



(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2021.09.13

(21) Номер заявки
201791496

(22) Дата подачи заявки
2016.01.29

(51) Int. Cl. *H04N 19/13* (2014.01)
H03M 7/40 (2006.01)
H03M 13/00 (2006.01)
H04N 19/593 (2014.01)
H04N 19/91 (2014.01)
H04N 19/182 (2014.01)

(54) КОДИРОВАНИЕ ПИКСЕЛОВ С УПРАВЛЯЮЩИМ КОДОМ ДЛЯ ПАЛИТРОВОГО КОДИРОВАНИЯ

(31) 62/110,395; 62/110,995; 15/009,678

(32) 2015.01.30; 2015.02.02; 2016.01.28

(33) US

(43) 2017.11.30

(86) PCT/US2016/015707

(87) WO 2016/123513 2016.08.04

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

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

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

(56) WO-A2-2015006724
Vivienne Sze ET AL.: "High Efficiency Video Coding (HEVC) : Algorithms and Architectures - Chapter 8: Entropy Coding in HEVC" In: "High Efficiency Video Coding (HEVC): Algorithms and Architectures - Chapter 8: Entropy Coding in HEVC", 1 January 2014 (2014-01-01), Springer International Publishing, XP055263413, ISBN: 978-3-319-06894-7, pages 209-269, *section 8.6.5.1*

GUO X ET AL.: "AHG8: Major-color-based screen content coding", 15. JCT-VC MEETING; 23-10-2013 - 1-11-2013; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-00182, 14 October 2013 (2013-10-14), XP030115218, *sections 1 and 2.1*

SHARMAN K ET AL.: "AHG18: Worst-case Escape Code Length Mitigation", 17. JCT-VC MEETING; 27-3-2014 - 4-4-2014; VALENCIA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-Q0073-v2, 24 March 2014 (2014-03-24), XP030115973, *section1*

"TEncSbac source: tags/
HM-16.2+SCM-3.0/source/Lib/TLibEncoder/TEncSbac.cpp",

11 November 2014 (2014-11-11), XP055263026, Retrieved from the Internet: URL: <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.2%2BSCM-3.0/source/Lib/TLibEncoder/TEncSbac.cpp> [retrieved on 2016-04-05] sentences 767, 898, 971

"TComDataCU.cpp source: tags/HM-16.2+SCM-3.0/source/Lib/TLibCommon/TComDataCU.cpp", 12 November 2014 (2014-11-12), XP055263001, Retrieved from the Internet: URL: <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.2%2BSCM-3.0/source/Lib/TLibCommon/TComDataCU.cpp> [retrieved on 2016-04-05] sentence 3650-sentence 3762

"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

GUO L ET AL.: "RCE3: Results of Test 3.1 on Palette Mode for Screen Content Coding", 14. JCT-VC MEETING; 25-7-2013 - 2-8-2013; VIENNA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-N0247, 16 July 2013 (2013-07-16), XP030114764, *section 2*

ZOU F ET AL.: "CE1 Related: On escape pixel coding for palette mode", 20. JCT-VC MEETING; 10-2-2105 - 18-2-2015; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-T0112-v3, 10 February 2015 (2015-02-10), XP030117256, the whole document

XIU X ET AL.: "Non-CE1: On escape color coding for palette coding mode", 20. JCT-VC MEETING; 10-2-2105 - 18-2-2015; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-T0118, 31 January 2015 (2015-01-31), XP030117264, the whole document

ZHANG K ET AL.: "CE1-related: Table based binarization for palette_escape_val", 20. JCT-VC MEETING; 10-2-2105 - 18-2-2015; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-T0060, 30 January 2015 (2015-01-30), XP030117181, the whole document

(57) Предложены способ и устройство декодирования видеоданных, которое включает в себя запоминающее устройство, выполненное с возможностью сохранять видеоданные, и видеодекoder, выполненный с возможностью декодировать экспоненциальное кодовое слово Голомба,

представляющее по меньшей мере часть значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных, видеодекодер выполнен с возможностью декодировать экспоненциальное кодовое слово Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3 и декодировать блок с использованием значения для пиксела с управляющим кодом.

038545 B1

038545 B1

Данная заявка притязает на приоритет предварительной заявки на патент (США) номер 62/110395, поданной 30 января 2015 года, и предварительной заявки на патент (США) номер 62/110995, поданной 2 февраля 2015 года, содержимое каждой из которых полностью содержится в данном документе по ссылке.

Область техники, к которой относится изобретение

Данное раскрытие сущности относится к кодированию и декодированию блоков видеоданных.

Уровень техники

Поддержка цифрового видео может быть включена в широкий диапазон устройств, включающих в себя цифровые телевизионные приемники, системы цифровой прямой широкоэвещательной передачи, беспроводные широкоэвещательные системы, персональные цифровые устройства (PDA), переносные или настольные компьютеры, планшетные компьютеры, устройства для чтения электронных книг, цифровые камеры, цифровые записывающие устройства, цифровые мультимедийные проигрыватели, устройства видеоигр, консоли для видеоигр, сотовые или спутниковые радиотелефоны, так называемые "смартфоны", устройства видеоконференц-связи, устройства потоковой передачи видео и т.п. Цифровые видеоустройства реализуют такие технологии сжатия видео как технологии сжатия видео, описанные в стандартах, заданных посредством стандартов MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, часть 10, усовершенствованное кодирование видео (AVC), ITU-T H.265 (также упоминаемого как стандарт высокоэффективного кодирования видео (HEVC)) и расширений таких стандартов. Видеоустройства могут передавать, принимать, кодировать, декодировать и/или сохранять цифровую видеоинформацию более эффективно посредством реализации таких технологий сжатия видео.

Технологии сжатия видео выполняют пространственное (внутрикадровое) прогнозирование и/или временное (межкадровое) прогнозирование для того, чтобы уменьшать или удалять избыточность, внутренне присущую в видеопоследовательностях. Для кодирования видео на основе блоков серия последовательных видеомакроблоков (т.е. видеокادر или часть видеокадра) может быть сегментирована на видеоблоки. Видеоблоки в серии внутренне кодированных последовательных (I-) макроблоков изображения кодируются с использованием пространственного прогнозирования относительно опорных выборок в соседних блоках в идентичном изображении. Видеоблоки в серии взаимно кодированных последовательных (P- или B-) макроблоков изображения могут использовать пространственное прогнозирование относительно опорных выборок в соседних блоках в идентичном изображении или временное прогнозирование относительно опорных выборок в других опорных изображениях. Изображения могут упоминаться как кадры, и опорные изображения могут упоминаться как опорные кадры.

Пространственное или временное прогнозирование приводит в результате к прогнозирующему блоку для блока, который должен быть кодирован. Остаточные данные представляют пиксельные разности между исходным блоком, который должен быть кодирован, и прогнозирующим блоком. Взаимно кодированный блок кодируется согласно вектору движения, который указывает на блок опорных выборок, формирующий прогнозный блок, и остаточные данные указывают разность между кодированным блоком и прогнозным блоком. Внутренне кодированный блок кодируется согласно режиму внутреннего кодирования и остаточным данным. Для дополнительного сжатия остаточные данные могут быть преобразованы из пиксельной области в область преобразования, приводя к остаточным коэффициентам, которые затем могут быть квантованы. Квантованные коэффициенты, первоначально размещаемые в двумерном массиве, могут сканироваться для того, чтобы формировать одномерный вектор коэффициентов, и может применяться энтропийное кодирование с тем, чтобы достигать еще большего сжатия.

Поток битов многовидового кодирования может формироваться посредством кодирования видов, например, с нескольких перспектив. Разработаны некоторые трехмерные видеостандарты, которые используют аспекты многовидового кодирования. Например, различные виды могут передавать виды для левого и правого глаза, чтобы поддерживать трехмерное видео. Альтернативно некоторые процессы кодирования трехмерного видео могут применять так называемое многовидовое кодирование с учетом глубины. При многовидовом кодировании с учетом глубины поток трехмерных видеобитов может содержать не только компоненты видов текстуры, но также и компоненты видов глубины. Например, каждый вид может содержать один компонент вида текстуры и один компонент вида глубины.

Сущность изобретения

В общем, это раскрытие сущности описывает технологии, связанные с кодированием видеоданных с использованием палитрового режима. Более конкретно технологии этого раскрытия сущности направлены на кодирование пикселей с управляющим кодом в кодированных в палитровом режиме блоках. Например, это раскрытие сущности описывает технологии для кодирования пикселей с управляющим кодом с использованием усеченного кодового слова на основе определения максимального квантованного пиксельного значения с управляющим кодом. Это раскрытие сущности также описывает различные технологии для определения максимального квантованного пиксельного значения с управляющим кодом. Кроме того, это раскрытие сущности описывает технологии для кодирования пиксельного значения с управляющим кодом в двух частях, например кодирования первой части с использованием кодирования кодом Голомба-Райса и второй части с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

В одном примере способ декодирования видеоданных включает в себя декодирование экспоненци-

ального кодового слова Голомба, представляющего по меньшей мере часть значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных, при этом декодирование экспоненциального кодового слова Голомба содержит декодирование экспоненциального кодового слова Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3 и декодирование блока с использованием значения для пиксела с управляющим кодом.

В другом примере устройство для декодирования видеоданных включает в себя запоминающее устройство, выполненное с возможностью сохранять видеоданные, и видеодекодер, выполненный с возможностью декодировать экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных, при этом видеодекодер выполнен с возможностью декодировать экспоненциальное кодовое слово Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3 и декодировать блок с использованием значения для пиксела с управляющим кодом.

В другом примере устройство для декодирования видеоданных включает в себя средство для декодирования экспоненциального кодового слова Голомба, представляющего по меньшей мере часть значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных, при этом средство для декодирования экспоненциального кодового слова Голомба содержит средство для декодирования экспоненциального кодового слова Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3 и средство для декодирования блока с использованием значения для пиксела с управляющим кодом.

В другом примере машиночитаемый носитель хранения данных имеет сохраненными инструкции, которые при выполнении инструктируют одному или более процессоров декодировать экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных, при этом инструкции, которые инструктируют процессорам декодировать экспоненциальное кодовое слово Голомба, содержат инструкции, которые инструктируют процессорам декодировать экспоненциальное кодовое слово Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3 и декодировать блок с использованием значения для пиксела с управляющим кодом.

В другом примере способ кодирования видеоданных включает в себя определение значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных и кодирование экспоненциального кодового слова Голомба, представляющего по меньшей мере часть значения для пиксела с управляющим кодом, при этом кодирование экспоненциального кодового слова Голомба содержит кодирование экспоненциального кодового слова Голомба с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

В другом примере устройство для кодирования видеоданных включает в себя запоминающее устройство, выполненное с возможностью сохранять видеоданные, и видеокодер, выполненный с возможностью определять значение для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных и кодировать экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пиксела с управляющим кодом, при этом видеокодер выполнен с возможностью кодировать экспоненциальное кодовое слово Голомба с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

В другом примере устройство для кодирования видеоданных включает в себя средство для определения значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных и средство для кодирования экспоненциального кодового слова Голомба, представляющего по меньшей мере часть значения для пиксела с управляющим кодом, при этом средство для кодирования экспоненциального кодового слова Голомба содержит средство для кодирования экспоненциального кодового слова Голомба с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

В другом примере машиночитаемый носитель хранения данных имеет сохраненными инструкции, которые при выполнении инструктируют одному или более процессоров определять значение для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных и кодировать экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пиксела с управляющим кодом, при этом инструкции, которые инструктируют процессорам кодировать экспоненциальное кодовое слово Голомба, содержат инструкции, которые инструктируют процессорам кодировать экспоненциальное кодовое слово Голомба с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

Подробности одного или более примеров изложены на прилагаемых чертежах и в нижеприведенном описании. Другие признаки, цели и преимущества должны становиться очевидными из описания и чертежей и из формулы изобретения.

