 1 является лишь одним примером. Методики для кодирования пикселей вне палитры в блоках, кодируемых в режиме на основе палитры, могут быть выполнены любым устройством кодирования и/или декодирования цифрового видео. Хотя обычно методики этого раскрытия выполняются устройством кодирования видеоинформации, методики также могут быть выполнены видеокодером/видеодекодером, обычно называемым "кодек" ("CODEC"). Кроме того, методики этого раскрытия также могут быть выполнены препроцессором видеоинформации. Устройство 12 источника и устройство 14 назначения являются лишь примерами таких кодеров, в которых устройство 12 источника формирует закодированные видеоданные для передачи устройству 14 назначения. В некоторых примерах устройства 12, 14 могут работать в значительной степени симметричным способом, так что каждое из устройств 12, 14 включают в себя компоненты кодирования и декодирования видеоинформации. Следовательно, система 10 может поддерживать одностороннюю или двухстороннюю передачу видеоинформации между видеоустройствами 12, 14, например для потоковой передачи видеоинформации, воспроизведения видео, видеовещания или видеотелефонии.

Источник 18 видеоинформации устройства 12 источника может включать в себя устройство видеосъемки, такое как видеокамера, видеоархив, содержащий ранее снятую видеоинформацию, и/или интерфейс видеоканала для приема видеоинформации от провайдера видеоконтента. В качестве дополнительной альтернативы источник 18 видеоинформации может формировать данные на основе компьютерной графики в качестве видеоинформации источника или комбинации "живой" видеоинформации, архивированной видеоинформации и сформированной с помощью компьютера видеоинформации. В некоторых случаях, если источник 18 видеоинформации является видеокамерой, устройство 12 источника и устройство 14 назначения могут формировать так называемые телефоны с камерой или видеотелефоны. Однако, как упомянуто выше, описанные в этом раскрытии методики могут быть применены к видеокодированию вообще и могут быть применены к приложениям беспроводной и/или проводной связи. В каждом случае снимаемая, предварительно снятая или сформированная с помощью компьютера видеоинформация может быть закодирована видеокодером 20. Закодированная видеоинформация затем может быть выдана выходным интерфейсом 22 на машиночитаемый носитель 16.

Машиночитаемый носитель 16 может включать в себя преходящие носители, такие как беспроводное широковещение или проводная сетевая передача, или запоминающие носители (то есть, непреходящие носители), такие как жесткий диск, карта флэш-памяти, компакт-диск, цифровой видеодиск, диск Blu-ray или другие машиночитаемые носители. В некоторых примерах сетевой сервер (не показан) может принимать закодированные видеоданные от устройства 12 источника и обеспечивать закодированные видеоданные для устройства 14 назначения, например, через сетевую передачу. Аналогичным образом, вычислительное устройство предприятия по производству носителей, такого как предприятие по штамповке дисков, может принимать закодированные видеоданные от устройства 12 источника и производить диск, содержащий закодированные видеоданные. Таким образом, можно понимать, что в различных примерах машиночитаемый носитель 16 включает в себя один или несколько машиночитаемых носителей различных форм.

Входной интерфейс 28 из устройства 14 назначения принимает информацию от машиночитаемого носителя 16. Информация машиночитаемого носителя 16 может включать в себя синтаксическую информацию, определенную видеокодером 20, которая также используется видеодекодером 30, включающую в себя синтаксические элементы, которые описывают характеристики и/или обработку блоков и других закодированных элементов, например, групп изображений (GOP). Устройство 32 отображения отображает декодированные видеоданные пользователю и может содержать любое из множества устройств отображения, таких как электронно-лучевая трубка (CRT), жидкокристаллический дисплей (LCD), плазменный дисплей, органический светодиодный (OLED) дисплей или устройство отображения другого типа.

Видеокодер 20 и видеодекодер 30 могут работать в соответствии со стандартом видеокодирования, таким как стандарт высокоэффективного кодирования (High Efficiency Video Coding, HEVC), также называемый ITU-T H.265. В качестве альтернативы видеокодер 20 и видеодекодер 30 могут работать в соответствии с другими частными или промышленными стандартами, такими как стандарт ITU-T H.264, альтернативно называемый MPEG-4. часть 10, усовершенствованное видеокодирование (Advanced Video Coding, AVC), или расширениями таких стандартов. Однако методики этого раскрытия не ограничены каким-либо конкретным стандартом кодирования. Другие примеры стандартов видеокодирования включают в себя MPEG-2 и ITU-T H.263. Хотя это не показано на фиг. 1, в некоторых аспектах каждый из видеокодера 20 и видеодекодера 30 может быть интегрирован со звуковым кодером и декодером и может включать в себя соответствующие элементы мультимплексора-демультиплексора или другое аппаратное и программное обеспечение для обработки кодирования как аудиоинформации, так и видеоинформации в потоке общих данных или в отдельных потоках данных. В соответствующих случаях элементы мультимплексора-демультиплексора могут соответствовать протоколу мультимплексора ITU H.223 или другим протоколам, таким как протокол пользовательских дейтаграмм (UDP).

Видеокодер 20 и видеодекодер 30 в соответствующих случаях могут быть реализованы как любая из множества подходящих схем кодера, например один или несколько микропроцессоров, процессоров

цифровых сигналов (DSP), специализированных интегральных схем (ASIC), программируемых пользователем вентильных матриц (FPGA), дискретных логических схем, программного обеспечения, аппаратных средств, программируемого оборудования или любой их комбинации. Когда методики реализованы частично в программном обеспечении, устройство может хранить команды для программного обеспечения в подходящем энергонезависимом машиночитаемом носителе и исполнять команды на аппаратных средствах с использованием одного или более процессоров, чтобы выполнить методики этого раскрытия. Каждый из видеокодера 20 и видеodeкодера 30 может быть включен в один или более кодеров или декодеров, любой из которых может быть интегрирован как часть комбинированного кодера/декодера (кодека) в соответствующем устройстве.

В целом в соответствии со стандартом ITU-T H.265 видеоизображение может быть разделено на последовательность элементов кодового дерева (CTU) (или наибольших элементов кодирования (LCU)), которые могут включать в себя отчеты яркости и цветности. В качестве альтернативы элементы CTU могут включить в себя монохромные данные (т.е., только отсчеты яркости). Синтаксические данные в битовом потоке могут определять размер для элемента CTU, который является наибольшим элементом кодирования с точки зрения количества пикселей. Слой включает в себя несколько последовательных элементов CTU в порядке кодирования. Видеоизображение может быть разбито на один или более слоев. Каждый элемент CTU может быть разбит на элементы кодирования (CU) в соответствии с деревом квадратов. В целом структура данных дерева квадратов включает в себя один узел на каждый элемент CU с корневым узлом, соответствующим элементу CTU. Если элемент CU разбит на четыре подэлемента CU, узел, соответствующий элементу CU, включает в себя четыре листовых узла, каждый из которых соответствует одному из подэлементов CU.

Каждый узел структуры данных дерева квадратов может обеспечить синтаксические данные для соответствующего элемента CU. Например, узел в дереве квадратов может включать в себя флаг разделения, указывающий, разбит ли элемент CU, соответствующий узлу, на подэлементы CU. Синтаксические элементы для элемента CU могут быть определены рекурсивно и могут зависеть от того, разбит ли элемент CU на подэлементы CU. Если элемент CU далее не разбит, он называется листовым элементом CU. В этом раскрытии четыре подэлемента CU листового элемента CU будут также упоминаться как листовые элементы CU, даже если не будет никакого явного разделения первоначального листового элемента CU. Например, если элемент CU с размером 16×16 далее не разбивается, четыре подэлемента CU с размером 8×8 также будут называться листовыми элементами CU, хотя элемент CU с размером 16×16 никогда не был разбит.

Элемент CU имеет такое же назначение, как макроблок стандарта H.264, за исключением того, что элемент CU не имеет различия по размеру. Например, элемент CTU может быть разбит на четыре дочерних узла (также называемых подэлементами CU), и каждый дочерний узел может в свою очередь являться родительским узлом и быть разбит еще на четыре дочерних узла. Заключительный, неразбитый дочерний узел, называемый листовым узлом дерева квадратов, содержит узел кодирования, также называемый листовым элементом CU. Синтаксические данные, соответствующие закодированному битовому потоку, могут определить максимальное количество разбиений элемента CTU, называемое максимальной глубиной элемента CU, и также может определить минимальный размер узлов кодирования. В соответствии с этим битовый поток также может определить наименьший элемент кодирования (SCU). Это раскрытие использует термин "блок", чтобы называть любой из элементов CU, элементов предсказания (PU) или элементов преобразования (TU) в контексте HEVC или аналогичные структуры данных в контексте других стандартов (например, макроблоки и их субблоки в стандарте H.264/AVC).

Элемент CU включает в себя узел кодирования и элементы предсказания (PU) и элементы преобразования (TU), соответствующие узлу кодирования. Размер элемента CU соответствует размеру узла кодирования и обычно имеет квадратную форму. Размер элемента CU может находиться в диапазоне от 8×8 пикселей до размера элемента CTU с максимальным размером 64×64 пикселей или больше. Каждый элемент CU может содержать один или несколько элементов PU и один или несколько элементов TU. Синтаксические данные, соответствующие элементу CU, могут описывать, например, разделение элемента CU на один или несколько элементов PU. Режимы разделения может отличаться в том, является ли элемент CU закодированным в режиме пропуска или в прямом режиме, закодированным в режиме интра-предсказания, или закодированным в режиме интер-предсказания. При разделении элементы PU могут являться не квадратными по форме. Синтаксические данные, соответствующие элементу CU, также могут описывать, например, разделение элемента CU на один или несколько элементов TU в соответствии с деревом квадратов. Элемент TU может являться квадратным или не квадратным (например, прямоугольным) по форме.

Стандарт HEVC учитывает преобразования в соответствии с элементами TU, которые могут отличаться для разных элементов CU. Элементы TU обычно измеряются на основе размера элементов PU в пределах данного элемента CU, заданного для разделенного элемента CTU, хотя это не всегда может иметь место. Элементы TU обычно имеют такой же или меньший размер, как элементы PU. В некоторых примерах разностные отсчеты, соответствующие элементу CU, могут быть подразделены на меньшие

элементы с использованием структуры дерева квадрантов, известной как "разностное дерево квадрантов" (RQT). Листовые узлы дерева RQT могут называться элементами преобразования (элементами TU). Значения разности пикселей, соответствующие элементам TU, могут быть преобразованы для получения коэффициентов преобразования, которые могут быть квантованы.

Листовой элемент CU может включать в себя один или несколько элементов предсказания (PU). В общем случае элемент PU представляет пространственную область, соответствующую всему или части соответствующего элемента CU, и может включать в себя данные для извлечения и/или формирования опорного отсчета для элемента PU. Кроме того, элемент PU включает в себя данные, относящиеся к предсказанию. Например, когда элемент PU закодирован в интра-режиме, данные для элемента PU могут быть включены в разностное дерево квадрантов (RQT), которое может включать в себя данные, описывающие режим интра-предсказания для элемента TU, соответствующего элементу PU. Дерево RQT также может упоминаться как дерево преобразования. В некоторых примерах режим интра-предсказания может быть сигнализирован в синтаксисе листового элемента CU вместо дерева RQT. В качестве другого примера, когда элемент PU закодирован в интер-режиме, элемент PU может включать в себя данные, определяющие информацию движения, такую как один или несколько векторов движения, для элемента PU. Данные, определяющие вектор движения для элемента PU, могут описывать, например, горизонтальную составляющую вектора движения, вертикальную составляющую вектора движения, разрешение для вектора движения (например, точность четверть пикселя или точность одна восьмая пикселя), опорное изображение, на которое указывает вектор движения, и/или список опорных изображений (например, список 0 или список 1) для вектора движения.

Листовой элемент CU, имеющий один или несколько элементов PU, также может включать в себя один или несколько элементов преобразования (элементов TU). Элементы преобразования могут быть определены с использованием дерева RQT (также называемого структурой дерева квадрантов элемента TU), как обсуждено выше. Например, флаг разделения может указывать, разбит ли листовой элемент CU на четыре элемента преобразования. Затем каждый элемент преобразования может быть далее разбит на дополнительные подэлементы TU. Когда элемент TU далее не разбит, он может называться листовым элементом TU. Обычно для интра-кодирования все листовые элементы TU, принадлежащие листовому элементу CU, совместно используют один и тот же режим интра-предсказания. Таким образом, один и тот же режим интра-предсказания обычно применяется для вычисления предсказанных значений для всех элементов TU листового элемента CU. Для интра-кодирования видеокoder может вычислить разностное значение для каждого листового элемента TU с использованием режима интра-предсказания как разность между секцией элемента CU, соответствующего элементу TU, и первоначальным блоком.

Элемент TU не обязательно ограничен размером элемента PU. Таким образом, элементы TU могут быть больше или меньше элемента PU. Для интра-кодирования элемент PU может быть расположен совместно с соответствующим листовым элементом TU для того же самого элемента CU. В некоторых примерах максимальный размер листового элемента TU может соответствовать размеру соответствующего листового элемента CU.

Кроме того, элементы TU листовых элементов CU также могут относиться к соответствующим структурам данных дерева квадрантов, называемым разностными деревьями квадрантов (RQT). Таким образом, листовой элемент CU может включать в себя дерево квадрантов, указывающее, каким образом листовой элемент CU разделен на элементы TU. Корневой узел дерева квадрантов элемента TU обычно соответствует листовому элементу CU, в то время как корневой узел дерева квадрантов элемента CU обычно соответствует элементу CTU (или элементу LCU). Элементы TU дерева RQT, которые не являются разбитыми, называются листовыми элементами TU. В общем случае это раскрытие использует термины элемент CU и элемент TU для обозначения листового элемента CU и листового элемента TU, соответственно, если не указано иначе.

Видеопоследовательность, как правило, включает в себя последовательность видеокадров или изображений, начинающихся с изображения с точкой произвольного доступа (RAP).

Видеопоследовательность может включать в себя синтаксические данные в множестве параметров последовательности (SPS), которые являются характеристиками видеопоследовательности. Каждый слой изображения может включать в себя синтаксические данные слоя, которые описывают режим кодирования для соответствующего слоя. Видеокoder 20, как правило, работает с видеоблоками в индивидуальных видеослоях, чтобы закодировать видеоданные. Видеоблок может соответствовать узлу кодирования в элементе CU. Видеоблоки могут иметь фиксированные или переменные размеры и могут отличаться по размеру в соответствии с заданным стандартом кодирования.

В качестве примера предсказание может быть выполнено для элементов PU различных размеров. В предположении, что размер конкретного элемента CU составляет $2N \times 2N$, интра-предсказание может быть выполнено на размерах элемента PU $2N \times 2N$ или $N \times N$, и интерпредсказание может быть выполнено на размерах симметричных элементов PU $2N \times 2N$, $2N \times N$, $N \times 2N$ или $N \times N$. Асимметричное разбиение для интер-предсказания также может быть выполнено для размеров элемента PU $2N \times nU$, $2N \times nD$, $nL \times 2N$ и $nR \times 2N$. При асимметричном разбиении одно направление элемента CU не разбивается, в то время как

другое направление разбивается на 25 и 75%. Часть элемента CU, соответствующая 25% разделению, обозначена как p, за которым следует показатель U (вверх), D (вниз), L (влево) или R (вправо). Таким образом, например, $2N \times pU$ относится к элементу CU с размером $2N \times 2N$, который разбит горизонтально на элемент PU с размером $2N \times 0,5N$ сверху и элемент PU с размером $2N \times 1,5N$ снизу.

В этом раскрытии " $N \times N$ " и " N на N " могут использоваться взаимозаменяемым образом для обозначения размерностей пикселей видеоблока с точки зрения вертикальной и горизонтальной размерностей, например 16×16 пикселей или 16 на 16 пикселей. В общем случае блок с размером 16×16 будет иметь 16 пикселей в вертикальном направлении ($y=16$) и 16 пикселей в горизонтальном направлении ($x=16$). Аналогичным образом блок с размером $N \times N$ обычно имеет N пикселей в вертикальном направлении и N пикселей в горизонтальном направлении, где N представляет неотрицательное целочисленное значение. Пиксели в блоке могут быть расположены в строках и столбцах. Кроме того, блоки не обязательно должны иметь одинаковое количество пикселей в горизонтальном направлении и в вертикальном направлении. Например, блоки могут содержать $N \times M$ пикселей, где M не обязательно равно N .

После кодирования с интра-предсказанием или интерпредсказанием с использованием элементов PU элемента CU видеокодер 20 может вычислить разностные данные для элементов TU элемента CU. Элементы PU могут содержать синтаксические данные, описывающие способ или режим формирования пиксельных данных предсказания в пространственной области (также называемой пиксельной областью), и элементы TU могут содержать коэффициенты в области преобразования после применения преобразования, например дискретного косинусного преобразования (DCT), целочисленного преобразования, вейвлетного преобразования или концептуально подобного преобразования к разностным видеоданным. Разностные данные могут соответствовать разностям между пикселями незакодированного изображения и значениями предсказания, соответствующими элементам PU. Видеокодер 20 может сформировать элементы TU, включающие в себя квантованные коэффициенты преобразования, представляющие разностные данные для элемента CU. Таким образом, видеокодер 20 может вычислить разностные данные (в форме разностного блока), преобразовать разностный блок, чтобы произвести блок коэффициентов преобразования, и затем квантовать коэффициенты преобразования, чтобы сформировать квантованные коэффициенты преобразования. Видеокодер 20 может сформировать элемент TU, включающий в себя квантованные коэффициенты преобразования, а также другую синтаксическую информацию (например, информацию разбиения для элемента TU).

Как отмечено выше, после любых преобразований, чтобы произвести коэффициенты преобразования, видеокодер 20 может выполнить квантование коэффициентов преобразования. Квантование обычно относится к процессу, в котором коэффициенты преобразования квантуются для возможного сокращения объема данных, используемых для представления коэффициентов, обеспечивая дополнительное сжатие. Процесс квантования может сократить битовую глубину, связанную с некоторыми или всеми коэффициентами. Например, во время квантования n -битовое значение может быть округлено в меньшую сторону до m -битового значения, где n больше m .

После квантования видеокодер может просканировать коэффициенты преобразования, производя одномерный вектор из двумерной матрицы, включающей в себя квантованные коэффициенты преобразования. Сканирование может быть разработано таким образом, чтобы помещать коэффициенты с более высокой энергией (и поэтому более низкой частотой) впереди массива и помещать коэффициенты с более низкой энергией (и поэтому более высокой частотой) позади массива. В некоторых примерах видеокодер 20 может использовать predetermined порядок сканирования для сканирования квантованных коэффициентов преобразования для получения сериализованного вектора, который может быть статистически закодирован. В других примерах видеокодер 20 может выполнить адаптивное сканирование. После сканирования квантованных коэффициентов преобразования для формирования одномерного вектора видеокодер 20 может статистически закодировать одномерный вектор, например, в соответствии с контекстно-адаптивным кодированием с переменной длиной (CAVLC), контекстно-адаптивным двоичным арифметическим кодированием (CABAC), основанным на синтаксисе контекстно-адаптивным двоичным арифметическим кодированием (SBAC), статистическим кодированием с разделением на интервалы вероятности (PIPE) или другой методологией статистического кодирования. Видеокодер 20 также может выполнять статистическое кодирование синтаксических элементов, соответствующих закодированным видеоданным, для использования видеодекодером 30 при декодировании видеоданных.

Для выполнения кодирования CABAC видеокодер 20 может присвоить контекст в пределах контекстной модели символу, который должен быть передан. Контекст может относиться, например, к тому, являются ли соседние значения символа ненулевыми. Для выполнения кодирования CAVLC видеокодер 20 может выбрать неравномерный код для символа, который должен быть передан. Кодовые слова в кодировании VLC могут быть созданы таким образом, что относительно более короткие коды соответствуют более вероятным символам, в то время как более длинные коды соответствуют менее вероятным символам. Таким образом, использование кодирования VLC может достигнуть экономии битов, например, с использованием кодовых слов равной длины для каждого символа, который должен быть передан. Определение вероятности может быть основано на контексте, присвоенном символу.

В целом видеodeкодер 30 выполняет в значительной степени подобный, хотя и взаимно-обратный, процесс относительно процесса, выполняемого видеodeкодером 20, чтобы декодировать закодированные данные. Например, видеodeкодер 30 выполняет обратное квантование и обратное преобразование коэффициентов принятого элемента ТУ, чтобы воспроизвести разностный блок. Видеodeкодер 30 использует сообщенный режим предсказания (интра- или интер-предсказание, или режим на основе палитры), чтобы сформировать предсказанный блок. Затем видеodeкодер 30 объединяет предсказанный блок и разностный блок (на попиксельной основе), чтобы воспроизвести первоначальный блок. Может быть выполнена дополнительная обработка, такая как выполнение процесса удаления блочности для сокращения визуальных артефактов вдоль границ блока. Кроме того, видеodeкодер 30 может декодировать синтаксические элементы с использованием кодирования САВАС методом, в значительной степени подобным, хотя и взаимно-обратным, процессу кодирования САВАС видеodeкодера 20.