Краткое описание чертежей

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

Фиг. 2 является блок-схемой, иллюстрирующей примерный видеокодер, который может реализовывать технологии, описанные в этом раскрытии сущности.

Фиг. 3 является блок-схемой, иллюстрирующей примерный видеодекодер, который может реализовать технологии, описанные в этом раскрытии сущности.

Фиг. 4 является блок-схемой последовательности операций, иллюстрирующей примерный способ для кодирования блока видеоданных в соответствии с технологиями этого раскрытия сущности.

Фиг. 5 является блок-схемой последовательности операций, иллюстрирующей другой примерный способ для кодирования блока видеоданных в соответствии с технологиями этого раскрытия сущности.

Фиг. 6 является блок-схемой последовательности операций, иллюстрирующей примерный способ для декодирования блока видеоданных в соответствии с технологиями этого раскрытия сущности.

Фиг. 7 является блок-схемой последовательности операций, иллюстрирующей другой примерный способ для декодирования блока видеоданных в соответствии с технологиями этого раскрытия сущности.

Подробное описание изобретения

Это раскрытие сущности предоставляет технологии, которые направлены на кодирование пиксела с управляющим кодом для палитрового кодирования, а более конкретно на извлечение максимального квантованного пиксельного значения с управляющим кодом. В некоторых примерах технологии, предложенные в этом раскрытии сущности, могут удалять или исключать одно или более вычислений, которые используют зависимые от кодера параметры и/или операции деления, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом. В некоторых примерах это раскрытие сущности предоставляет новое кодовое слово, которое может быть спроектировано независимо от максимального квантованного пиксельного значения с управляющим кодом.

В некоторых примерах технологии этого раскрытия сущности могут не использовать зависимые от кодера параметры для того, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом в декодере. Вместо этого технологии этого раскрытия сущности могут использовать только размер шага квантования при извлечении параметров. Кроме того, в некоторых примерах технологии этого раскрытия сущности могут использовать суммирование и сравнение вместо операций деления в ходе извлечения максимального квантованного пиксельного значения с управляющим кодом.

В некоторых примерах технологии этого раскрытия сущности могут использовать параметры кодера для того, чтобы извлекать максимальное квантованное значение с управляющим кодом в декодере. Таким образом, технологии этого раскрытия сущности могут сохранять связанные с кодером параметры в таблице поиска на стороне декодера. Кроме того, технологии этого раскрытия сущности могут использовать операции циклической обработки или сдвига для того, чтобы извлекать максимальное число битов, чтобы представлять квантованное значение с управляющим кодом.

В некоторых примерах технологии этого раскрытия сущности могут использовать параметры кодера для того, чтобы извлекать максимальное квантованное значение с управляющим кодом без операций отсечения и деления. Таким образом, технологии этого раскрытия сущности могут сохранять связанные с кодером параметры в таблице поиска на стороне декодера. Как подробнее описано в этом раскрытии сущности, в некоторых примерах, технологии этого раскрытия сущности могут извлекать максимальное квантованное пиксельное значение с управляющим кодом с использованием параметра qP квантования и цветового компонента $bitDepth$. В частности, технологии этого раскрытия сущности могут получать значение $sMax$ с использованием таблицы поиска с qP и $bitDepth$.

При традиционном кодировании видео изображения предположительно имеют непрерывный тон и пространственную плавность. На основе этих допущений разработаны различные инструментальные средства, такие как преобразование на основе блоков, фильтрация и т.д., и такие инструментальные средства демонстрируют хорошую производительность для видео естественного контента. Тем не менее, в таких приложениях как удаленный рабочий стол, совместная работа и беспроводной дисплей, машиногенерируемый экранный контент (например, такой как текст или компьютерная графика) может представлять собой доминирующий контент, который должен сжиматься. Этот тип контента имеет тенденцию иметь дискретный тон и рисовать резкие линии и высококонтрастные границы объектов. Допущение касательно непрерывного тона и плавности больше не может применяться для экранного контента, и в силу этого традиционные технологии кодирования видео не могут представлять собой эффективные способы сжимать видеоданные, включающие в себя экранный контент.

Это раскрытие сущности описывает кодирование на основе палитр, которое может быть, в частности, подходящим для кодирования экраногенерируемого контента. Например, при условии, что конкретная область видеоданных имеет относительно небольшое число цветов, видеокодер (например, видеокодер или видеодекодер) может формировать так называемую "палитру", чтобы представлять видеоданные конкретной области. Палитра может выражаться как таблица цветов или пиксельных значений, представляющих видеоданные конкретной области (например, данного блока). Например, палитра может включать наиболее доминирующие пиксельные значения в данный блок. В некоторых случаях наиболее доминирующие пиксельные значения могут включать в себя одно или более пиксельных значений, которые возникают наиболее часто в блоке. Дополнительно в некоторых случаях видеокодер может применять пороговое значение, чтобы определять то, должно ли нет пиксельное значение быть включено в качестве одного из наиболее доминирующих пиксельных значений в блок. Согласно различным аспектам кодирования на основе палитр видеокодер может кодировать значения индекса, указывающие одно или

более пиксельных значений текущего блока, вместо кодирования фактических пиксельных значений или их остатков для текущего блока видеоданных. В контексте кодирования на основе палитры значения индекса указывают соответствующие записи в палитре, которые используются для того, чтобы представлять отдельные пиксельные значения текущего блока.

Например, видеокодер может кодировать блок видеоданных посредством определения палитры для блока (например, явного кодирования палитры, прогнозирования палитры или комбинации вышеозначенного), нахождения записи в палитре, чтобы представлять одно или более пиксельных значений, и кодирования блока с помощью значений индекса, которые указывают запись в палитре, используемую для того, чтобы представлять пиксельные значения блока. В некоторых примерах видеокодер может передавать в служебных сигналах палитру и/или значения индекса в кодированном потоке битов. В свою очередь, видеодекодер может получать из кодированного потока битов палитру для блока, а также значения индекса для отдельных пикселей блока. Видеодекодер может связывать значения индекса пикселей с записями палитры, чтобы восстанавливать различные пиксельные значения блока.

Более конкретно пиксели блока, кодированного с использованием палитрового режима, могут кодироваться с использованием "индексного" режима, в котором пиксел кодируется с использованием ссылки на палитру, или режима с копированием сверху, в котором пиксел кодируется с использованием ссылки на соседний сверху пиксел. Третий вариант заключается в том, чтобы кодировать пиксел в качестве пиксела с управляющим кодом. В этом случае значение пиксела (или квантованное значение для пиксела) передается в служебных сигналах непосредственно.

В соответствии с технологиями этого раскрытия сущности видеокодеры (например, видеокодеры и видеодекодеры) могут быть выполнены с возможностью энтропийно кодировать квантованное значение для пиксела с управляющим кодом с использованием комбинации режимов энтропийного кодирования. В частности, в одном примере видеокодер кодирует первую часть квантованного значения для пиксела с управляющим кодом с использованием кодирования кодом Голомба-Райса и вторую часть квантованного значения с использованием экспоненциального кода Голомба с параметром 3. Для множества синтаксических элементов экспоненциальный код Голомба с параметром 3 дает в результате относительно низкую производительность, но эвристическое тестирование раскрывает неожиданно высокую производительность при энтропийном кодировании квантованных значений для пикселей с управляющим кодом с использованием экспоненциального кода Голомба с параметром 3.

Соответственно видеокодер может сначала определять кодировать блок пикселей (например, единицу кодирования (CU) или единицу прогнозирования (PU)) с использованием палитрового режима. Видеокодер может анализировать пиксели, чтобы формировать палитру для блока, и затем сканировать пиксели, чтобы определять пиксельный режим кодирования для каждого пиксела. Например, видеокодер может выбирать индексный режим для пиксела, если значение пиксела включено в палитру, режим с копированием сверху, если пиксел имеет значение, которое равно соседнему сверху пикселу, или режим кодирования управляющим кодом, если пиксел не находится в палитре и также не может копироваться сверху. Когда видеокодер определяет то, что пиксел должен кодироваться с использованием режима кодирования управляющим кодом, видеокодер может энтропийно кодировать значение пиксела в качестве первой части и второй части, причем первая часть и вторая часть, при конкатенации, представляют полное значение для пиксела. В частности, видеокодер может кодировать первую часть с использованием кодирования кодом Голомба-Райса и вторую часть с использованием экспоненциального кода Голомба с параметром 3. Помимо этого, видеокодер может передавать в служебных сигналах то, что пиксел кодируется с использованием режима кодирования управляющим кодом, посредством передачи в служебных сигналах значения индекса, которое равно размеру палитры для блока. Видеокодер также может кодировать палитру для блока, например с использованием палитры ранее кодированного блока в качестве опорной палитры.

Видеодекодер аналогично может принимать кодированные данные для блока. Кодированные данные могут включать в себя данные, указывающие то, что блок прогнозируется с использованием палитрового режима, данные, представляющие палитру (например, кодированная версия палитры), и данные, представляющие значения и режимы кодирования для каждого из пикселей блока. Кроме того, для каждого кодированного в режиме кодирования управляющим кодом пиксела видеодекодер может энтропийно декодировать первую часть значения пиксела с использованием кодирования кодом Голомба-Райса и вторую часть значения с использованием экспоненциального кода Голомба с параметром 3 и затем конкатенировать первую и вторую части для того, чтобы воспроизводить значение пиксела.

Стандарты кодирования видео включают в себя ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 или ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual и ITU-T H.264 (также известный как ISO/IEC MPEG-4 AVC), включающий в себя расширения масштабируемого кодирования видео (SVC) и кодирования многовидового видео (MVC).

Недавно завершено проектирование нового стандарта кодирования видео, а именно стандарта высокоэффективного кодирования видео (HEVC), посредством Объединенной группы для совместной работы над видеостандартами (JCT-VC) Экспертной группы в области кодирования видео (VCEG) ITU-T и Экспертной группы по киноизображению (MPEG) ISO/IEC. Последняя HEVC-спецификация, называемая

"HEVC-версией 1" в дальнейшем в этом документе, описывается в ITU-T H.265, доступном по адресу <http://www.itu.int/rec/T-REC-H.265-201304-I>. Расширения диапазона в HEVC, а именно HEVC-RExt, также разрабатываются посредством JCT-VC. Недавний рабочий проект (WD) расширений диапазона, называемый "RExt WD7" в дальнейшем в этом документе, доступен по адресу phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1005-v4.zip.

Недавно JCT-VC начал разработку кодирования экранного контента (SCC), которое основано на HEVC-RExt, и определенные основные технологии находятся на рассмотрении. Проект документа (R. Joshi, J. Xu "HEVC Screen Content Coding Draft Text 2", JCTVC-S1005), описывающий HEVC-кодирование экранного контента, содержится по адресу hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.2+SCM-3.0.

В некоторых примерах технологии кодирования на основе палитр могут быть выполнены с возможностью использования в одном или более режимов кодирования HEVC-стандарта или HEVC SCC-стандарта. В других примерах технологии кодирования на основе палитр могут использоваться независимо либо в качестве части других существующих или будущих систем или стандартов. В некоторых примерах технологии для кодирования на основе палитр видеоданных могут использоваться с одной или более других технологий кодирования, таких как технологии для взаимного прогнозирующего кодирования или внутреннего прогнозирующего кодирования видеоданных. Например, как подробнее описано ниже, кодер, или декодер, или комбинированный кодер-декодер (кодек) может быть выполнен с возможностью осуществлять взаимное и внутреннее прогнозирующее кодирование, а также кодирование на основе палитр.

Относительно инфраструктуры HEVC в качестве примера технологии кодирования на основе палитр могут быть выполнены с возможностью использоваться в качестве режима на основе единиц кодирования (CU). В других примерах технологии кодирования на основе палитр могут быть выполнены с возможностью использоваться в качестве режима на основе единиц прогнозирования (PU) в инфраструктуре HEVC. Соответственно все следующие раскрытые процессы, описанные в контексте CU-режима, дополнительно или альтернативно, могут применяться к PU. Тем не менее, эти примеры на основе HEVC не должны считаться ограничением или недочетом технологий кодирования на основе палитр, описанных в данном документе, поскольку технологии в данном документе могут применяться, чтобы работать независимо либо в качестве части других существующих или будущих разрабатываемых систем/стандартов. В этих случаях единица для палитрового кодирования может представлять собой квадратные блоки, прямоугольные блоки или даже области непрямоугольной формы.

Базовая идея касательно кодирования на основе палитр состоит в том, что для каждой единицы кодирования (CU) извлекается палитра, которая содержит (и может состоять из) наиболее доминирующие пиксельные значения в текущей CU. Размер и элементы палитры сначала передаются из видеокодера в видеодекодер. Размер и/или элементы палитры могут непосредственно кодироваться или прогнозирующе кодироваться с использованием размера и/или элементов палитры в соседних CU (например, верхней и/или левой кодированной CU). После этого пиксельные значения в CU кодируются на основе палитры согласно определенному порядку сканирования. Для каждого пиксельного местоположения в CU, флаг, например, `palette_flag`, сначала передается, чтобы указывать то, включено или нет пиксельное значение в палитру. Для тех пиксельных значений, которые увязываются с записью в палитре, индекс палитры, ассоциированный с этой записью, передается в служебных сигналах для данного пиксельного местоположения в CU. Для тех пиксельных значений, которые не существуют в палитре, специальный индекс может назначаться пикселу, и фактическое пиксельное значение передается для данного пиксельного местоположения в CU. Эти пикселы упоминаются как "пикселы с управляющим кодом". Пиксел с управляющим кодом может кодироваться с использованием любого существующего способа энтропийного кодирования, такого как кодирование фиксированной длины, унарное кодирование и т.д.

Это раскрытие сущности описывает способы для того, чтобы поддерживать кодирование видеоконтента, в частности, экранного контента с палитровым кодированием. Предложенные способы относятся главным образом к кодированию пиксела с управляющим кодом для кодирования в палитровом режиме.

На основе характеристик видео экранного контента введено палитровое кодирование, чтобы повысить эффективность SCC, описанную в JCTVC-M0323. В частности, палитровое кодирование вводит таблицу поиска, т.е. цветовую палитру, чтобы сжимать повторяющиеся пиксельные значения на основе того факта, что в SCC цвета в одной CU обычно концентрируются в нескольких пиковых значениях. С учетом палитры для конкретной CU, пикселы в CU увязываются с индексом палитры. На второй стадии предложен эффективный способ кодирования по длинам серий с копированием слева, чтобы эффективно сжимать повторяющийся шаблон индексного блока. Позднее в работе авторов Guo и др. "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, Вена, Австрия, 25 июля - 2 августа 2013 года, режим кодирования на основе индексов палитр обобщен таким образом, чтобы копировать слева и копировать сверху с кодированием по длинам серий. Следует отметить, что процесс преобразования не активируется для палитрового кодирования, чтобы не допускать размывания резких контуров, которое может иметь негативные воздействия на визуальное качество экранного контента.

Палитра представляет собой структуру данных, которая сохраняет пары (индекса, пиксельного значения). Спроектированная палитра может определяться в кодере, например, посредством гистограммы пиксельных значений в текущем блоке видеоданных, таком как текущая CU. Например, пиковые значения в гистограмме добавляются в палитру, в то время как пиксельные значения низкой частоты не включены в палитру. Размер палитры может ограничиваться таким образом, что он находится в диапазоне от 0 до `max_palette_size`, который может быть равным 31.

Для SCC, CU-блоки в одной серии последовательных макроблоков могут совместно использовать множество доминирующих цветов. Следовательно, можно прогнозировать палитру текущего блока с использованием палитр предыдущих CU на основе палитрового режима (в порядке декодирования CU) в качестве ссылки. В частности, двоичный вектор 0-1 может передаваться в служебных сигналах, чтобы указывать то, многократно используются или нет пиксельные значения в опорной палитре посредством текущей палитры. В качестве примера в нижеприведенных табл. 1 и 2 предполагается, что опорная палитра имеет 6 элементов. Вектор (1, 0, 1, 1, 1, 1) передается в служебных сигналах с текущей палитрой, который указывает то, что v_0 , v_2 , v_3 , v_4 и v_5 многократно используются в текущей палитре, тогда как v_1 не используется многократно. Если текущая палитра содержит цвета, которые не являются прогнозируемыми из опорной палитры, число непрогнозируемых цветов кодируется, и затем эти цвета непосредственно передаются в служебных сигналах. Например, в табл. 1 и 2, u_0 и u_1 непосредственно передаются в служебных сигналах в поток битов.

Таблица 1. Опорная палитра

Индекс	Пиксельное значение
0	V_0
1	V_1
2	V_2
3	V_3
4	V_4
5	V_5

Таблица 2. Текущая палитра

Флаг прогнозирования	Индекс	Пиксельное значение
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 помощью палитрового режима, палитра может прогнозироваться из записей палитры ранее закодированных в палитровом режиме блоков, может явно передаваться в служебных сигналах в качестве новых записей, либо палитра ранее закодированного блока может полностью многократно использоваться. Второй случай называется совместным использованием палитр, и флаг `palette_share_flag` передается в служебных сигналах, чтобы указывать то, что полная палитра предыдущего блока многократно используется без модификации как есть. Примеры опорной палитры и текущей палитры показаны в вышеприведенных табл. 1 и 2. В частности, табл. 1 иллюстрирует примерную опорную палитру, и табл. 2 иллюстрирует примерную текущую палитру, которая может прогнозироваться из опорной палитры табл. 1.

В текущем эталонном программном обеспечении SCM3.0 два первичных аспекта палитрового кодирования с нормативной точки зрения заключаются в кодировании палитры и кодировании индекса палитры для каждой выборки в кодируемом блоке в палитровом режиме. Кодирование индексов палитр выполняется с использованием двух первичных режимов, "индексного" режима и режима "с копированием вверх". Это передается в служебных сигналах посредством кодирования флага `palette_mode`. "Индексный" режим также используется для того, чтобы указывать выборки с управляющим кодом, т.е. вы-

борки, которые не принадлежат палитре. В текущем проектном решении режим "с копированием вверх" является невозможным для первой строки палитрового блока. Помимо этого, режим "с копированием вверх" не может следовать после другого режима "с копированием вверх". В этих случаях "индексный" режим логически выводится.

В частности, для палитрового режима пиксели в CU кодируются в горизонтальном/вертикальном змеевидном порядке сканирования следующим образом.

1. "Индексный" режим. В этом режиме один индекс палитры сначала передается в служебных сигналах. Если индекс равен размеру палитры, это указывает то, что выборка представляет собой выборку с управляющим кодом. В этом случае выборочное значение или значение квантованных выборок для каждого компонента передается в служебных сигналах. Например, если размер палитры равен 4 для выборок без управляющего кода, индексы палитр находятся в диапазоне [0, 3]. В этом случае значение индекса 4 обозначает выборку с управляющим кодом. Если индекс указывает выборку без управляющего кода, передается в служебных сигналах длина серии, которая указывает число последующих выборок в порядке сканирования, которые совместно используют идентичный индекс, посредством неотрицательного значения $n-1$, указывающего длину серии, что означает то, что следующие n пикселей, включающие в себя текущий пиксел, имеют пиксельный индекс, идентичный первому передаваемому в служебных сигналах пиксельному индексу.

2. Режим серий "с копированием сверху" (CA). В этом режиме только неотрицательное значение $m-1$ длины серии передается, чтобы указывать то, что для следующих m пикселей, включающих в себя текущий пиксел, индексы палитр являются идентичными своим соседним узлам непосредственно выше соответственно. Следует отметить, что этот режим отличается от "индексного" режима в том смысле, что индексы палитр могут отличаться в режиме серий с копированием сверху.

В текущем проектном решении палитровый режим передается в служебных сигналах на уровне CU, но может быть возможным передавать его в служебных сигналах на уровне PU. Флаг, `palette_esc_val_present_flag`, также передается в служебных сигналах, чтобы указывать присутствие выборок с управляющим кодом в текущем блоке.

В палитровом режиме пиксельное сканирование в блоке может иметь два типа: вертикальное пересекающееся или горизонтальное пересекающееся (змеевидное) сканирование. Шаблон сканирования, используемый в блоке, извлекается согласно флагу `palette_transpose_flag`, передаваемому в служебных сигналах в расчете на единицу блока.

В ходе кодирования на основе индексов палитр может применяться процесс регулирования индексов палитр. Начиная со второго пикселя в блоке, он состоит из проверки палитрового режима ранее кодированного пикселя. Во-первых, размер палитры уменьшается на 1, и если режим слева равен режиму серий, то индекс палитры, который должен кодироваться, уменьшается на 1, если индекс превышает индекс левой палитры либо если режим слева представляет собой режим копирования, то индекс палитры, который должен кодироваться, уменьшается на 1, если индекс превышает индекс верхней палитры. Описание предоставляется со стороны кодирования, и соответствующий процесс также может выполняться в обратном порядке на стороне декодера.

В SCM 3.0 приспособлены следующие оптимизации синтаксиса.

Если размер палитры равен 0, все пиксели с управляющим кодом извлекаются, и флаг присутствия управляющего кода, палитровый режим, индекс палитры, серия палитры и флаг транспонирования палитры не передаются в служебных сигналах, и флаг присутствия управляющего кода логически выводится равным 1, палитровый режим логически выводится равным индексному режиму, индекс палитры задается равным управляющему коду, значение серии палитры задается равным размеру блока, и флаг транспонирования палитры задается равным 0.

Если размер палитры равен 1, и пиксели с управляющим кодом не используются в блоке, то палитровый режим, серия палитры или флаг транспонирования палитры не передаются в служебных сигналах, и палитровый режим извлекается равным индексному режиму, индекс палитры задается равным 0, значение серии палитры задается равным размеру блока и флаг транспонирования палитры задается равным 0.

В SCM3.0 усеченное двоичное (TB) кодовое слово используется для того, чтобы передавать в служебных сигналах квантованные пиксели с управляющим кодом. Это кодовое слово может упоминаться как синтаксический элемент `palette_escape_val`. Чтобы использовать усеченный двоичный режим, требуется общее число возможных значений. Например, если усеченное двоичное кодовое слово используется для того, чтобы обозначать набор N из возможных значений $\{0, 1, 2, 3, \dots, N-1\}$, N используется для того, чтобы составлять усеченное двоичное кодовое слово (где $N-1$ является максимальным значением). Эквивалентно, когда максимальное значение известно, $N-1$ в этом примере (при условии начала с 0), соответствующее усеченное двоичное кодовое слово может составляться идентично. Чтобы получать максимальные квантованные значения с управляющим кодом, следующая процедура используется в SCM3.0.

1. Параметр qP квантования извлекается следующим образом:

$$qP = (cIdx == 0) ? Qp'_{\gamma} : ((cIdx == 1) ? Qp'_{cb} : Qp'_{cr})$$

2. Коэффициент `quantScale` масштабирования при квантовании, смещение

`rightShiftOffset` со сдвигом вправо и параметр `rightShiftParam` сдвига вправо извлекаются на основе таблицы поиска для HEVC-квантования следующим образом:

```
quantScale=g_quantScales[qP%6], где g_quantScales={26214,
23302, 20560, 18396, 16384, 14564}
rightShiftParam=14+⌊qP/6⌋
rightShiftOffset=1<<(rightShiftParam-1)
```

3. Параметр `qStep` размера шага квантования извлекается следующим образом:

```
qStep=(qP==0)?Round(2(qP-4)/6):1
```

4. Максимальное возможное квантованное значение `maxValue` извлекается следующим образом:

```
maxValue=((1<<bitDepth)-1)/qStep
```

5. Число `numBins` элементов выборки для того, чтобы представлять `maxValue`, извлекается следующим образом:

```
while(maxValue) {
maxValue=maxValue>>1
numBins++
}
```

6. Максимальный параметр `sMax` для усеченного двоичного кодового слова извлекается следующим образом:

```
sMax=clip3(0, 2numBins, ((1<<bitDepth)-1)*quantScale+rightShiftOffset)>>rightShiftParam)
```

Возникают по меньшей мере две потенциальных проблемы с текущим проектным решением на основе пикселей с управляющим кодом.