Видеodeкодер 20 может затем отправить синтаксические данные, такие как синтаксические данные на основе блока, синтаксические данные на основе изображения и синтаксические данные на основе последовательности, видеodeкодеру 30, например, в заголовке изображения, заголовке блока, заголовке слоя или других синтаксических данных, таких как множество параметров последовательности (SPS), множество параметров изображения (PPS) или множество видеопараметров (VPS). Видеodeкодер 30 может использовать эти параметры, чтобы декодировать видеоданные.

В соответствии с методиками этого раскрытия видеodeкодер 20 может выполнить следующий процесс, для квантования значения пикселя вне палитры во время кодирования в режиме на основе палитры:

1) Видеodeкодер 20 получает параметр qP квантования в соответствии со значениями индекса ($cIdx$) другого цветового компонента следующим образом:

$$qP = (cIdx == 0) ? Qp'Y : ((cIdx == 1) ? Qp'Cb : Qp'Cr)$$

2) Видеodeкодер 20 получает отношение $qPper$ квантования и остаток $qPrem$ квантования следующим образом:

$$qPper = qP / 6$$

$$qPrem = qP \% 6$$

3) Видеodeкодер 20 получает параметр $quantRightShift$ сдвига вправо и параметр $Offset$ сдвига следующим образом:

$$quantRightShift = 14 + qPper$$

$$Offset = 1 \ll (quantRightShift - 1)$$

4) Квантованное значение пикселя вне палитры выводится следующим образом

$$EspValue = (pixelValue * quantScale[qPrem] + Offset) >> quantRightShift$$

Аналогично, как взаимно-обратный процесс, видеodeкодер 30 может выполнить следующий процесс для обратного квантования (или деквантования) значения пикселя вне палитры во время кодирования в режиме на основе палитры:

1) Видеodeкодер 30 получает параметр квантования qP в соответствии со значениями индекса ($cIdx$) другого цветового компонента следующим образом:

$$qP = (cIdx == 0) ? Qp'Y : ((cIdx == 1) ? Qp'Cb : Qp'Cr)$$

2) Видеodeкодер 30 выводит отношение $qPper$ квантования и остаток $qPrem$ квантования следующим образом:

$$qPper = qP / 6$$

$$qPrem = qP \% 6$$

3) Видеodeкодер 30 выводит параметр $invQuantRightShift$ сдвига вправо и параметр $addOffset$ смещения следующим образом:

$$invQuantRightShift = 6 - qPper$$

$$addOffset = invQuantRightShift == 0 ? 0 : 1 \ll (invQuantRightShift - 1)$$

4A) Если $invQuantRightShift$ больше 0, видеodeкодер 30 выполняет следующую процедуру:

$$deQuantEspValue = (EspValue * invQuantScale[qPrem] + addOffset) >> invQuantRightShift$$

4B) Иначе ($invQuantRightShift \leq 0$), видеodeкодер 30 выполняет следующую процедуру:

$$invQuantLeftShift = -invQuantRightShift$$

$$deQuantEspValue = EspValue * invQuantScale[qPrem] \ll invQuantLeftShift$$

5) Видеodeкодер 30 затем модифицирует значение $deQuantEspValue$, чтобы оно находилось в диапазоне глубины следующим образом:

$$deQuantEspValue = clip3(0, 2^{bitDepth} - 1, deQuantEspValue)$$

$quantScale$ [.] и $invQuantScale$ [.] могут представлять собой таблицы поиска (например, реализованные как массивы), которые могут быть равны {26214, 23302, 20560, 18396, 16384, 14564}, {40, 45, 51, 57, 64, 72} соответственно, или другую таблицу поиска (или массив) из 6 записей, когда разрешено адаптив-

ное масштабирование квантования. Например, когда разрешено адаптивное масштабирование квантования, видеодекoder 30 может скорректировать `invQuantRightShift` и/или `invQuantLeftShift`, чтобы нормализовать масштабный коэффициент, введенный посредством адаптивного масштабирования квантования.

Видеодекoder 20 и видеодекoder 30 в соответствующих случаях могут быть реализованы как любая из множества подходящих схем кодера или декодера, например один или несколько микропроцессоров, процессоров цифровых сигналов (DSP), специализированных интегральных схем (ASIC), программируемых пользователем вентильных матриц (FPGA), дискретных логических схем, программного обеспечения, аппаратных средств, программируемого оборудования или любой их комбинации. Каждый из видеодекодера 20 и видеодекодера 30 может быть включен в один или несколько кодеров или декодеров, любой из которых может быть интегрирован как часть объединенного видеодекодера/видеодекодера (кодека; CODEC). Устройство, включающее в себя видеодекoder 20 и/или видеодекoder 30, может содержать интегральную схему, микропроцессор и/или устройство беспроводной связи, такое как мобильный телефон.

Фиг. 2 является блок-схемой, иллюстрирующей пример видеодекодера 20, который может реализовать методики для кодирования пикселей вне палитры в блоках, кодируемых в режиме на основе палитры, в соответствии с методиками этого раскрытия. Видеодекoder 20 может выполнять интра- и интер-кодирование видеоблоков в видеослоях. Интра-кодирование полагается на пространственное предсказание для сокращения или удаления пространственной избыточности в видеоинформации в данном видеокадре или изображении. Интер-кодирование полагается на временное предсказание для сокращения или удаления временной избыточности в видеоинформации в смежных кадрах или изображениях видеопоследовательности. Режим интра-предсказания (режим I) может относиться к любому из нескольких пространственных режимов кодирования. Режимы интерпредсказания, такие как однонаправленное предсказание (режим P) или двунаправленное предсказание (режим B), могут относиться к любому из нескольких временных режимов кодирования.

Как показано на фиг. 2, видеодекoder 20 принимает текущий видеоблок в видеокадре, который будет закодирован. В примере на фиг. 2 видеодекoder 20 включает в себя модуль 40 выбора режима, память 64 опорных изображений (которая также может упоминаться как буфер декодированных изображений (DPB)), сумматор 50, модуль 52 обработки преобразования, модуль 54 квантования и модуль 56 энтропийного кодирования. Модуль 40 выбора режима, в свою очередь, включает в себя модуль 44 компенсации движения, модуль 42 оценки движения, модуль 46 интра-предсказания, модуль 49 кодирования режима на основе палитры и модуль 48 разделения. Для воссоздания видеоблоков видеодекoder 20 также включает в себя модуль 58 обратного квантования, модуль 60 обратного преобразования и сумматор 62. Фильтр удаления блочности (не показан на фиг. 2) также может быть включен в состав для фильтрации границ блока, чтобы удалить артефакты блочности из воссозданной видеоинформации. При желании фильтр удаления блочности, как правило, выполняет фильтрацию выходной информации сумматора 62. Дополнительные фильтры (в цикле или после цикла) также могут использоваться в дополнение к фильтру удаления блочности. Такие фильтры для краткости не показаны, но при желании могут выполнять фильтрацию выходной информации сумматора 50 (как в фильтре в цикле).

Во время процесса кодирования видеодекoder 20 принимает видеокадр или слой, который будет закодирован. Кадр или слой могут быть разделены на несколько видеоблоков. Модуль 42 оценки движения и модуль 44 компенсации движения выполняют кодирование с интерпредсказанием принятого видеоблока относительно одного или более блоков в одном или более опорных кадрах, чтобы обеспечить временное предсказание. Модуль 46 интра-предсказания в качестве альтернативы может выполнить кодирование с интра-предсказанием принятого видеоблока относительно одного или более соседних блоков в том же кадре или слое, где находится блок, который будет закодирован, чтобы обеспечить пространственное предсказание. Видеодекoder 20 может выполнить несколько проходов кодирования, например, чтобы выбрать подходящий режим кодирования для каждого блока видеоданных.

Кроме того, модуль 48 разделения может разделить блоки видеоданных на подблоки на основе оценки предыдущих схем разделения на предыдущих проходах кодирования. Например, модуль 48 разделения может первоначально разделить кадр или слой на элементы STU и разделить каждый из элементов STU на подэлементы CU на основе анализа отношения скорости передачи и искажений (например, оптимизация отношения скорости передачи и искажений). Модуль 40 выбора режима затем может проинформировать структуру данных дерева квадрантов, указывающую разделение элемента STU на подэлементы CU. Элементы CU листовых узлов дерева квадрантов могут включать в себя один или более элементов PU и один или более элементов TU.

Модуль 40 выбора режима может выбрать один из режимов предсказания, режим интра- или интер-предсказания, например, на основе результатов ошибок и предоставляет полученный в результате предсказанный блок сумматору 50 для формирования разностных данных и сумматору 62 для воссоздания закодированного блока для использования в качестве опорного кадра. Кроме того, модуль 40 выбора режима может выбрать режим на основе палитры как альтернативу режимам интра- и интер-предсказания. Модуль 40 выбора режима также обеспечивает синтаксические элементы, такие как векторы движения, индикаторы режима интра-предсказания, информацию разделения и другую такую синтаксическую информацию модулю 56 энтропийного кодирования.

Модуль 42 оценки движения и модуль 44 компенсации движения могут быть в значительной степени интегрированы, но проиллюстрированы отдельно в концептуальных целях. Оценка движения, выполненная модулем 42 оценки движения, является процессом формирования векторов движения, которые оценивают движение для видеоблоков. Вектор движения, например, может указывать смещение элемента PU видеоблока в текущем видеокадре или изображении относительно блока предсказания в опорном кадре (или другом закодированном элементе) относительно текущего блока, закодированного в текущем кадре (или другого закодированного элемента). Блок предсказания представляет собой блок, который оказывается близко сопоставимым блоком, который должен быть закодирован, с точки зрения различия пикселей, что может быть определено посредством суммы абсолютной разности (SAD), суммы разности квадратов (SSD) или других метрик различия. В некоторых примерах видеокодер 20 может вычислить значения для дробных позиций пикселя опорных изображений, сохраненных в памяти 64 опорных изображений. Например, видеокодер 20 может интерполировать значения позиций одной четверти пикселя, позиций одной восьмой пикселя или других дробных позиций пикселя опорного изображения. Таким образом, модуль 42 оценки движения может выполнить поиск движения относительно позиций целых пикселей и позиций дробных пикселей и выдать вектор движения с точностью до дробного пикселя.