Во-первых, на стороне декодера коэффициент масштабирования при квантовании, смещение со сдвигом вправо и параметры не доступны, поскольку эти параметры являются типичными параметрами кодера и не стандартизированы посредством декодера. Различный кодер может иметь различный коэффициент масштабирования, параметр сдвига вправо и смещение, и в силу этого означенное может приводить к некорректным вычислениям `sMax` с использованием несогласованных параметров между различным кодером и декодером.

Во-вторых, извлечение `maxValue` включает в себе операцию деления, которая не требуется для декодера. В частности, различные декодеры имеют различную точность или интерпретации для операций деления и в силу этого могут иметь различные вычисления `maxValue` на различных платформах декодеров. Следовательно, несогласованные вычисления `maxValue` могут приводить к рассогласованию кодера/декодера.

Это раскрытие сущности описывает различные технологии, которые могут предлагать решения этих и других проблем. Различные технологии этого раскрытия сущности включают в себя удаление вычисления с использованием зависимых от кодера параметров и/или удаление операций деления.

Дополнительно или альтернативно может быть спроектировано новое кодовое слово, которое может быть независимым от максимального значения.

В одном примере зависимые от кодера параметры не используются для того, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом в декодере. Вместо этого только размер шага квантования используется при извлечении параметров. Кроме того, суммирование и сравнение используются вместо операций деления в ходе извлечения максимального квантованного пиксельного значения с управляющим кодом. Следующая процедура предложена, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом.

1. Параметр `qP` квантования извлекается на основе цветового компонента. Различные цветовые компоненты могут иметь различные параметры квантования. Ниже приводится пример для того, чтобы получать `Qp'Y`, `Qp'Cb` или `Qp'Cr`, для компонента `Y`, `Cb` и `Cr` соответственно, которые могут отличаться или не отличаться.

$$qP = (cIdx == 0) ? Qp'_{Y} : ((cIdx == 1) ? Qp'_{Cb} : Qp'_{Cr}) \quad (1)$$

2. Параметр `qStep` размера шага квантования извлекается с использованием формулы с параметром `qP` квантования в качестве ввода. Следующая формула является примером, в котором `Round(.)` является оператором округления. Следует отметить, что она не ограничена только `Round(.)`. Вместо этого она также может заменяться посредством других операторов, например оператора округления $\lfloor \cdot \rfloor$ в меньшую сторону или оператора $\lceil \cdot \rceil$ округления в большую сторону:

$$qStep = (qP == 0) ? Round(2^{(qP-4)/6}) : 1 \quad (2)$$

3. Максимальное возможное квантованное значение `sMax` (инициализированное равным 0) извлека-

ется с учетом максимального пиксельного значения перед квантованием и $qStep$ с использованием цикла следующим образом ($qStepTemp$ инициализируется как $qStep$):

```
while (qStepTemp<=((1<<bitDepth)-1)) {
    qStepTemp+=qStep
    cMax++
} (3)
```

Кроме того, затем $cMax$ используется для того, чтобы составлять усеченное двоичное кодовое слово, чтобы кодировать квантованные пиксели с управляющим кодом для текущей единицы кодирования. Следует отметить, что все квантованные пиксели с управляющим кодом ограничиваются в диапазоне $[0, cMax]$ включительно.

Альтернативно на этапе 3 $cMax$ (инициализированное равным 1) может извлекаться с использованием операций сдвига следующим образом с меньшей точностью:

```
while (qStep<=((1<<bitDepth)-1)) {
    qStep<<=1
    cMax<<=1
}
cMax-- (4)
```

В другом примере параметры кодера (например, значение ($quantScale$) масштабирования при квантовании, значение ($rightShiftOffset$) смещения со сдвигом вправо и значение параметра ($rightShiftParam$) сдвига вправо) могут использоваться для того, чтобы извлекать максимальное квантованное значение с управляющим кодом в декодере. Таким образом, связанные с кодером параметры могут сохраняться в таблице поиска на стороне декодера. Кроме того, операции циклической обработки или сдвига могут использоваться для того, чтобы извлекать максимальное число битов, чтобы представлять квантованное значение с управляющим кодом. Следующая процедура может использоваться для того, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом.

1. Параметр qP квантования извлекается на основе цветового компонента. Различный цветовой компонент может иметь различные параметры квантования. Ниже приводится пример для того, чтобы получать Qp'_Y , Qp'_{Cb} или Qp'_{Cr} , для компонента Y , Cb и Cr соответственно, которые могут отличаться или не отличаться.

$$qP = (cIdx == 0) ? Qp'_Y : ((cIdx == 1) ? Qp'_{Cb} : Qp'_{Cr})$$

2. Коэффициент $quantScale$ масштабирования при квантовании, смещение $rightShiftOffset$ со сдвигом вправо и параметр $rightShiftParam$ сдвига вправо извлекаются на основе таблицы поиска, сохраненной, например, в декодере:

```
quantScale=g_quantScales[qP%6], где g_quantScales={26214,
23302, 20560, 18396, 16384, 14564}
rightShiftParam=14+ $\lfloor qP/6 \rfloor$ 
rightShiftOffset=1<<(rightShiftParam-1)
```

3. Параметр $qStep$ размера шага квантования извлекается с использованием формулы с параметром qP квантования в качестве ввода. Следующая формула является примером, чтобы извлекать $qStep$, в котором $Round(.)$ является оператором округления. Следует отметить, что она не ограничена только $Round(.)$. Вместо этого она также может заменяться посредством других операторов, например, оператора $\lfloor \cdot \rfloor$ округления в меньшую сторону или оператора $\lceil \cdot \rceil$ округления в большую сторону:

$$qStep = (qP == 0) ? Round(2^{(qP-4)/6}) : 1$$

4. Максимальное возможное квантованное значение $maxValue$ (инициализированное равным 0) извлекается следующим образом

```
(qStepTemp инициализируется как qStep):
while (qStepTemp<=((1<<bitDepth)-1)) {
    qStepTemp+=qStep
    maxValue++
} (5)
```

5. Число $numBins$ элементов выборки для того, чтобы представлять $maxValue$, извлекается следующим образом:

```

while (maxValue) {
    maxValue=maxValue>>1
    numBins++
} (6)

```

6. Максимальный параметр $cMax$ для усеченного двоичного кодового слова извлекается следующим образом:

$$cMax = clip3(0, 2^{numBins}, ((1 \ll bitDepth) - 1) * quantScale + rightShiftOffset) \gg rightShiftParam \quad (7)$$

Следует отметить, что в этом случае квантованный пиксел с управляющим кодом ограничивается таким образом, что он находится в пределах диапазона $[0, cMax]$ включительно.

В другом примере параметры кодера могут использоваться для того, чтобы извлекать максимальное квантованное значение с управляющим кодом, без выполнения операций отсечения и/или деления. Таким образом, связанные с кодером параметры должны сохраняться в таблице поиска на стороне декодера. Следующая процедура может использоваться для того, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом.

1. Параметр qP квантования извлекается на основе цветового компонента. Различный цветовой компонент может иметь различные параметры квантования. Ниже приводится пример для того, чтобы получать Qp'_Y , Qp'_{Cb} или Qp'_{Cr} , для компонента Y , Cb и Cr соответственно, которые могут отличаться или не отличаться:

$$qP = (cIdx == 0) ? Qp'_Y : ((cIdx == 1) ? Qp'_{Cb} : Qp'_{Cr})$$

2. Коэффициент $quantScale$ масштабирования при квантовании, смещение $rightShiftOffset$ со сдвигом вправо и параметр $rightShiftParam$ сдвига вправо извлекаются на основе таблицы поиска, сохраненной, например, в декодере:

$$\begin{aligned}
 quantScale &= g_quantScales[qP \% 6], \quad \text{где } g_quantScales = \{26214, \\
 &23302, 20560, 18396, 16384, 14564\} \\
 rightShiftParam &= 14 + \lfloor qP / 6 \rfloor \\
 rightShiftOffset &= 1 \ll (rightShiftParam - 1)
 \end{aligned}$$

3. Максимальный параметр $cMax$ для усеченного двоичного кодового слова извлекается следующим образом:

$$cMax = ((1 \ll bitDepth) - 1) * quantScale + rightShiftOffset \gg rightShiftParam \quad (8)$$

Следует отметить, что в этом случае квантованный пиксел с управляющим кодом ограничивается в диапазоне $[0, cMax]$ включительно.

В другом примере максимальное квантованное пиксельное значение с управляющим кодом может извлекаться с использованием параметра qP квантования и цветового компонента $bitDepth$, вместе с таблицей поиска. В частности, $cMax$ может получаться с использованием просмотровой таблицы с циклической обработкой с qP и $bitDepth$. Ниже приводится пример $cMax$ извлечения.

1. Параметр qP квантования извлекается на основе цветового компонента. Различный цветовой компонент может иметь различные параметры квантования. Ниже приводится пример для того, чтобы получать Qp'_Y , Qp'_{Cb} или Qp'_{Cr} , для компонента Y , Cb и Cr соответственно, которые могут отличаться или не отличаться:

$$qP = (cIdx == 0) ? Qp'_Y : ((cIdx == 1) ? Qp'_{Cb} : Qp'_{Cr})$$

2. Базовый параметр $baseQp$ квантования извлекается с данным $bitDepth$:

$$baseQp = qP - 6 * (bitDepth - 8)$$

3. Если $baseQp$ равен или выше 0, $cMax$ может быть найден через таблицу поиска с учетом $baseQp$ в качестве входной записи.

4. В противном случае ($baseQp$ является отрицательным), и $cMax$ может извлекаться следующим образом:

$$cMax = Table[qP] \gg (bitDepth - 8) \quad (9)$$

Таблица 3. Пример таблицы поиска $cMax$

qP	0	1	2	3	4	5	6	7	8	9	10	11	12	1	1
$cMax$	40	36	32	28	25	22	20	18	16	14	12	11	10	9	8
	5	1	1	6	5	7	2	0	1	3	8	4	1	0	0

qP	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
cMax	72	64	57	51	45	40	36	32	28	25	23	20	18	16	14
qP	30	31	32	32	34	35	36	37	38	39	40	41	42	43	44
cMax	13	11	10	9	8	7	6	6	5	4	4	4	3	3	3
qP				45	46	47	48	49	50	51					
cMax				2	2	2	2	1	1	1					

Таблица 4. Другой пример таблицы поиска cMax

qP	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
cMax	25	25	25	25	25	22	20	18	16	14	12	11	10	9	8
	5	5	5	5	5	7	2	0	1	3	8	4	1	0	0
qP	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
cMax	72	64	57	51	45	40	36	32	28	25	23	20	18	16	14
qP	30	31	32	32	34	35	36	37	38	39	40	41	42	43	44
cMax	13	11	10	9	8	7	6	6	5	4	4	4	3	3	3
qP				45	46	47	48	49	50	51					
cMax				2	2	2	2	1	1	1					

В некоторых примерах система кодирования, такая как кодер или декодер, может извлекать максимальное значение с использованием формулы на основе уравнений на основе параметров битовой глубины и квантования, объединенно или отдельно в зависимости от примера. Следующая процедура является одним примером, чтобы объединенно рассматривать параметр битовой глубины и квантования.

Параметр qP квантования извлекается на основе цветового компонента. Различный цветовой компонент может иметь различные параметры квантования. Ниже приводится пример для того, чтобы получить Qp'_Y , Qp'_{Cb} или Qp'_{Cr} , для компонента Y, Cb и Cr соответственно, которые могут отличаться или не отличаться:

$$qP = (cIdx == 0) ? Qp'_Y : ((cIdx == 1) ? Qp'_{Cb} : Qp'_{Cr})$$

Базовый параметр qPBase квантования предварительно задается как в кодере, так и в декодере или передается в служебных сигналах на разных уровнях, таких как уровни SPS, PPS, серии последовательных макроблоков и т.д.; либо адаптивно извлекается на основе каждого значения qP и битовой глубины объединенно или отдельно. Ниже приводится пример для того, чтобы использовать фиксированное число для qPBase:

$$qPBase = 4 \quad (10)$$

Коэффициент quantBits уменьшения числа битов квантования извлекается с использованием уравнения или на основе таблицы поиска, сохраненной как в кодере, так и в декодере следующим образом. Следует отметить, что операция, активированная в данном документе, не ограничена операцией $\lfloor \cdot \rfloor$ округления в меньшую сторону, как показано в следующем уравнении. Операция $\lceil \cdot \rceil$ округления в большую сторону может использоваться вместо операции округления в меньшую сторону.

$$quantBits = \begin{cases} (qP - qPBase) > 0 ? \lceil (qP - qPBase) / 6 \rceil : 0 \\ \end{cases} \quad (11)$$

Максимальный битовый параметр maxBit может извлекаться на основе битовой глубины и quantBits, и ниже приводится пример:

$$maxBit = bitDepth - quantBits \quad (12)$$

Максимальный параметр cMax для усеченного двоичного кодового слова (или другого кодового слова, которое требует вычисления возможного максимального значения, например, кодового слова фиксированной длины) извлекается следующим образом:

$$cMax = (1 \ll maxBit) - 1 \quad (13)$$

Следует отметить, что в этом случае квантованный пиксел с управляющим кодом ограничивается в диапазоне $[0, cMax]$ включительно.

Также следует отметить, что вышеприведенные примеры разрешают случай, когда текущий палитровый режим работает в режиме с потерями. Когда текущий палитровый режим работает в режиме без потерь, максимальный параметр cMax может быть оставлен без изменений, аналогично текущему SCM3.0, в форме

$$cMax = ((1 \ll bitDepth) - 1) .$$

Кроме того, различные примеры, поясненные выше, применяются к кодовому слову переменной длины. Если кодовое слово фиксированной длины используется для того, чтобы кодировать квантованные пиксели с управляющим кодом, процедура, аналогичная процедуре в вышеприведенных примерах, может использоваться для того, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом. Кроме того, затем цикл может использоваться для того, чтобы определять число битов, чтобы представлять максимальное квантованное пиксельное значение с управляющим кодом.

Это раскрытие сущности также описывает технологии, которые могут использоваться для того, чтобы проектировать кодовое слово, независимо от ограничения по максимальному значению. Эти технологии могут использоваться отдельно или в комбинации с различными примерами, поясненными выше. В частности, кодовое слово без ограничения по максимальному значению может использоваться в ходе кодирования на основе квантованных пикселей с управляющим кодом. В одном примере конкатенация кодового слова Голомба-Райса и экспоненциального кодового слова Голомба используется для того, чтобы кодировать квантованные пиксели с управляющим кодом. Альтернативно унарный код или кодовые слова семейства кодов Голомба либо их комбинации также могут применяться в данном документе в ходе кодирования квантованных пикселей с управляющим кодом. Например, кодер или декодер может использовать код Голомба-Райса с параметром K Райса и/или экспоненциальный код Голомба с параметром L , где K и L могут быть любым целым числом, $0, 1, 2, 3, \dots$. Значения параметров (L, K) могут зависеть от битовой глубины. Квантованный пиксел с управляющим кодом может ограничиваться 15-битовой точностью (знак "плюс") для 8-битового входного сигнала, в то время как точность может сохраняться в качестве $\max(15, \text{bitDepth}+6)$. Например, кодирование кодом Голомба-Райса может использоваться для того, чтобы кодировать первую часть пиксельного значения с управляющим кодом, и экспоненциальный код Голомба с параметром 3 может использоваться для того, чтобы кодировать вторую часть пиксельного значения с управляющим кодом.

При кодировании экспоненциальным кодом Голомба с параметром L значение, которое должно кодироваться, делится на 2^L . Результат этого деления кодируется с использованием экспоненциального кодового слова Голомба порядка 0 , и остаток (т.е. значение по модулю 2^L) кодируется в двоичной форме. Таким образом, параметр L оказывает влияние на число битов, включенных в остаток. Следует понимать, что идентичное значение, при кодировании с использованием различных значений для параметра L , может приводить к экспоненциальным кодовым словам Голомба, имеющим различные длины. Кроме того, в некоторых случаях большее значение для параметра L может приводить к более короткому кодовому слову для значения, которое должно кодироваться, чем меньшее значение для параметра L для идентичного значения, которое должно кодироваться.

Хотя описываются как независимые от ограничения по максимальному значению, технологии для кодирования пиксельного значения с управляющим кодом с использованием комбинации кода Голомба-Райса и экспоненциального кода Голомба с параметром 3 могут использоваться в комбинации с любой из других технологий, поясненных выше для определения максимального квантованного пиксельного значения с управляющим кодом. Например, пиксельное значение с управляющим кодом по-прежнему может усекается (например, один бит может удалиться). В качестве одного примера вторая часть пиксельного значения с управляющим кодом (т.е. часть, кодированная с использованием экспоненциального кода Голомба с параметром 3 , как пояснено выше) может усекается на основе определенного максимального квантованного пиксельного значения с управляющим кодом, как пояснено выше.

Это раскрытие сущности описывает способ извлечения максимального квантованного пиксельного значения с управляющим кодом. Согласно технологиям этого раскрытия сущности видеокoder и/или видеокoder могут определять блок, указывающий множество пикселей; определять максимальное квантованное пиксельное значение с управляющим кодом, по меньшей мере, частично на основе параметра квантования; и кодировать блок с использованием максимального квантованного пиксельного значения с управляющим кодом.

Например, технологии этого раскрытия сущности могут не использовать зависимые от кодера параметры для того, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом в декодере. Вместо этого технологии этого раскрытия сущности могут использовать только размер шага квантования при извлечении параметров. Кроме того, в некоторых примерах технологии этого раскрытия сущности могут использовать суммирование и сравнение вместо операций деления в ходе извлечения максимального квантованного пиксельного значения с управляющим кодом.

В качестве другого примера технологии этого раскрытия сущности могут использовать параметры кодера для того, чтобы извлекать максимальное квантованное значение с управляющим кодом в декодере. Таким образом, технологии этого раскрытия сущности могут сохранять связанные с кодером параметры в таблице поиска на стороне декодера. Кроме того, технологии этого раскрытия сущности могут использовать операции циклической обработки или сдвига для того, чтобы извлекать максимальное число битов, чтобы представлять квантованное значение с управляющим кодом.

В некоторых примерах технологии этого раскрытия сущности могут использовать параметры кодера для того, чтобы извлекать максимальное квантованное значение с управляющим кодом без операций отсечения и деления. Таким образом, технологии этого раскрытия сущности могут сохранять связанные

с кодером параметры в таблице поиска на стороне декодера. Как подробнее описано в этом раскрытии сущности, в некоторых примерах, технологии этого раскрытия сущности могут извлекать максимальное квантованное пиксельное значение с управляющим кодом с использованием параметра qP квантования и цветового компонента $bitDepth$. В частности, технологии этого раскрытия сущности могут получать значение $sMax$ с использованием просмотровой таблицы с циклической обработкой с qP и $bitDepth$.

Фиг. 1 является блок-схемой, иллюстрирующей примерную систему 10 кодирования видео, которая может использовать технологии этого раскрытия сущности. При использовании в данном документе термин "видеокодер" относится обобщенно к видеокодерам и видеодекодерам. В этом раскрытии сущности термины "кодирование видео" или "кодирование" могут относиться обобщенно к кодированию видео или декодированию видео. Видеокодер 20 и видеодекодер 30 системы 10 кодирования видео представляют примеры устройств, которые могут быть выполнены с возможностью осуществлять технологии для кодирования видео на основе палитр в соответствии с различными примерами, описанными в этом раскрытии сущности. Например, видеокодер 20 и видеодекодер 30 могут быть выполнены с возможностью избирательно кодировать различные блоки видеоданных, к примеру CU или PU при HEVC-кодировании, с использованием кодирования на основе палитр либо кодирования не на основе палитр. Режимы кодирования не на основе палитр могут означать различные режимы взаимного прогнозирующего временно-го кодирования или режимы внутреннего прогнозирующего пространственного кодирования, к примеру различные режимы кодирования, указываемые посредством HEVC-проекта 10.

Как показано на фиг. 1, система 10 кодирования видео включает в себя исходное устройство 12 и целевое устройство 14. Исходное устройство 12 формирует кодированные видеоданные. Соответственно исходное устройство 12 может упоминаться в качестве устройства кодирования видео или устройства кодирования видео. Целевое устройство 14 может декодировать кодированные видеоданные, сформированные посредством исходного устройства 12. Соответственно целевое устройство 14 может упоминаться в качестве устройства декодирования видео или устройства декодирования видео. Исходное устройство 12 и целевое устройство 14 могут быть примерами устройств кодирования видео или устройств кодирования видео.

Исходное устройство 12 и целевое устройство 14 могут содержать широкий диапазон устройств, включающих в себя настольные компьютеры, мобильные вычислительные устройства, ноутбуки (например, переносные компьютеры), планшетные компьютеры, абонентские приставки, телефонные трубки, к примеру так называемые смартфоны, телевизионные приемники, камеры, устройства отображения, цифровые мультимедийные проигрыватели, консоли для видеоигр, встроенные в автомобиль компьютеры и т.п.

Целевое устройство 14 может принимать кодированные видеоданные из исходного устройства 12 через канал 16. Канал 16 может содержать одну или более сред или устройств, допускающих перемещение кодированных видеоданных из исходного устройства 12 в целевое устройство 14. В одном примере канал 16 может содержать одну или более сред связи, которые предоставляют возможность исходному устройству 12 передавать кодированные видеоданные непосредственно в целевое устройство 14 в реальном времени. В этом примере исходное устройство 12 может модулировать кодированные видеоданные согласно стандарту связи, такому как протокол беспроводной связи, и может передавать модулированные видеоданные в целевое устройство 14. Одна или более сред связи могут включать в себя беспроводные среды связи и/или проводные среды связи, к примеру радиочастотный (RF) спектр или одну или более физических линий передачи. Одна или более сред связи могут составлять часть сети с коммутацией пакетов, такой как локальная вычислительная сеть, глобальная вычислительная сеть или глобальная сеть (например, Интернет). Одна или более сред связи могут включать в себя маршрутизаторы, коммутаторы, базовые станции или другое оборудование, которое упрощает передачу из исходного устройства 12 в целевое устройство 14.

В другом примере канал 16 может включать в себя носитель хранения данных, который сохраняет кодированные видеоданные, сформированные посредством исходного устройства 12. В этом примере целевое устройство 14 может осуществлять доступ к носителю хранения данных через доступ к диску или доступ по карте. Носитель хранения данных может включать в себя множество локально доступных носителей хранения данных, таких как Blu-Ray-диски, DVD, CD-ROM, флэш-память или другие подходящие цифровые носители хранения данных для сохранения кодированных видеоданных.

В дополнительном примере канал 16 может включать в себя файловый сервер или другое промежуточное устройство хранения данных, которое сохраняет кодированные видеоданные, сформированные посредством исходного устройства 12. В этом примере целевое устройство 14 может осуществлять доступ к кодированным видеоданным, сохраненным на файловом сервере или другом промежуточном устройстве хранения данных через потоковую передачу или загрузку. Файловый сервер может представлять собой тип сервера, допускающего сохранение кодированных видеоданных и передачу кодированных видеоданных в целевое устройство 14. Примерные файловые серверы включают в себя веб-серверы (например, для веб-узла), серверы по протоколу передачи файлов (FTP), устройства по протоколу системы хранения данных с подключением по сети (NAS) и локальные накопители на дисках.