Модуль 42 оценки движения вычисляет вектор движения для элемента PU видеоблока в интеркодированной секции, сравнивая позицию элемента PU с позицией блока предсказания опорного изображения. Опорное изображение может быть выбрано из первого списка опорных изображений (список 0) или второго списка опорных изображений (список 1), каждый из которых идентифицирует одно или несколько опорных изображений, сохраненных в памяти 64 опорных изображений. Модуль 42 оценки движения отправляет вычисленный вектор движения модулю 56 статистического кодирования и модулю 44 компенсации движения.

Компенсация движения, выполненная модулем 44 компенсации движения, может включать в себя выборку или формирование блока предсказания на основе вектора движения, определенного модулем 42 оценки движения. Снова в некоторых примерах модуль 42 оценки движения и модуль 44 компенсации движения могут быть функционально объединены. После приема вектора движения для элемента PU текущего видеоблока модуль 44 компенсации движения может определить местоположение блока предсказания, на который указывает вектор движения, в одном из списков опорных изображений. Сумматор 50 формирует разностный видеоблок, вычитая значения пикселей блока предсказания из значений пикселей текущего закодированного видеоблока, формируя значения разности пикселей, как пояснено ниже. В целом модуль 42 оценки движения выполняет оценку движения относительно компонентов яркости, и модуль 44 компенсации движения использует векторы движения, вычисленные на основе компонентов яркости, и для компонентов цветности, и для компонентов яркости. Модуль 40 выбора режима также может формировать синтаксические элементы, связанные с видеоблоками и видеослоем для использования видеодекодером 30 при декодировании видеоблоков видеослоя.

Модуль 46 интра-предсказания может выполнить интра-предсказание текущего блока в качестве альтернативы интер-предсказанию, выполненному модулем 42 оценки движения и модулем 44 компенсации движения, как описано выше. В частности, модуль 46 интра-предсказания может определить режим интра-предсказания для использования, чтобы закодировать текущий блок. В некоторых примерах модуль 46 интра-предсказания может закодировать текущий блок с использованием различных режимов интра-предсказания, например, во время отдельных проходов кодирования, и модуль 46 интра-предсказания (или в некоторых примерах модуль 40 выбора режима) может выбрать из испытанных режимов подходящий режим интра-предсказания для использования.

Например, модуль 46 интра-предсказания может вычислить значения отношения скорости передачи и искажений с использованием анализа отношения скорости передачи и искажений для различных испытанных режимов интра-предсказания и выбрать режим интра-предсказания, имеющий наилучшие характеристики отношения скорости передачи и искажения среди испытанных режимов. Анализ отношения скорости передачи и искажений обычно определяет величину искажения (или ошибки) между закодированным блоком и первоначальным незакодированным блоком, который был закодирован, чтобы произвести закодированный блок, а также скорости передачи (то есть, количества биты), использованных для производства закодированного блока. Модуль 46 интра-предсказания может вычислить отношения из искажений и скоростей передачи для различных закодированных блоков, чтобы определить, какой режим интра-предсказания демонстрирует наилучшее значение отношения скорости передачи и искажений для блока.

После выбора режима интра-предсказания для блока модуль 46 интра-предсказания может предоставить информацию, являющуюся показателем выбранного режима интра-предсказания для блока, модулю 56 энтропийного кодирования. Модуль 56 энтропийного кодирования может закодировать информацию, указывающую выбранный режим интра-предсказания. Видеокодер 20 может включить в переданный битовый поток данные конфигурации, которые могут включать в себя множество индексных таблиц режима интра-предсказания и множество модифицированных индексных таблиц режима интра-предсказания (также называемых таблицами отображения кодовых слов), определения контекстов кодирования для различных блоков и показатели относительно наиболее вероятного режима интра-

предсказания, индексной таблицы режима интра-предсказания и модифицированной индексной таблицы режима интра-предсказания для использования для каждого из контекстов.

Видеокодер 20 формирует разностный видеоблок, вычитая данные предсказания от модуля 40 выбора режима из первоначального закодированного видеоблока. Сумматор 50 представляет компонент или компоненты, которые выполняют эту операцию вычитания. Модуль 52 обработки преобразования применяет преобразование, такое как дискретное косинусное преобразование (DCT) или концептуально подобное преобразование к разностному блоку, производя видеоблок, содержащий значения коэффициентов преобразования. Вейвлетные преобразования, целочисленные преобразования, преобразования поддиапазонов, дискретные синусные преобразования (DST) или другие типы преобразований могут использоваться вместо DCT. В любом случае модуль 52 обработки преобразования применяет преобразование к разностному блоку, производя блок коэффициентов преобразования. Преобразование может преобразовать разностную информацию из области пикселей в область преобразования, такую как частотная область. Модуль 52 обработки преобразования может отправить полученные в результате коэффициенты преобразования модулю 54 квантования. Модуль 54 квантования квантует коэффициенты преобразования для дополнительного сокращения битрейта. Процесс квантования может сократить битовую глубину, связанную с некоторыми или всеми коэффициентами. Степень квантования может быть модифицирована посредством корректировки параметра квантования.

После квантования модуль 56 энтропийного кодирования выполняет энтропийное кодирование квантованных коэффициентов преобразования. Например, модуль 56 энтропийного кодирования может выполнить контекстно-адаптивное кодирование переменной длины (CAVLC), контекстно-адаптивное двоичное арифметическое кодирование (CABAC), основанное на синтаксисе контекстно-адаптивное двоичное арифметическое кодирование (SBAC), энтропийное кодирование с разделением на интервалы вероятности (PIPE) или другую методику энтропийного кодирования. В случае основанного на контексте энтропийного кодирования контекст может быть основан на соседних блоках. После энтропийного кодирования модулем 56 энтропийного кодирования закодированный битовый поток может быть передан другому устройству (например, видеодекодеру 30) или архивирован для более поздней передачи или извлечения.

Модуль 58 обратного квантования и модуль 60 обратного преобразования применяют обратное квантование и обратное преобразование, соответственно, чтобы воссоздать разностный блок в области пикселей. В частности, сумматор 62 добавляет воссозданный разностный блок к блоку предсказания с компенсацией движения, ранее произведенному модулем 44 компенсации движения или модулем 46 интра-предсказания, чтобы произвести воссозданный видеоблок для сохранения в памяти 64 опорных изображений. Воссозданный видеоблок может использоваться модулем 42 оценки движения и модулем 44 компенсации движения как опорный блок для интер-кодирования блока в последующем видеокадре.

В соответствии с методиками этого раскрытия видеокодер 20 может быть выполнен с возможностью выполнять кодирование на основе палитры. В частности, модуль 49 кодирования на основе палитры может выполнять кодирование в режиме на основе палитры блока (например, элемента CU или элемента PU) видеоданных. Относительно структуры стандарта HEVC в качестве примера методики кодирования на основе палитры могут быть выполнены для использования в качестве режима элемента CU. В других примерах методики кодирования на основе палитры могут быть выполнены для использования в качестве режима элемента PU в структуре стандарта HEVC. В соответствии с этим все раскрытые процессы, описанные в настоящем документе (во всем этом раскрытии) в контексте режима элемента CU, дополнительно или в качестве альтернативы могут относиться к режиму элемента PU. Однако эти основанные на стандарте HEVC примеры нельзя считать ограничением методик кодирования на основе палитры, описанных в настоящем документе, поскольку методики могут быть применены для независимой работы или как часть других существующих или еще ожидающих разработки систем/стандартов. В этих случаях элемент для кодирования на основе палитры может представлять собой квадратные блоки, прямоугольные блоки или даже регионы непрямоугольной формы.

Модуль 49 кодирования на основе палитры, например, может выполнять кодирование на основе палитры, когда режим кодирования на основе палитры, например, выбран для элемента CU или элемента PU. Например, модуль 49 кодирования на основе палитры может быть выполнен с возможностью формировать палитру, имеющую записи, указывающие значения пикселей, выбирать значения пикселей в палитре для представления значений пикселей, по меньшей мере, некоторых позиций блока видеоданных и сигнализировать информацию, привязывающую, по меньшей мере, некоторые позиции блока видеоданных к записям в палитре, соответствующей выбранным значениям пикселей. Хотя различные функции описаны как выполняемые модулем 49 кодирования на основе палитры, некоторые или все такие функции могут быть выполнены другими модулями обработки или комбинацией разных модулей обработки.

В соответствии с одной или более методиками этого раскрытия видеокодер 20, а более конкретно модуль 49 кодирования на основе палитры, может выполнить видеокодирование на основе палитры предсказанных видеоблоков. Как описано выше, палитра, сформированная видеокодером 20, может быть явно закодирована, предсказана на основе предыдущих записей палитры, предсказана на основе преды-

дущих значений пикселей или представлять собой комбинацию этого.

В частности, модуль 40 выбора режима может определить режим кодирования для блока (например, элемента CU или элемента PU) видеоданных, такой как интер-предсказание, интра-предсказание или режим на основе палитры. В предположении выбранного режима на основе палитры модуль 49 кодирования на основе палитры может сформировать палитру для блока на основе статистики значений пикселей для блока. Для каждого пикселя блока модуль 49 кодирования на основе палитры может определить, имеет ли пиксель соответствующее значение в палитре, и если имеет, сигнализировать индекс в палитре для соответствующего значения для пикселя. Модуль 49 кодирования на основе палитры может также сигнализировать значение длины серии, представляющие количество пикселей, имеющих такое же значение, как предыдущий пиксель.

В качестве альтернативы, если последовательность пикселей имеет значения, равные соседним пикселям сверху, модуль 49 кодирования на основе палитры может сигнализировать значение длины серии для режима "копирования сверху", где длина серии представляет количество пикселей, имеющих значения, равные соседним пикселям сверху.

Если ни индексный режим, ни режим копирования сверху адекватно не представляют значение текущего пикселя блока, кодируемого в режиме на основе палитры, модуль 49 кодирования на основе палитры может использовать методики этого раскрытия, чтобы закодировать пиксель как пиксель вне палитры. Таким образом, эти методики могут выполняться для квантования значения пикселя вне палитры, причем значение пикселя вне палитры представляет фактическое значение кодируемого пикселя (например, значения яркости и/или цветности). В соответствии с методиками этого раскрытия видеокодер 20 может квантовать значение кодируемого пикселя как пиксель вне палитры, например, как пояснено выше относительно фиг. 1. Таким образом, модуль 49 кодирования на основе палитры может выполнить следующий процесс квантования:

1) Модуль 49 кодирования на основе палитры выводит qP в соответствии с индексом $cIdx$ другого цветового компонента следующим образом:

$$qP = (cIdx == 0) ? Qp'Y : ((cIdx == 1) ? Qp'Cb : Qp'Cr)$$

2) Модуль 49 кодирования на основе палитры выводит отношение $qPper$ квантования и остаток $qPrem$ квантования следующим образом:

$$qPper = qP / 6$$

$$qPrem = qP \% 6$$

3) Модуль 49 кодирования на основе палитры выводит параметр сдвига вправо $invQuantRightShift$ и параметр $addOffset$ смещения следующим образом:

$$\begin{aligned} invQuantRightShift &= 6 - qPper \\ addOffset &= invQuantRightShift == 0 ? 0 : 1 << (invQuantRightShift \\ &\quad - 1) \end{aligned}$$

4) Модуль 49 кодирования на основе палитры выводит деквантованный пиксель вне палитры $deQuantEspValue$ на основе декодированного посредством энтропийного декодирования $EspValue$ следующим образом:

$$\begin{aligned} deQuantEspValue &= (EspValue * invQuantScale[qPrem] + addOffset \\ &\quad) >> invQuantRightShift \end{aligned}$$

5) Модуль 49 кодирования на основе палитры затем модифицирует значение $deQuantEspValue$, чтобы оно находилось в диапазоне глубины следующим образом:

$$deQuantEspValue = clip3(0, 2^{bitDepth}-1, deQuantEspValue)$$

В этом примере $EspValue$ представляет квантованное значение вне палитры для пикселя, имеющего исходное значение, представленное как $pixelValue$.

Таким образом, видеокодер 20, показанный на фиг. 2, представляет пример видеокодера, выполненного с возможностью определять, что значение для параметра сдвига влево для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что значение для параметра сдвига влево меньше нуля, устанавливать значение для параметра сдвига вправо равным положительному значению, имеющему абсолютное значение, равное абсолютному значению параметра сдвига влево; и выполнять квантование пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

Фиг. 3 является блок-схемой, иллюстрирующей пример видеодекодера 30, который может реализовать методики для декодирования пикселей вне палитры в блоках, кодируемых в режиме на основе палитры, в соответствии с методиками этого раскрытия. В примере на фиг. 3 видеодекодер 30 включает в себя модуль 70 энтропийного декодирования, модуль 72 компенсации движения, модуль 74 интра-предсказания, модуль 75 декодирования на основе палитры, модуль 76 обратного квантования, модуль 78 обратного преобразования, память 82 опорных изображений и сумматор 80. Видеодекодер 30 в некоторых примерах может выполнять проход декодирования, в целом взаимно-обратный по отношению к проходу кодирования, описанному относительно видеокодера 20 (фиг. 2). Модуль 72 компенсации движения может формировать данные предсказания на основе векторов движения, принятых от модуля 70

энтропийного декодирования, в то время как модуль 74 интра-предсказания может формировать данные предсказания на основе индикаторов режима интра-предсказания, принятых от модуля 70 энтропийного декодирования.

Во время процесса декодирования видеодекoder 30 принимает закодированный видеопоток, который представляет видеоблоки закодированного видеослоя и соответствующие синтаксические элементы, от видеокodера 20. Модуль 70 энтропийного декодирования видеодекodера 30 выполняет энтропийное декодирование битового потока, чтобы сформировать квантованные коэффициенты, векторы движения или индикаторы режима интра-предсказания и другие синтаксические элементы. Модуль 70 энтропийного декодирования перенаправляет векторы движения и другие синтаксические элементы модулю 72 компенсации движения. Видеодекoder 30 может принять синтаксические элементы на уровне видеослоя и/или на уровне видеоблока.

Когда видеослой закодирован как интра-кодированный (I) слой, модуль 74 интра-предсказания может сформировать данные предсказания для видеоблока текущего видеослоя на основе сигнализированного режима интра-предсказания и данных из ранее декодированных блоков текущего кадра или изображения. Когда изображение закодировано как интер-кодированный (то есть, В или Р) слой, модуль 72 компенсации движения производит блоки предсказания для видеоблока текущего видеослоя на основе векторов движения и других синтаксических элементов, принятых от модуля 70 энтропийного декодирования. Блоки предсказания могут быть произведены из одного из опорных изображений в одном из списков опорных изображений. Видеодекoder 30 может построить списки опорных изображений, список 0 и список 1, с использованием методик построения по умолчанию на основе опорных изображений, сохраненных в памяти 82 опорных изображений. Модуль 72 компенсации движения определяет информацию предсказания для видеоблока текущего видеослоя посредством синтаксического анализа векторов движения и других синтаксических элементов и использует информацию предсказания, чтобы произвести блоки предсказания для текущего декодируемого видеоблока. Например, модуль 72 компенсации движения использует некоторые из принятых синтаксических элементов для определения режима предсказания (например, интра- или интер-предсказание), использованного для кодирования видеоблоков видеослоя, тип слоя интер-предсказания (например, В-слой или Р-слой), информации построения для одного или нескольких списков опорных изображений для слоя, векторов движения для каждого интер-кодированного видеоблока слоя, статуса интерпредсказания для каждого интер-кодированного видеоблока слоя и другой информации, чтобы декодировать видеоблоки в текущем видеослое.

Модуль 72 компенсации движения также может выполнить интерполяцию на основе фильтров интерполяции. Модуль 72 компенсации движения может использовать фильтры интерполяции, как они используются видеокodером 20 во время кодирования видеоблоков, чтобы вычислить интерполированные значения для дробных пикселей опорных блоков. В этом случае модуль 72 компенсации движения может определить фильтры интерполяции, используемые видеокodером 20, на основе принятых синтаксических элементов и использовать фильтры интерполяции, чтобы произвести блоки предсказания.

Модуль 76 обратного квантования выполняет обратное квантование квантованных коэффициентов преобразования, обеспеченные в битовом потоке и декодированных модулем 70 энтропийного декодирования. Процесс обратного квантования может включать в себя использование параметра QP_Y квантования, вычисленного видеокodером 30 для каждого видеоблока в видеослое, чтобы определить степень квантования и, аналогичным образом, степень обратного квантования, которое должно быть применено.

Модуль 78 обратного преобразования применяет обратное преобразование, например, обратное преобразование DCT, обратное целочисленное преобразование или концептуально подобный процесс обратного преобразования к коэффициентам преобразования, чтобы произвести разностные блоки в области пикселей.

После того, как модуль 72 компенсации движения формирует блок предсказания для текущего видеоблока на основе векторов движения и других синтаксических элементов, видеодекoder 30 формирует декодированный видеоблок, суммируя разностные блоки от блока обратного преобразования 78 с соответствующими блоками предсказания, сформированными модулем 72 компенсации движения. Сумматор 80 представляет компонент или компоненты, которые выполняют эту операцию суммирования. При желании также может быть применен фильтр удаления блочности, чтобы отфильтровать декодированные блоки для удаления артефактов блочности. Другие петлевые фильтры (либо в цикле кодирования, либо после цикла кодирования) также могут использоваться, чтобы сгладить переходы пикселей или иным образом улучшить качество видеоизображения. Декодированные видеоблоки в данном кадре или изображении затем сохраняются в памяти 82 опорных изображений, которая хранит опорные изображения, используемые для последующей компенсации движения. Память 82 опорных изображений также хранит декодированную видеoinформацию для более позднего представления на устройстве отображения, такого как устройство 32 отображения на фиг. 1.

В соответствии с различными примерами этого раскрытия видеодекoder 30 может быть выполнен с возможностью выполнять декодирование на основе палитры. В частности, видеодекoder 30 включает в себя модуль 75 декодирования на основе палитры, который может выполнять декодирование на основе палитры. Например, модуль 75 декодирования на основе палитры может быть выполнен с возможностью

формировать палитру, имеющую записи, указывающие значения пикселей. Кроме того, в этом примере модуль 75 декодирования на основе палитры может принимать информацию, такую как синтаксические элементы, показанные на фиг. 3, привязывающие по меньшей мере некоторые позиции блока видеоданных к записям в палитре. В этом примере модуль 75 декодирования на основе палитры может выбрать значения пикселей в палитре на основе информации. Кроме того, в этом примере модуль 75 декодирования на основе палитры может воссоздать значения пикселей блока на основе выбранных значений пикселей. Хотя различные функции описаны как выполняемые модулем 75 декодирования на основе палитры, некоторые или все такие функции могут быть выполнены другими модулями обработки или комбинацией разных модулей обработки.

Модуль 75 декодирования на основе палитры может принять информацию режима кодирования на основе палитры и выполнить упомянутые выше операции, когда информация режима кодирования на основе палитры указывает, что режим кодирования на основе палитры применяется к блоку. Когда информация режима кодирования на основе палитры указывает, что режим кодирования на основе палитры не относится к блоку, или когда другая информация режима указывает использование другого режима, модуль 75 декодирования на основе палитры декодирует блок видеоданных с использованием режима кодирования не на основе палитры, например, такого как режим кодирования с интер-предсказанием или интра-предсказанием стандарта HEVC, когда информация о режиме кодирования на основе палитры указывает, что режим кодирования на основе палитры не применяется к блоку. Блок видеоданных может представлять собой, например, элемент CU или элемент PU, сформированный в соответствии с процессом кодирования HEVC. Видеодекодер 30 может декодировать некоторые блоки в режиме кодирования с временным интер-предсказанием или пространственным интра-предсказанием и декодировать другие блоки в режиме кодирования на основе палитры. Режим кодирования на основе палитры может содержать один из множества различных режимов кодирования на основе палитры или может являться единственным режимом кодирования на основе палитры.

В соответствии с одной или более методик этого раскрытия видеодекодер 30, и более конкретно модуль 75 декодирования на основе палитры, может выполнить декодирование видеoinформации на основе палитры закодированных на основе палитры видеоблоков. Как описано выше, палитра, декодируемая видеодекодером 30, может быть явно закодирована и сигнализирована, воссоздана видеодекодером 30 относительно принятого закодированного на основе палитры блока, предсказанного на основе предыдущих записей палитры, предсказанных на основе предыдущих значений пикселей, или их комбинации.

В частности, модуль 150 энтропийного декодирования может декодировать информацию, указывающую, что блок (например, элемент PU или элемент CU) видеоданных закодирован с использованием режима на основе палитры. Модуль 150 энтропийного декодирования также может декодировать информацию, представляющую, каким образом каждый пиксель блока закодирован с использованием режима на основе палитры. Например, модуль 150 энтропийного декодирования может декодировать индексное значение и значение длины серии как пару (индекс, длина серии). Индексное значение представляет запись палитры для блока, причем запись определяет значение пикселя, и значение длины серии указывает количество пикселей в дополнение к текущему пикселю, которые имеют то же самое значение.