Целевое устройство 14 может осуществлять доступ к кодированным видеоданным через стандартное подключение для передачи данных, к примеру Интернет-подключение. Примерные типы подключе-

ний для передачи данных могут включать в себя беспроводные каналы (например, Wi-Fi-подключения), проводные подключения (например, DSL, кабельный модем и т.д.) или комбинации вышеозначенного, которые являются подходящими для осуществления доступа к кодированным видеоданным, сохраненным на файловом сервере. Передача кодированных видеоданных из файлового сервера может представлять собой потоковую передачу, передачу на основе загрузки или комбинацию вышеозначенного.

Технологии этого раскрытия сущности не ограничены приложениями или настройками беспроводной связи. Технологии могут применяться к кодированию видео в поддержку множества мультимедийных приложений, таких как телевизионные широкоэмитательные передачи по радиointерфейсу, кабельные телевизионные передачи, спутниковые телевизионные передачи, потоковые передачи видео, например через Интернет, кодирование видеоданных для хранения на носителе хранения данных, декодирование видеоданных, сохраненных на носителе хранения данных или другие приложения. В некоторых примерах система 10 кодирования видео может быть выполнена с возможностью поддерживать одностороннюю или двустороннюю передачу видео, чтобы поддерживать такие приложения как потоковая передача видео, воспроизведение видео, широкоэмитательная передача видео и/или видеотелефония.

Система 10 кодирования видео, проиллюстрированная на фиг. 1, является просто примером, и технологии этого раскрытия сущности могут применяться к настройкам кодирования видео (например, кодирования видео или декодирования видео), которые не обязательно включают в себя передачу данных между устройствами кодирования и декодирования. В других примерах данные извлекаются из локального запоминающего устройства, передаются в потоковом режиме по сети и т.п. Устройство кодирования видео может кодировать и сохранять данные в запоминающем устройстве и/или устройство декодирования видео может извлекать и декодировать данные из запоминающего устройства. Во многих примерах кодирование и декодирование выполняется посредством устройств, которые не обмениваются данными друг с другом, а просто кодируют данные в запоминающее устройство и/или извлекают и декодируют данные из запоминающего устройства.

В примере по фиг. 1 исходное устройство 12 включает в себя видеоисточник 18, видеокодер 20 и интерфейс 22 вывода. В некоторых примерах, интерфейс 22 вывода может включать в себя модулятор/демодулятор (модем) и/или передающее устройство. Видеоисточник 18 может включать в себя устройство видеозахвата, например видеокамеру, видеоархив, содержащий ранее захваченные видеоданные, интерфейс прямых видеотрансляций, чтобы принимать видеоданные от поставщика видеоконтента, и/или компьютерную графическую систему для формирования видеоданных либо комбинацию таких источников видеоданных.

Видеокодер 20 может кодировать видеоданные из видеоисточника 18. В некоторых примерах исходное устройство 12 непосредственно передает кодированные видеоданные в целевое устройство 14 через интерфейс 22 вывода. В других примерах кодированные видеоданные также могут быть сохраняться на носителе хранения данных или файловом сервере для последующего доступа посредством целевого устройства 14 для декодирования и/или воспроизведения.

В примере по фиг. 1 целевое устройство 14 включает в себя интерфейс 28 ввода, видеодекодер 30 и устройство 32 отображения. В некоторых примерах интерфейс 28 ввода включает в себя приемное устройство и/или модем. Интерфейс 28 ввода может принимать кодированные видеоданные по каналу 16. Устройство 32 отображения может быть интегрировано или может быть внешним для целевого устройства 14. В общем, устройство 32 отображения отображает декодированные видеоданные. Устройство 32 отображения может содержать множество устройств отображения, таких как жидкокристаллический дисплей (LCD), плазменный дисплей, дисплей на органических светодиодах (OLED) или другой тип устройства отображения.

Это раскрытие сущности в общем может означать "передачу в служебных сигналах" или "передачу" определенной информации посредством видеокодера 20 в другое устройство, такое как видеодекодер 30. Термин "передача служебных сигналов" или "передача" в общем может означать передачу синтаксических элементов и/или других данных, используемых для того, чтобы декодировать сжатые видеоданные. Эта связь может осуществляться в реальном или практически в реальном времени. Альтернативно эта связь может осуществляться в промежутке времени, к примеру может осуществляться при сохранении синтаксических элементов на машиночитаемом носителе хранения данных в кодированном потоке битов во время кодирования, которые затем могут извлекаться посредством устройства декодирования видео в любое время после сохранения на этом носителе. Таким образом, тогда как видеодекодер 30 может упоминаться как "принимающий" определенную информацию, прием информации не обязательно осуществляется в реальном или практически в реальном времени и может извлекаться из носителя в определенное время после хранения.

Видеокодер 20 и видеодекодер 30 могут быть реализованы как любая из множества надлежащих схем, к примеру как один или более микропроцессоров, процессоров цифровых сигналов (DSP), специализированных интегральных схем (ASIC), программируемых пользователем вентильных матриц (FPGA), дискретная логика, аппаратные средства либо любые комбинации вышеозначенного. Если технологии реализуются частично в программном обеспечении, устройство может сохранять инструкции для программного обеспечения на подходящем энергонезависимом машиночитаемом носителе хранения данных

и может выполнять инструкции в аппаратных средствах с использованием одного или более процессоров, чтобы осуществлять технологии этого раскрытия сущности. Любое из вышеозначенного (включающее в себя аппаратные средства, программное обеспечение, комбинацию аппаратных средств и программного обеспечения и т.д.) может рассматриваться в качестве одного или более процессоров. Каждый из видеокодера 20 и видеодекодера 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 некоторых примерах видеокодер 20 и видеодекодер 30 работают согласно такому стандарту сжатия видео, как HEVC-стандарт, упомянутый выше и описанный в HEVC-проекте 10. В дополнение к базовому HEVC-стандарту прилагаются постоянные усилия для формирования расширений для масштабируемого кодирования видео, кодирования многовидового видео и трехмерного кодирования для HEVC. Помимо этого, режимы кодирования на основе палитр, например, как описано в этом раскрытии сущности, могут предоставляться для расширения HEVC-стандарта. В некоторых примерах технологии, описанные в этом раскрытии сущности для кодирования на основе палитр, могут применяться к кодерам и декодерам, выполненным с возможностью работы согласно другим стандартам кодирования видео, таким как ITU-T-H.264/AVC-стандарт или будущие стандарты. Соответственно применение режима кодирования на основе палитр для кодирования единиц кодирования (CU) или единиц прогнозирования (PU) в HEVC-кодеке описывается для целей примера.

В HEVC и других стандартах кодирования видео видеопоследовательность типично включает в себя последовательность изображений. Изображения также могут упоминаться в качестве "кадров". Изображение может включать в себя три массива выборок, обозначаемых как S_L , S_{Cb} и S_{Cr} . S_L представляет собой двумерный массив (т.е. блок) выборок сигнала яркости. S_{Cb} представляет собой двумерный массив выборок S_b -цветности. S_{Cr} представляет собой двумерный массив выборок S_r -цветности. Выборки цветности также могут упоминаться в данном документе как выборки "сигнала цветности". В других случаях изображение может быть монохромным и может включать в себя только массив выборок сигнала яркости.

Для того чтобы формировать кодированное представление изображения, видеокодер 20 может формировать набор единиц дерева кодирования (CTU). Каждая из CTU может представлять собой блок дерева кодирования выборок сигнала яркости, два соответствующих блока дерева кодирования выборок сигнала цветности и синтаксические структуры, используемые для того, чтобы кодировать выборки блоков дерева кодирования. Блок дерева кодирования может представлять собой блок $N \times N$ выборок. CTU также может упоминаться в качестве "древовидного блока" или наибольшей единицы кодирования (LCU). CTU HEVC в широком смысле могут быть аналогичными макроблокам других стандартов, таких как H.264/AVC. Тем не менее, CTU не обязательно ограничивается конкретным размером и может включать в себя одну или более единиц кодирования (CU). Серия последовательных макроблоков может включать в себя целое число CTU, упорядоченных последовательно в растровом сканировании. Кодированная серия последовательных макроблоков может содержать заголовок серии последовательных макроблоков и данные серии последовательных макроблоков. Заголовок серии последовательных макроблоков серии последовательных макроблоков может представлять собой синтаксическую структуру, которая включает в себя синтаксические элементы, которые предоставляют информацию относительно серии последовательных макроблоков. Данные серии последовательных макроблоков могут включать в себя кодированные CTU серии последовательных макроблоков.

Это раскрытие сущности может использовать термин "видеоединица", или "видеоблок", или "блок" для того, чтобы означать один или более блоков выборок и синтаксических структур, используемых для того, чтобы кодировать выборки одного или более блоков выборок. Примерные типы видеоединиц или блоков могут включать в себя CTU, CU, PU, единицы преобразования (TU), макроблоки, сегменты макроблока и т.д. В некоторых контекстах пояснение PU может заменяться на пояснение макроблоков или сегментов макроблока.

Для того чтобы формировать кодированную CTU, видеокодер 20 может рекурсивно выполнять сегментацию на дерево квадрантов для блоков дерева кодирования CTU для того, чтобы разделять блоки дерева кодирования на блоки кодирования, отсюда имя "единицы дерева кодирования". Блок кодирования представляет собой блок $N \times N$ выборок. CU может представлять собой блок кодирования выборок сигнала яркости и два соответствующих блока кодирования выборок сигнала цветности изображения, которое имеет массив выборок сигнала яркости, массив S_b -выборок и массив S_r -выборок, и синтаксические структуры, используемые для того, чтобы кодировать выборки блоков кодирования. Видеокодер 20 может сегментировать блок кодирования CU на один или более прогнозных блоков. Прогнозный блок может представлять собой прямоугольный (т.е. квадратный или неквадратный) блок выборок, к которым применяется идентичное прогнозирование. Единица прогнозирования (PU) CU может представлять собой прогнозный блок выборок сигнала яркости, два соответствующих прогнозных блока выборок сигнала цветности изображения и синтаксические структуры, используемые для того, чтобы прогнозировать выборки прогнозных блоков. Видеокодер 20 может формировать прогнозирующие блоки сигналов яркости, прогнозирующие S_b -блоки и прогнозирующие S_r -блоки для прогнозных блоков сигналов яркости,

прогнозных Сb-блоков и прогнозных Сг-блоков каждой PU CU.

Видеокодер 20 может использовать внутреннее прогнозирование или взаимное прогнозирование для того, чтобы формировать прогнозирующие блоки для PU. Если видеокодер 20 использует внутреннее прогнозирование для того, чтобы формировать прогнозирующие блоки PU, видеокодер 20 может формировать прогнозирующие блоки PU на основе декодированных выборок изображения, ассоциированного с PU.

Если видеокодер 20 использует взаимное прогнозирование для того, чтобы формировать прогнозирующие блоки PU, видеокодер 20 может формировать прогнозирующие блоки PU на основе декодированных выборок одного или более изображений, за исключением изображения, ассоциированного с PU. Видеокодер 20 может использовать унипрогнозирование или бипрогнозирование для того, чтобы формировать прогнозирующие блоки PU. Когда видеокодер 20 использует унипрогнозирование для того, чтобы формировать прогнозирующие блоки для PU, PU может иметь один вектор движения (MV). Когда видеокодер 20 использует бипрогнозирование для того, чтобы формировать прогнозирующие блоки для PU, PU может иметь два MV.