Если индексное значение равно размеру палитры, модуль 150 энтропийного декодирования может определить, что текущий пиксель является пикселем вне палитры. Таким образом, модуль 150 энтропийного декодирования может выполнить энтропийное декодирование значения, представляющего пикселя вне палитры. В частности, модуль 150 энтропийного декодирования может декодировать квантованное значение пикселя вне палитры для пикселя вне палитры. Модуль 150 энтропийного декодирования затем может передать квантованное значение пикселя вне палитры модулю 75 декодирования на основе палитры.

В соответствии с методиками этого раскрытия модуль 75 декодирования на основе палитры может выполнить обратное квантование квантованного значения пикселя вне палитры. Чтобы выполнить обратное квантование квантованного значения пикселя вне палитры, модуль 75 декодирования на основе палитры может выполнить следующий процесс:

1) Блок 75 декодирования на основе палитры выводит параметр qP квантования в соответствии с индексом $cIdx$ другого цветового компонента следующим образом:

$$qP = (cIdx == 0) ? Qp'Y : ((cIdx == 1) ? Qp'Cb : Qp'Cr)$$

2) Модуль 75 декодирования на основе палитры выводит отношение qP_{per} квантования и остаток qP_{rem} квантования следующим образом:

$$qP_{per} = qP / 6$$

$$qP_{rem} = qP \% 6$$

3) Модуль 75 декодирования на основе палитры выводит параметр $invQuantRightShift$ сдвига вправо и параметр $addOffset$ смещения следующим образом:

$$invQuantRightShift = 6 - qP_{per}$$

$$addOffset = invQuantRightShift == 0 ? 0 : 1 << (invQuantRightShift$$

$$- 1)$$

4A) Если `invQuantRightShift` больше 0, модуль 75 декодирования на основе палитры выполняет следующую процедуру:

```
deQuantEspValue=( EspValue * invQuantScale[qPrem]+addOffset
)>> invQuantRightShift
```

4B) В ином случае (`invQuantRightShift <= 0`) модуль 75 декодирования на основе палитры выполняет следующую процедуру:

```
invQuantLeftShift=- invQuantRightShift
deQuantEspValue=EspValue * invQuantScale[qPrem] <<
invQuantLeftShift
```

5) Модуль 75 декодирования на основе палитры далее модифицирует значение `deQuantEspValue`, чтобы оно находилось в диапазоне глубины, следующим образом:

```
deQuantEspValue=clip3 (0, 2bitDepth-1, deQuantEspValue)
```

Таблица поиска `invQuantScale[.]` может быть реализована как массив, имеющий записи {40, 45, 51, 57, 64, 72}, или другая таблица поиска, например, из 6 записей, и может использоваться, когда разрешено адаптивное масштабирование квантования. Например, когда разрешено адаптивное масштабирование квантования, видеодекoder 30 может скорректировать `invQuantRightShift` и/или `invQuantLeftShift`, чтобы нормализовать масштабный коэффициент, введенный посредством адаптивного масштабирования квантования.

Таким образом, видеодекoder 30 на фиг. 3 представляет пример видеодекодера, выполненного с возможностью определять, что значение для параметра сдвига вправо для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что исходное значение для параметра сдвига вправо меньше нуля, устанавливать значение для параметра сдвига влево равным положительному значению, имеющему абсолютное значение, равное абсолютному значению исходного значения, и выполнять обратное квантование пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

Фиг. 4 является блок-схемой последовательности этапов, демонстрирующей иллюстративный способ кодирования блока видеоданных в соответствии с методиками этого раскрытия. Способ на фиг. 4 может быть выполнен видеодекoderом 20 и его компонентами (например, проиллюстрированными на фиг. 2).

В этом примере модуль 40 выбора режима первоначально принимает блок видеоданных (этап 100). Блок может представлять собой, например, элемент предсказания (PU) или элемент кодирования (CU). Затем модуль 40 выбора режима определяет режим кодирования для блока (этап 102). Например, модуль 40 выбора режима может испытать различные режимы кодирования и сравнить режимы с использованием процесса оптимизации отношения скорости передачи и искажений (RDO). Кроме того, модуль 40 выбора режима также может сравнить различные размеры блока и схемы разделения блока с использованием процесса RDO.

Модуль 40 выбора режима может выбрать режим интра- или интер-предсказания, в этом случае модуль 42 оценки движения и модуль 44 компенсации движения или модуль 46 интра-предсказания могут предсказать пиксели блока с использованием интра-предсказания или интер-предсказания (этап 104), соответственно, формируя предсказанный блок. Видеодекoder 20 затем может сформировать и обработать разностные значения блока (этап 106). Например, модуль 50 формирования разности может вычесть первоначальный блок из предсказанного блока на попиксельной основе, формируя разностный блок. Модуль 52 обработки преобразования затем может преобразовать разностный блок с использованием преобразования, такого как, например, преобразование DCT, формируя блок преобразования. Модуль 54 квантования затем может выполнить квантование коэффициентов преобразования блока преобразования и обеспечить квантованные коэффициенты преобразования модулю 56 энтропийного кодирования. Модуль 40 выбора режима также обеспечивает информацию, представляющую режим предсказания (например, интра/интер, выбранный режим интра-предсказания, если используется интра-предсказание, или параметры движения, если используется интер-предсказание) модулю 56 энтропийного кодирования. Таким образом, модуль 56 энтропийного кодирования выполняет энтропийное кодирование информации предсказания и разностных значений (т.е., квантованных коэффициентов преобразования) (этап 108).

В качестве альтернативы модуль 40 выбора режима может выбрать режим на основе палитры для кодирования блока, в этом случае модуль 49 кодирования на основе палитры анализирует статистику пикселей для блока (этап 110). Например, модуль 49 кодирования на основе палитры может определить часто используемые значения пикселей. Модуль 49 кодирования на основе палитры затем формирует палитру для блока на основе статистики (этап 112). Хотя не показано на фиг. 4, модуль 56 энтропийного кодирования может выполнить энтропийное кодирование данных для палитры. Например, палитра может быть закодирована с предсказанием относительно ранее использованной палитры, например, как описано выше относительно табл. 1 и 2.

Модуль 49 кодирования на основе палитры затем может просканировать пиксели блока (этап 114), чтобы определить, каким образом следует закодировать пиксели. Например, модуль 49 кодирования на основе палитры может определить, включено ли текущее значение пикселя в палитру. Если значение

пикселя включено в палитру, модуль 49 кодирования на основе палитры может обеспечить индекс из палитры, который соответствует значению пикселя, модулю 56 энтропийного кодирования, который может выполнить энтропийное кодирование индексного значения (этап 116). Кроме того, модуль 49 кодирования на основе палитры может определить количество пикселей в строке после предыдущего пикселя, которые имеют такое же значение, и обеспечить значение "длины серии" модулю 56 энтропийного кодирования, который может выполнить энтропийное кодирование значения длины серии (этап 118).

В качестве альтернативы, если текущий пиксель не имеет значения в палитре, модуль 49 кодирования на основе палитры может определить, совпадает ли значение для пикселя с соседним сверху значением пикселя. В некоторых примерах это определение может быть сделано до определения, соответствует ли значение пикселя значению в палитре. В любом случае, если текущий пиксель имеет значение, которое равно соседнему сверху значению пикселя, модуль 49 кодирования на основе палитры может обеспечить значение длины серии, описывающее количество пикселей, которые имеют значения, равные их соседним сверху пикселям, модулю 56 энтропийного кодирования, который может выполнить энтропийное кодирование значения длины серии (этап 120).

Если текущий пиксель не соответствует значению в палитре и не имеет значения, равного соседнему сверху пикселю, модуль 49 кодирования на основе палитры может закодировать пиксель как пиксель вне палитры. В частности, модуль 49 кодирования на основе палитры может обеспечить индексное значение, равное размеру палитры, модулю 56 энтропийного кодирования, который может выполнить энтропийное кодирование индексного значения как размера палитры (этап 122). Индексное значение, равное размеру палитры, может сигнализировать, что пиксель кодируется как пиксель вне палитры. Модуль 49 кодирования на основе палитры далее может обеспечить значение пикселя вне палитры модулю 56 энтропийного кодирования.

В соответствии с методиками этого раскрытия модуль 49 кодирования на основе палитры может выполнить квантование значения пикселя (этап 124). Квантование значения пикселя обычно может включать в себя битовую операцию сдвига. В частности, как описано выше, если модуль 49 кодирования на основе палитры во время квантования определяет, что параметр сдвига влево меньше или равен нулю, модуль 49 кодирования на основе палитры может вместо этого выполнить битовый сдвиг вправо с использованием параметра сдвига вправо, имеющего абсолютное значение, равное абсолютному значению параметра сдвига влево, но также имеющего положительное значение. Модуль 49 кодирования на основе палитры затем может обеспечить квантованное значение пикселя вне палитры модулю 56 энтропийного кодирования, который может выполнить энтропийное кодирование квантованного значения пикселя (этап 126).

Видеокодер 20 может выполнить этот процесс (например, один из последовательности этапов 116 и 118, этап 120 или последовательности этапов 122-126) для каждого пикселя блока, закодированного в режиме на основе палитры.

Таким образом, способ на фиг. 4 представляет пример способа кодирования видеоданных, включающий в себя определение, что значение для параметра сдвига влево для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что значение для параметра сдвига влево меньше нуля, установку значения для параметра сдвига вправо равным положительному значению, имеющему абсолютное значение, равное абсолютному значению параметра сдвига влево; и квантование пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

Фиг. 5 является блок-схемой последовательности этапов, демонстрирующей иллюстративный способ декодирования блока видеоданных в соответствии с методиками этого раскрытия. Способ на фиг. 5 может быть выполнен видеодекодером 30 и его компонентами (например, проиллюстрированными на фиг. 3).

Первоначально модуль 70 энтропийного декодирования может декодировать данные, указывающие режим кодирования для блока видеоданных (этап 150). Модуль 70 энтропийного декодирования может использовать эти данные, чтобы определить режим кодирования для блока (этап 152), например, один режим из интра-предсказания, интер-предсказания или режима на основе палитры.

В случае, когда режимом кодирования является интра-предсказание или интер-предсказание, модуль 70 энтропийного декодирования может декодировать информацию предсказания (например, параметры режима интра-предсказания или движения) и обеспечить информацию предсказания подходящему модулю из модуля 72 компенсации движения или модуля 74 интра-предсказания, чтобы выполнить интра-предсказание или интер-предсказание (этап 154). Например, модуль 74 интра-предсказания может использовать режим интра-предсказания, чтобы построить блок предсказания на основе соседних пикселей для блока. В качестве другого примера модуль 72 компенсации движения может использовать параметры движения, чтобы извлечь (и потенциально обработать, например, отфильтровать) опорный блок из ранее декодированного изображения из памяти 82 опорных изображений.

Кроме того, видеодекодер 30 может декодировать и обработать разностные значения блока (этап 156). Например, модуль 76 обратного квантования может выполнить обратное квантование квантованных коэффициентов преобразования и модуль 78 обратного преобразования выполнить обратное преоб-

разование коэффициентов преобразования, чтобы воссоздать разностный блок. Затем сумматор 80 может объединить разностные значения разностного блока и предсказанные значения предсказанного блока (этап 158), чтобы воссоздать первоначальный блок.

В качестве альтернативы модуль 70 энтропийного декодирования может определить, что режимом кодирования для блока является режим на основе палитры. В этом случае модуль 70 энтропийного декодирования может выполнить энтропийное декодирование данных для палитры блока, в то время как модуль 75 декодирования на основе палитры может декодировать палитру для блока (этап 160) с использованием декодированных посредством энтропийного декодирования данных. Как описано выше относительно табл. 1 и 2, палитра может быть закодирована с предсказанием относительно предыдущей палитры. Таким образом, модуль 70 энтропийного декодирования может обеспечить декодированные посредством энтропийного декодирования данные для палитры модулю 75 декодирования на основе палитры, который может воссоздать палитру для блока с использованием декодированных данных.

Модуль 70 энтропийного декодирования также может декодировать данные для пикселей блока (этап 164). Например, декодированные данные могут соответствовать индексному значению, которое меньше размера палитры. В этом случае модуль 70 энтропийного декодирования также может декодировать значение длины серии (этап 166) и обеспечить индекс и значение длины серии модулю 75 декодирования на основе палитры. Модуль 75 декодирования на основе палитры может установить значение пикселя и каждый из пикселей в серии равными значению пикселя палитры, которое соответствует индексному значению (этап 168).

В качестве другого примера декодированные данные могут представлять собой значение длины серии без индексного значения. Такое значение длины серии без индексного значения может указать количество пикселей, закодированных с использованием режима копирования сверху. В этом случае модуль 70 энтропийного декодирования может обеспечить значение длины серии модулю 75 декодирования на основе палитры, который может установить значения для каждого из пикселей в серии равными значениям соответствующих соседних сверху значений пикселей (этап 170).

В качестве другого примера декодированные данные могут представлять собой индексное значение, которое равно размеру палитры. В этом случае модуль 70 энтропийного декодирования может определить, что текущий пиксель закодирован как пиксель вне палитры. Таким образом, модуль 70 энтропийного декодирования может выполнить энтропийное декодирование квантованного значения для пикселя вне палитры (этап 172) и обеспечить квантованное значение модулю 75 декодирования на основе палитры. Модуль 75 декодирования на основе палитры, в свою очередь, может выполнить обратное квантование квантованного значения (174). В частности, модуль 75 декодирования на основе палитры может выполнить обратное квантование квантованного значения с использованием методики этого раскрытия, например, как описано более подробно выше и относительно фиг. 6 ниже.

Видеодекoder 30 может выполнить этот процесс (например, один из последовательности этапов 166 и 168, этапа 170 или последовательности этапов 172 и 174) для каждого пикселя блока, кодированного в режиме на основе палитры, тем самым декодируя блок, кодированный в режиме на основе палитры.

Таким образом, способ на фиг. 5 представляет пример способа, включающего в себя определение, что значение для параметра сдвига вправо для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что исходное значение для параметра сдвига вправо меньше нуля, установку значения для параметра сдвига влево равным положительному значению, имеющему абсолютное значение, равное абсолютному значению исходного значения; и выполнение обратного квантования пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

Фиг. 6 является блок-схемой последовательности этапов, демонстрирующей иллюстративную методику, посредством которой модуль 75 декодирования на основе палитры может выполнить обратное квантование квантованного значения пикселя вне палитры закодированного в режиме на основе палитры блока видеоданных в соответствии с методиками этого раскрытия. В целом способ фиг. 6 может соответствовать элементу 174 на фиг. 5. Таким образом, фиг. 6 представляет один пример способа обратного квантования квантованного значения пикселя вне палитры закодированного в режиме на основе палитры блока видеоданных. Этапы, показанные на фиг. 6, не обязательно должны выполняться в показанном порядке, и некоторые этапы могут быть выполнены параллельно.

В этом примере первоначально модуль 75 декодирования на основе палитры выводит параметр квантования (QP) для закодированного в режиме на основе палитры блока видеоданных (этап 200). Например, модуль 75 декодирования на основе палитры может определить параметр QP с использованием следующей формулы:

$$QP = (cIdx == 0) ? QP'Y : ((cIdx == 1) ? QP'Cb : QP'Cr)$$

В этой иллюстративной формуле cIdx представляет индекс контекста для блока видеоданных. Значение cIdx может быть установлено на основе того, является ли блок блоком яркости, блоком цветности синего или блоком цветности красного.

Модуль 75 декодирования на основе палитры может затем определить значение отношения параметра QP и значение остатка параметра QP (этап 202). Например, модуль 75 декодирования на основе

палитры может определить значение отношения параметра QP (qPper) и значение остатка параметра QP (qPrem) в соответствии со следующими формулами:

$$qPper = qP / 6$$

$$qPrem = qP \% 6$$

Модуль 75 декодирования на основе палитры затем может вывести значение параметра сдвига вправо (этап 204). Например, модуль 75 декодирования на основе палитры может вывести значение параметра сдвига вправо (invQuantRightShift) в соответствии со следующей формулой:

$$invQuantRightShift = 6 - qPper$$

Кроме того, модуль 75 декодирования на основе палитры может вывести значение параметра смещения (этап 206). Модуль 75 декодирования на основе палитры может вывести значение параметра смещения (addOffset) в соответствии со следующей формулой:

$$addOffset = invQuantRightShift == 0 ? 0 : 1 << (invQuantRightShift - 1)$$

Модуль 75 декодирования на основе палитры затем может определить, меньше или равно нулю значение параметра сдвига вправо (этап 208). Если нет (т.е., если значение параметра сдвига вправо больше нуля) (ветвь "нет" этапа 208), модуль 75 декодирования на основе палитры может вычислить деквантованное значение для пикселя вне палитры на основе квантованного значения для пикселя вне палитры с использованием значения параметра сдвига вправо и значения параметра смещения (этап 210). Например, модуль 75 декодирования на основе палитры может вычислить деквантованное значение (deQuantEspValue) в соответствии со следующей формулой:

$$deQuantEspValue = (EspValue * invQuantScale[qPrem] + addOffset) >> invQuantRightShift,$$

где EspValue представляет квантованное значение и invQuantScale[.] представляет таблицу поиска, например {40, 45, 51, 57, 64, 72}.

С другой стороны, если параметр сдвига вправо меньше или равен нулю (ветвь "да" на этапе 208), модуль 75 декодирования на основе палитры может вычислить значение параметра сдвига влево как положительное значение, равное абсолютному значению значения параметра сдвига вправо (этап 212). Модуль 75 декодирования на основе палитры затем может вычислить деквантованное значение на основе квантованного значения с использованием значения параметра сдвига влево (этап 214). Например, модуль 75 декодирования на основе палитры может вычислить деквантованное значение (deQuantEspValue) в соответствии со следующей формулой:

$$deQuantEspValue = EspValue * invQuantScale[qPrem] << invQuantLeftShift$$

Модуль 75 декодирования на основе палитры может затем округлить деквантованное значение (216). Например, модуль 75 декодирования на основе палитры может округлить деквантованное значение с использованием следующей приведенной в качестве примера операции отсечения:

$$deQuantEspValue = clip3(0, 2^{bitDepth}-1, deQuantEspValue),$$

где clip3 - функция, определенная в стандарте ITU-T H.265. В частности, стандарт H.265 определяет clip3 (x, y, z) следующим образом:

$$clip3(x, y, z) = \begin{cases} x; & z < x \\ y; & z > y \\ z; & \text{в ином случае} \end{cases}$$

В других примерах могут использоваться другие операции округления, чтобы гарантировать, что деквантованное значение пикселя вне палитры находится в соответствующем диапазоне глубины.

Таким образом, способ на фиг. 6 представляет пример способа, включающего в себя определение, что значение для параметра сдвига вправо для пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных меньше нуля; на основе того, что исходное значение для параметра сдвига вправо меньше нуля, установку значения для параметра сдвига влево равным положительному значению, имеющему абсолютное значение, равное абсолютному значению исходного значения; и выполнение обратного квантования пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево.

Следует понимать, что в зависимости от примера некоторые действия или события любого из описанных здесь способов могут быть выполнены в другой последовательности, могут быть добавлены, объединены или пропущены все вместе (например, не все описанные действия или события необходимы для исполнения методики на практике). Кроме того, в некоторых примерах действия или события могут быть выполнены одновременно, например, посредством многопоточной обработки, обработки с прерываниями или с помощью нескольких процессоров, а не последовательно.

В одном или нескольких примерах описанные функции могут быть реализованы в аппаратных средствах, программном обеспечении, программируемом оборудовании или любой их комбинации. Если функции реализованы в программном обеспечении, они могут быть сохранены или переданы как одна или несколько команд или код на машиночитаемом носителе и исполнены аппаратным блоком обработки. Машиночитаемые носители могут включать в себя машиночитаемые запоминающие носители, кото-

рые соответствуют материальным носителям, таким как носители данных, или средства связи, включающие в любую среду, которая обеспечивает возможность передачи компьютерной программы от одного места на другое, например, в соответствии с протоколом связи. Таким образом, машиночитаемые носители обычно могут соответствовать (1) материальным машиночитаемым запоминающим носителям, которые являются непереходными, или (2) среде связи, такой как сигнал или несущая. Запоминающие носители данных могут быть любыми доступными носителями, к которым могут получить доступ один или несколько компьютеров или один или несколько процессоров для извлечения команд, кода и/или структур данных для реализации методик, описанных в этом раскрытии. Компьютерный программный продукт может включать в себя машиночитаемый носитель.