После того как видеокодер 20 формирует прогнозирующие блоки (например, прогнозирующие блоки сигналов яркости, прогнозирующие Сb-блоки и прогнозирующие Сг-блоки) для одной или более PU CU, видеокодер 20 может формировать остаточные блоки для CU. Каждая выборка в остаточном блоке CU может указывать разность между выборкой в прогнозирующем блоке PU CU и соответствующей выборкой в блоке кодирования CU. Например, видеокодер 20 может формировать остаточный блок сигналов яркости для CU. Каждая выборка в остаточном блоке сигналов яркости CU указывает разность между выборкой сигнала яркости в одном из прогнозирующих блоков сигналов яркости CU и соответствующей выборкой в исходном блоке кодирования сигналов яркости CU. Помимо этого, видеокодер 20 может формировать остаточный Сb-блок для CU. Каждая выборка в остаточном Сb-блоке CU может указывать разность между Сb-выборкой в одном из прогнозирующих Сb-блоков CU и соответствующей выборкой в исходном Сb-блоке кодирования CU. Видеокодер 20 также может формировать остаточный Сг-блок для CU. Каждая выборка в остаточном Сг-блоке CU может указывать разность между Сг-выборкой в одном из прогнозирующих Сг-блоков CU и соответствующей выборкой в исходном Сг-блоке кодирования CU.

Кроме того, видеокодер 20 может использовать сегментацию на дерево квадрантов для того, чтобы разлагать остаточные блоки (например, остаточные блоки сигналов яркости, остаточные Сb-блоки и остаточные Сг-блоки) CU на один или более блоков преобразования (например, на блоки преобразования сигналов яркости, Сb-блоки преобразования и Сг-блоки преобразования). Блок преобразования может представлять собой прямоугольный блок выборок, к которым применяется идентичное преобразование. Единица преобразования (TU) CU может представлять собой блок преобразования выборок сигнала яркости, два соответствующих блока преобразования выборок сигнала цветности и синтаксические структуры, используемые для того, чтобы преобразовывать выборки блока преобразования. Таким образом, каждая TU CU может быть ассоциирована с блоком преобразования сигналов яркости, Сb-блоком преобразования и Сг-блоком преобразования. Блок преобразования сигналов яркости, ассоциированный с TU, может представлять собой субблок остаточного блока сигналов яркости CU. Сб-блок преобразования может представлять собой субблок остаточного Сb-блока CU. Сг-блок преобразования может представлять собой субблок остаточного Сг-блока CU.

Видеокодер 20 может применять одно или более преобразований к блоку преобразования для того, чтобы формировать блок коэффициентов для TU. Блок коэффициентов может представлять собой двумерный массив коэффициентов преобразования. Коэффициент преобразования может быть скалярной величиной. Например, видеокодер 20 может применять одно или более преобразований к блоку преобразования сигналов яркости TU для того, чтобы формировать блок коэффициентов сигнала яркости для TU. Видеокодер 20 может применять одно или более преобразований к Сb-блоку преобразования TU для того, чтобы формировать Сb-блок коэффициентов для TU. Видеокодер 20 может применять одно или более преобразований к Сг-блоку преобразования TU для того, чтобы формировать Сг-блок коэффициентов для TU.

После формирования блока коэффициентов (например, блока коэффициентов сигнала яркости, Сb-блока коэффициентов или Сг-блока коэффициентов) видеокодер 20 может квантовать блок коэффициентов. Квантование в общем означает процесс, в котором коэффициенты преобразования квантуются, чтобы, возможно, уменьшать объем данных, используемых для того, чтобы представлять коэффициенты преобразования, обеспечивая дополнительное сжатие. После того как видеокодер 20 квантует блок коэффициентов, видеокодер 20 может энтропийно кодировать синтаксические элементы, указывающие квантованные коэффициенты преобразования. Например, видеокодер 20 может выполнять контекстно-адаптивное двоичное арифметическое кодирование (САВАС) для синтаксических элементов, указывающих квантованные коэффициенты преобразования. Видеокодер 20 может выводить энтропийно кодированные синтаксические элементы в потоке битов. Поток битов также может включать в себя синтаксические элементы, которые не кодируются энтропийно.

Видеокодер 20 может выводить поток битов, который включает в себя энтропийно кодированные синтаксические элементы. Поток битов может включать в себя последовательность битов, которая формирует представление кодированных изображений и ассоциированных данных. Поток битов может со-

держат последовательность единиц уровня абстрагирования от сети (NAL). Каждая из NAL-единиц включает в себя заголовок NAL-единицы и инкапсулирует первичную байтовую последовательность данных (RBSP). Заголовок NAL-единицы может включать в себя синтаксический элемент, который указывает код типа NAL-единицы. Код типа NAL-единицы, указываемый посредством заголовка NAL-единицы для NAL-единицы, указывает тип NAL-единицы. RBSP может представлять собой синтаксическую структуру, содержащую целое число байтов, которое инкапсулируется в NAL-единице. В некоторых случаях RBSP включает в себя нулевые биты.

Различные типы NAL-единиц могут инкапсулировать различные типы RBSP. Например, первый тип NAL-единицы может инкапсулировать RBSP для набора параметров изображения (PPS), второй тип NAL-единицы может инкапсулировать RBSP для кодированной серии последовательных макроблоков, третий тип NAL-единицы может инкапсулировать RBSP для дополнительной улучшающей информации (SEI) и т.д. NAL-единицы, которые инкапсулируют RBSP для данных кодирования видео (в противоположность RBSP для наборов параметров и SEI-сообщений), могут упоминаться в качестве NAL-единиц слоя кодирования видео (VCL).

Видеодекoder 30 может принимать поток битов, сформированный посредством видеокodера 20. Помимо этого, видеодекoder 30 может получать синтаксические элементы из потока битов. Например, видеодекoder 30 может синтаксически анализировать поток битов, чтобы декодировать синтаксические элементы из потока битов. Видеодекoder 30 может восстанавливать изображения видеоданных, по меньшей мере, частично на основе синтаксических элементов, полученных (к примеру, декодированных) из потока битов. Процесс для того, чтобы восстанавливать видеоданные, в общем может быть обратным по отношению к процессу, выполняемому посредством видеокodера 20. Например, видеодекoder 30 может использовать MV PU для того, чтобы определять прогнозирующие блоки выборки (т.е. прогнозирующие блоки) для PU текущей CU. Помимо этого, видеодекoder 30 может обратно квантовать блоки коэффициентов преобразования, ассоциированные с TU текущей CU. Видеодекoder 30 может выполнять обратные преобразования для блоков коэффициентов преобразования для того, чтобы восстанавливать блоки преобразования, ассоциированные с TU текущей CU. Видеодекoder 30 может восстанавливать блоки кодирования текущей CU посредством суммирования выборок прогнозирующих блоков выборки для PU текущей CU с соответствующими выборками блоков преобразования TU текущей CU. Посободством восстановления блоков кодирования для каждой CU изображения видеодекoder 30 может восстанавливать изображение.

В некоторых примерах видеокoder 20 и видеодекoder 30 могут быть выполнены с возможностью осуществлять кодирование на основе палитр. Например, при кодировании на основе палитр, вместо выполнения технологий внутреннего прогнозирующего или взаимного прогнозирующего кодирования, описанных выше, видеокoder 20 и видеодекoder 30 могут кодировать так называемую палитру в качестве таблицы цветов или пиксельных значений, представляющих видеоданные конкретной области (например, данного блока). Таким образом, вместо кодирования фактических пиксельных значений или их остатков для текущего блока видеоданных видеокoder может кодировать значения индекса для одного или более пиксельных значений текущего блока, причем значения индекса указывают записи в палитре, которые используются для того, чтобы представлять пиксельные значения текущего блока.

Например, видеокoder 20 может кодировать блок видеоданных посредством определения палитры для блока, нахождения записи в палитре, имеющей значение, представляющее значение одного или более отдельных пикселей блока, и кодирования блока с помощью значений индекса, которые указывают запись в палитре, используемую для того, чтобы представлять одно или более отдельных пиксельных значений блока. Дополнительно видеокoder 20 может передавать в служебных сигналах значения индекса в кодированном потоке битов. В свою очередь, устройство декодирования видео (например, видеодекoder 30) может получать из кодированного потока битов палитру для блока, а также значения индекса, используемые для определения различных отдельных пикселей блока с использованием палитры. Видеодекoder 30 может увязывать значения индекса отдельных пикселей с записями палитры, чтобы восстанавливать пиксельные значения блока. В случаях когда значение индекса, ассоциированное с отдельным пикселем, не увязывается ни с одним значением индекса соответствующей палитры для блока, видеодекoder 30 может идентифицировать такой пиксел в качестве пиксела с управляющим кодом в целях кодирования на основе палитр.

В другом примере видеокoder 20 может кодировать блок видеоданных согласно следующим операциям. Видеокoder 20 может определять значения остатка прогнозирования для отдельных пикселей блока, определять палитру для блока и находить запись (например, значение индекса) в палитре, имеющую значение, представляющее значение одного или более значений остатка прогнозирования отдельных пикселей. Дополнительно видеокoder 20 может кодировать пиксели блока со значениями индекса, которые указывают запись в палитре, используемую для того, чтобы представлять соответствующее значение остатка прогнозирования для каждого отдельного пиксела блока. Видеодекoder 30 может получать из кодированного потока битов, передаваемого в служебных сигналах посредством исходного устройства 12, палитру для блока, а также значения индекса для значений остатка прогнозирования, соответствующих отдельным пикселям блока. Как описано выше, значения индекса могут соответствовать записям в

палитре, ассоциированной с текущим блоком. В свою очередь, видеodeкодер 30 может связывать значения индекса остаточных прогнозных значений с записями палитры, чтобы восстанавливать остаточные прогнозныe значения блока. Остаточные прогнозныe значения могут суммироваться с прогнозными значениями (например, полученными с использованием внутреннего или взаимного прогнозирования), чтобы восстанавливать пиксельные значения блока.

Как подробнее описано ниже, базовая идея касательно кодирования на основе палитр состоит в том, что для блока видеоданных, которые должны кодироваться, видеodeкодер 20 может извлекать палитру, которая включает в себя наиболее доминирующие пиксельные значения в блоке. Например, палитра может означать число пиксельных значений, которые определяются или допускаются как доминирующие и/или характерные для текущей CU. Видеodeкодер 20 может кодировать данные, представляющие размер и элементы палитры. Дополнительно видеodeкодер 20 может кодировать пиксельные значения в блоке согласно определенному порядку сканирования. Для каждого пикселя, включенного в блок, видеodeкодер 20 может определять то, включено или нет значение пикселя в палитру. Если пиксел включен в палитру, видеodeкодер 20 может передавать в служебных сигналах значение индекса, которое уязвляет пиксельное значение с соответствующей записью в палитре. Дополнительно видеodeкодер 20 может передавать в служебных сигналах значение серии, указывающее то, сколько пикселей после пикселя, для которого передан в служебных сигналах индекс, имеет идентичное значение. В качестве альтернативы, если последовательности пикселей имеют значения, которые являются идентичными значениями из соседних пикселей выше, видеodeкодер 20 может кодировать пиксели с использованием режима "с копированием сверху", в котором значение серии (но не значение индекса) указывает число пикселей, которые совместно используют значение с пикселями сверху.

С другой стороны, если значение текущего пикселя блока не включено в палитру (т.е. не существует записи палитры, которая указывает конкретное пиксельное значение закодированного в палитровом режиме блока) и пиксел не использует значение совместно с соседним сверху пикселем, то такой пиксел может задаваться как "пиксел с управляющим кодом". В соответствии с кодированием на основе палитр, видеodeкодер 20 может кодировать и передавать в служебных сигналах значение индекса, которое зарезервировано для пикселя с управляющим кодом. Например, значение индекса, представляющее пиксел с управляющим кодом, может представлять собой размер палитры при условии, что палитра имеет нулевой индекс. Иными словами, для палитры, имеющей четыре записи с индексами от нуля до трех, видеodeкодер 20 может передавать в служебных сигналах значение четыре (4) для пикселя, чтобы указывать то, что пиксел закодирован как пиксел с управляющим кодом. В некоторых примерах видеodeкодер 20 также может кодировать и передавать в служебных сигналах пиксельное значение или остаточное значение (либо их квантованные версии) для пикселя с управляющим кодом, включенного в блок.