В качестве примера, но без ограничения, такие машиночитаемые носители могут содержать оперативное запоминающее устройство (ОЗУ; RAM), постоянное запоминающее устройство (ПЗУ; ROM), электрически стираемое программируемое постоянное запоминающее устройство (ЭСППЗУ; EEPROM), компакт-диск (CD-ROM) или другую память на оптическом диске, память на магнитном диске или другие магнитные устройства хранения, флэш-память или любой другой носитель, который может использоваться для хранения желаемого программного кода в виде команд или структур данных и к которому может получить доступ компьютер. Кроме того, любое соединение правильно называть машиночитаемым носителем. Например, если команды переданы с веб-сайта, сервера или другого удаленного источника с использованием коаксиального кабеля, волоконно-оптического кабеля, витой пары, цифровой абонентской линии (DSL) или беспроводной технологии, такой как инфракрасное, радиочастотное и микроволновое излучение, то коаксиальный кабель, волоконно-оптический кабель, витая пара, DSL или беспроводные технологии, такие как инфракрасное, радиочастотное и микроволновое излучение, включены в определение носителя. Однако следует понимать, что машиночитаемые запоминающие носители данных не включают в себя соединения, несущие сигналы или другие переходные носители, но вместо этого направлены на непереходные, материальные носители. Используемый здесь термин "диск" включает в себя компакт-диск (CD), лазерный диск, оптический диск, цифровой универсальный диск (DVD), гибкий диск и диск Blu-ray, причем диски обычно воспроизводят данные магнитным способом или оптическим способом с помощью лазера. Комбинации упомянутого выше также должны быть включены в контекст машиночитаемых носителей.

Команды могут быть исполнены одним или несколькими процессорами, такими как один или несколько процессоров цифровых сигналов (DSP), микропроцессоров общего назначения, специализированных интегральных схем (ASIC), программируемых пользователем вентильных матриц (FPGA) или других эквивалентных интегрированных или дискретных логических схем. В соответствии с этим используемый здесь термин "процессор" может относиться к любой упомянутой структуре или любой другой структуре, подходящей для реализации описанных здесь методик. Кроме того, в некоторых аспектах описанная здесь функциональность может быть обеспечена в пределах специализированных аппаратных средств и/или программных модулей, сконфигурированных для кодирования и декодирования или встроенных в объединенный кодек. Кроме того, методики могут быть полностью реализованы в одной или нескольких схемах или логических элементах.

Методики этого раскрытия могут быть реализованы в широком разнообразии устройств, в том числе в беспроводном телефоне, интегральной схеме (IC) или множестве интегральных схем (например, в чипсете). Различные компоненты, модули или элементы описаны в этом раскрытии, чтобы подчеркнуть функциональные аспекты устройств, выполненных с возможностью выполнять раскрытые методики, но не обязательно требуют реализации разными аппаратными элементами. Вместо этого, как описано выше, различные элементы могут быть объединены в аппаратном модуле кодека или обеспечены посредством набора взаимодействующих аппаратных модулей, включающих в себя один или несколько процессоров, как описано выше, вместе с подходящим программным обеспечением и/или программируемым оборудованием.

Были описаны различные примеры. Эти и другие примеры содержатся в объеме следующей формулы изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ декодирования видеоданных для пикселей вне палитры в блоках, кодируемых в режиме на основе палитры, способ содержит этапы, на которых

принимают блок, кодируемый в режиме на основе палитры, видеоданных, содержащий пиксели, которые отображаются в цветовую палитру посредством индекса палитры и пикселя, кодируемого как пиксель вне палитры, который не отображается в цветовую палитру;

получают значение для параметра сдвига вправо для обратного квантования пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных;

устанавливают значение для параметра сдвига влево равным положительному значению и

выполняют обратное квантование пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево для сдвига влево значения пикселя, кодируемого как пиксель вне

палитры;

причем выполнение обратного квантования пикселя, кодируемого как пиксель вне палитры, содержит вычисление значения для пикселя, кодируемого как пиксель вне палитры (deQuantEspValue), в соответствии с формулой

$$\text{deQuantEspValue} = \text{EspValue} * \text{invQuantScale}[\text{qPrem}] \ll \text{invQuantLeftShift},$$

где EspValue представляет закодированное квантованное значение вне палитры, закодированное для пикселя, кодируемого как пиксель вне палитры, invQuantScale представляет массив значений шкалы обратного квантования, qPrem представляет значение остатка параметра квантования, invQuantLeftShift представляет значение параметра сдвига влево и \ll представляет битовый оператор сдвига влево.

2. Способ по п.1, дополнительно содержащий этап, на котором определяют значение для параметра сдвига вправо на основе значения параметра (qP) квантования для блока.

3. Способ по п.2, дополнительно содержащий этап, на котором определяют значение qP на основе индекса (cIdx) цветового компонента для цветового компонента, включающего в себя пиксель, кодируемый как пиксель вне палитры.

4. Способ по п.2, в котором определение значения для параметра сдвига вправо содержит этапы, на которых

вычисляют отношение (qPper) квантования в соответствии с формулой $\text{qPper} = \text{qP}/6$, где qP представляет значение qP ; и

вычисляют значения для параметра ($\text{invQuantRightShift}$) сдвига вправо в соответствии с формулой $\text{invQuantRightShift} = 6 - \text{qPper}$.

5. Способ по п.1, в котором invQuantScale содержит {40, 45, 51, 57, 64, 72}.

6. Способ по п.1, дополнительно содержащий этапы, на которых декодируют палитру для блока, кодируемого в режиме на основе палитры; декодируют по меньшей мере один другой пиксель блока с использованием палитры и воссоздают блок с использованием пикселя, кодируемого как пиксель вне палитры, и декодированного по меньшей мере одного другого пикселя.

7. Устройство для декодирования видеоданных для пикселей вне палитры в блоках, кодируемых в режиме на основе палитры, устройство содержит

средство для приема блока, кодируемого в режиме на основе палитры, видеоданных, содержащего пиксели, которые отображаются в цветовую палитру посредством индекса палитры и пикселя, кодируемого как пиксель вне палитры, который не отображается в цветовую палитру;

средство для получения значения для параметра сдвига вправо для обратного квантования пикселя, кодируемого как пиксель вне палитры, блока, кодируемого в режиме на основе палитры, видеоданных;

средство для установки значения для параметра сдвига влево равным положительному значению и

средство для выполнения обратного квантования пикселя, кодируемого как пиксель вне палитры, с использованием значения параметра сдвига влево для сдвига влево значения пикселя, кодируемого как пиксель вне палитры;

причем средство для выполнения обратного квантования пикселя, кодируемого как пиксель вне палитры, содержит средство для вычисления значения для пикселя, кодируемого как пиксель вне палитры (deQuantEspValue), в соответствии с формулой

$$\text{deQuantEspValue} = \text{EspValue} * \text{invQuantScale}[\text{qPrem}] \ll \text{invQuantLeftShift},$$

где EspValue представляет закодированное квантованное значение вне палитры, закодированное для пикселя, кодируемого как пиксель вне палитры, invQuantScale представляет массив значений шкалы обратного квантования, qPrem представляет значение остатка параметра квантования, invQuantLeftShift представляет значение параметра сдвига влево и \ll представляет битовый оператор сдвига влево.

8. Устройство по п.7, дополнительно содержащее средство для определения значения для параметра сдвига вправо на основе значения параметра (qP) квантования для блока.

9. Устройство по п.8, дополнительно содержащее средство для определения значения qP на основе индекса (cIdx) цветового компонента для цветового компонента, включающего в себя пиксель, кодируемый как пиксель вне палитры.

10. Устройство по п.8, в котором средство для определения значения для параметра сдвига вправо содержит

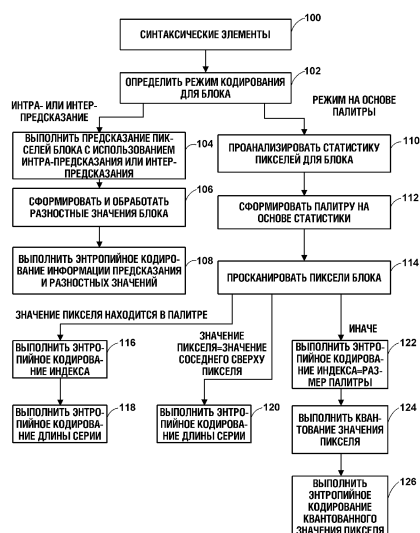
средство для вычисления отношения (qPper) квантования в соответствии с формулой $\text{qPper} = \text{qP}/6$, где qP представляет значение qP ; и

средство для вычисления значения для параметра ($\text{invQuantRightShift}$) сдвига вправо в соответствии с формулой $\text{invQuantRightShift} = 6 - \text{qPper}$.

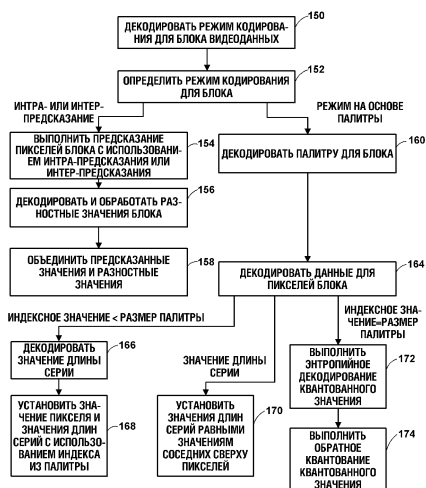
11. Устройство по п.7, в котором invQuantScale содержит {40, 45, 51, 57, 64, 72}.

12. Устройство по п.7, дополнительно содержащее

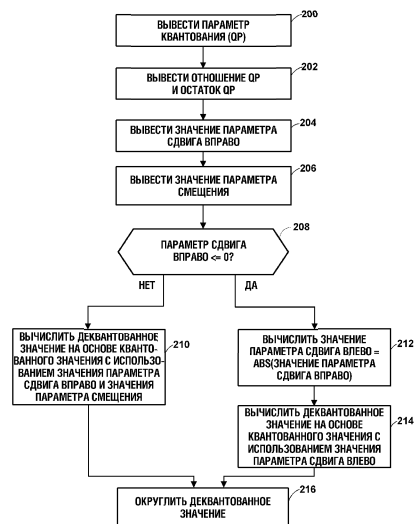
средство для декодирования палитры для блока, кодируемого в режиме на основе палитры;



Фиг. 4



Фиг. 5



Фиг. 6



Евразийская патентная организация, ЕАПВ

Россия, 109012, Москва, Малый Черкасский пер., 2