В соответствии с технологиями этого раскрытия сущности видеodeкодер 20 может кодировать значение пикселя с управляющим кодом в двух частях, первой части и второй части. Первая и вторая части при конкатенации могут представлять значение пикселя с управляющим кодом. Видеodeкодер 20 может кодировать первую часть с использованием кодового слова Голомба-Райса и вторую часть с использованием кодового слова с экспоненциальным кодом Голомба с параметром 3.

После приема закодированного потока видеобитов, передаваемого в служебных сигналах посредством видеodeкодера 20, видеodeкодер 30 может сначала определять палитру на основе информации, принимаемой из видеodeкодера 20. Видеodeкодер 30 затем может уязвывать принятые значения индекса, ассоциированные с пиксельными местоположениями в данном блоке, с записями палитры, чтобы восстанавливать пиксельные значения данного блока. В некоторых случаях видеodeкодер 30 может определять то, что пиксел закодированного в палитровом режиме блока представляет собой пиксел с управляющим кодом, к примеру посредством определения того, что пиксел кодируется в палитровом режиме со значением индекса, зарезервированным для пикселей с управляющим кодом. В случаях когда видеodeкодер 30 идентифицирует пиксел с управляющим кодом в закодированном в палитровом режиме блоке, видеodeкодер 30 может принимать пиксельное значение или остаточное значение (либо их квантованные версии) для пикселя с управляющим кодом, включенного в данный блок. Видеodeкодер 30 может восстанавливать закодированный в палитровом режиме блок посредством уязвки отдельных пиксельных значений с соответствующими записями палитры и посредством использования пиксельного значения или остаточного значения (либо их квантованных версий), чтобы восстанавливать все пиксели с управляющим кодом, включенные в закодированный в палитровом режиме блок.

Видеodeкодер 20 и видеodeкодер 30 могут быть выполнены с возможностью работать согласно технологиям, описанным в этом раскрытии сущности, чтобы извлекать максимальное квантованное пиксельное значение с управляющим кодом. Например, видеodeкодер 20 и видеodeкодер 30 могут определять блок, указывающий множество пикселей; определять максимальное квантованное пиксельное значение с управляющим кодом, по меньшей мере, частично на основе параметра квантования, и кодировать блок с использованием максимального квантованного пиксельного значения с управляющим кодом.

В некоторых примерах определение максимального квантованного пиксельного значения с управляющим кодом дополнительно содержит определение параметра квантования на основе цветового компонента; определение размера шага квантования на основе параметра квантования; определение макси-

мального возможного квантованного значения на основе размера шага квантования; и формирование усеченного двоичного кодового слова, чтобы кодировать квантованные пиксели с управляющим кодом, на основе максимального квантованного пиксельного значения с управляющим кодом. В некоторых примерах максимальное квантованное пиксельное значение с управляющим кодом определяется с использованием только шага квантования. В некоторых примерах определение максимального квантованного пиксельного значения с управляющим кодом не основано на зависимом от кодера параметре.

В некоторых примерах технологии этого раскрытия сущности включают в себя то, что определение максимального квантованного пиксельного значения с управляющим кодом выполняется без использования операций деления. В некоторых примерах определение максимального квантованного пиксельного значения с управляющим кодом выполняется с использованием суммирования и сравнения. В некоторых примерах определение максимального квантованного пиксельного значения с управляющим кодом дополнительно содержит определение параметра квантования на основе цветового компонента; определение, на основе параметра квантования, по меньшей мере одного из коэффициента масштабирования при квантовании или параметра сдвига; определение максимального возможного квантованного значения на основе по меньшей мере одного из коэффициента масштабирования при квантовании или параметра сдвига; и формирование усеченного двоичного кодового слова, чтобы кодировать квантованные пиксели с управляющим кодом, на основе максимального квантованного пиксельного значения с управляющим кодом.

В некоторых примерах технологии этого раскрытия сущности включают в себя определение параметра размера шага квантования на основе параметра квантования; и при этом максимальное возможное квантованное значение дополнительно определяется на основе размера шага квантования. В некоторых примерах по меньшей мере один из коэффициента масштабирования при квантовании или параметра сдвига представляет собой параметр кодера. В некоторых примерах технологии этого раскрытия сущности включают в себя сохранение по меньшей мере одного из коэффициента масштабирования при квантовании или параметра сдвига в таблице поиска; и при этом определение на основе параметра квантования по меньшей мере одного из коэффициента масштабирования при квантовании или параметра сдвига дополнительно содержит выполнение поиска в таблице поиска, чтобы определять по меньшей мере один из коэффициента масштабирования при квантовании или параметра сдвига.

В некоторых примерах определение максимального квантованного пиксельного значения с управляющим кодом дополнительно включает в себя определение параметра квантования на основе цветового компонента; определение базового параметра квантования на основе битовой глубины; определение того, удовлетворяет или нет базовый параметр квантования пороговому значению; и когда базовый параметр квантования удовлетворяет пороговому значению, определение максимального возможного квантованного значения из таблицы поиска и на основе базового параметра квантования, и когда базовый параметр квантования не удовлетворяет пороговому значению, определение максимального возможного квантованного значения из таблицы поиска и на основе битовой глубины.

В некоторых примерах квантованные пиксели с управляющим кодом кодируются в кодовом слове фиксированной длины. В некоторых примерах кодовое слово, которое включает в себя максимальное квантованное пиксельное значение с управляющим кодом, формируется без ограничения по максимальному значению. В некоторых примерах максимальное квантованное пиксельное значение с управляющим кодом определяется, по меньшей мере, без выполнения операции деления или без использования зависимого от кодера параметра.

Фиг. 2 является блок-схемой, иллюстрирующей примерный видеокодер 20, который может реализовывать различные технологии этого раскрытия сущности. Фиг. 2 предоставляется для целей пояснения и не должен считаться ограничением технологий, как проиллюстрировано и описано в общих чертах в этом раскрытии сущности. Для целей пояснения, это раскрытие сущности описывает видеокодер 20 в контексте HEVC-кодирования. Тем не менее, технологии этого раскрытия сущности могут быть применимыми к другим стандартам или способам кодирования.

В примере по фиг. 2 видеокодер 20 включает в себя запоминающее устройство 98 видеоданных, модуль 100 обработки прогнозирования, модуль 102 формирования остатков, модуль 104 обработки преобразования, модуль 106 квантования, модуль 108 обратного квантования, модуль 110 обработки обратного преобразования, модуль 112 восстановления, модуль 114 фильтрации, буфер 116 декодированных изображений и модуль 118 энтропийного кодирования. Модуль 100 обработки прогнозирования включает в себя модуль 120 обработки взаимного прогнозирования и модуль 126 обработки внутреннего прогнозирования. Модуль 120 обработки взаимного прогнозирования включает в себя модуль оценки движения и модуль компенсации движения (не показаны). Видеокодер 20 также включает в себя модуль 122 кодирования на основе палитр, выполненный с возможностью осуществлять различные аспекты технологий кодирования на основе палитр, описанных в этом раскрытии сущности. В других примерах видеокодер 20 может включать в себя большее, меньшее число или другие функциональные компоненты.

Запоминающее устройство 98 видеоданных может сохранять видеоданные, которые должны кодироваться посредством компонентов видеокодера 20. Видеоданные, сохраненные в запоминающем устройстве 98 видеоданных, могут получаться, например, из видеисточника 18. Буфер 116 декодированных

изображений может представлять собой запоминающее устройство опорных изображений, которое сохраняет опорные видеоданные для использования при кодировании видеоданных посредством видеокодера 20, например в режимах внутреннего или взаимного кодирования. Запоминающее устройство 98 видеоданных и буфер 116 декодированных изображений могут формироваться посредством любого из множества запоминающих устройств, к примеру, как динамическое оперативное запоминающее устройство (DRAM), включающее в себя синхронное DRAM (SDRAM), магниторезистивное RAM (MRAM), резистивное RAM (RRAM) или другие типы запоминающих устройств. Запоминающее устройство 98 видеоданных и буфер 116 декодированных изображений могут предоставляться посредством идентичного запоминающего устройства или отдельных запоминающих устройств. В различных примерах запоминающее устройство 98 видеоданных может быть внутрикристалльным с другими компонентами видеокодера 20 или внекристалльным относительно этих компонентов.

Видеокодер 20 может принимать видеоданные. Видеокодер 20 может кодировать каждую CTU в серии последовательных макроблоков изображения видеоданных. Каждая из CTU может быть ассоциирована с блоками дерева кодирования (CTB) сигнала яркости одинакового размера и соответствующими CTB изображения. В качестве части кодирования CTU модуль 100 обработки прогнозирования может выполнять сегментацию на дерево квадрантов для того, чтобы разделять CTB CTU на постепенно меньшие блоки. Меньший блок может представлять собой блоки кодирования CU. Например, модуль 100 обработки прогнозирования может сегментировать CTB, ассоциированный с CTU, на четыре субблока одинакового размера, сегментировать один или более субблоков на четыре субсубблока одинакового размера и т.д.

Видеокодер 20 может кодировать CU CTU для того, чтобы формировать закодированные представления CU (т.е. закодированные CU). В качестве части кодирования CU модуль 100 обработки прогнозирования может сегментировать блоки кодирования, ассоциированные с CU, из числа одной или более PU CU. Таким образом, каждая PU может быть ассоциирована с прогнозным блоком сигналов яркости и соответствующими прогнозными блоками сигнала цветности. Видеокодер 20 и видеодекодер 30 могут поддерживать PU, имеющие различные размеры. Как указано выше, размер CU может означать размер блока кодирования сигналов яркости CU и размер PU может означать размер прогнозного блока сигналов яркости PU. При условии, что размер конкретной CU составляет $2N \times 2N$, видеокодер 20 и видеодекодер 30 могут поддерживать PU-размеры $2N \times 2N$ или $N \times N$ для внутреннего прогнозирования и симметричные PU-размеры $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ или аналогичные для взаимного прогнозирования. Видеокодер 20 и видеодекодер 30 также могут поддерживать асимметричное сегментирование для PU-размеров $2N \times nU$, $2N \times nD$, $nL \times 2N$ и $nR \times 2N$ для взаимного прогнозирования.

Модуль 120 обработки взаимного прогнозирования может формировать прогнозирующие данные для PU посредством выполнения взаимного прогнозирования для каждой PU CU. Прогнозирующие данные для PU могут включать в себя один или более прогнозирующих блоков выборок PU и информацию движения для PU. Модуль 121 взаимного прогнозирования может выполнять различные операции для PU CU в зависимости от того, находится PU в серии последовательных I-макроблоков, серии последовательных P-макроблоков или серии последовательных B-макроблоков. В серии последовательных I-макроблоков все PU внутренне прогнозируются. Следовательно, если PU находится в серии последовательных I-макроблоков, модуль 121 взаимного прогнозирования не выполняет взаимное прогнозирование для PU. Таким образом, для блоков, закодированных в I-режиме, прогнозирующий блок формируется с использованием пространственного прогнозирования из ранее закодированных соседних блоков в идентичном кадре.

Если PU находится в серии последовательных P-макроблоков, модуль оценки движения модуля 120 обработки взаимного прогнозирования может выполнять поиск в опорных изображениях в списке опорных изображений (например, в RefPicList0) на предмет опорной области для PU. Опорная область для PU может представлять собой область в опорном изображении, которая содержит блоки выборок, которые наиболее близко соответствуют блокам выборок PU. Модуль оценки движения может формировать опорный индекс, который указывает позицию в RefPicList0 опорного изображения, содержащего опорную область для PU. Помимо этого, модуль оценки движения может формировать MV, который указывает пространственное смещение между блоком кодирования PU и опорным местоположением, ассоциированным с опорной областью. Например, MV может представлять собой двумерный вектор, который предоставляет смещение от координат в текущем декодированном изображении до координат в опорном изображении. Модуль оценки движения может выводить опорный индекс и MV в качестве информации движения PU. Модуль компенсации движения модуля 120 обработки взаимного прогнозирования может формировать прогнозирующие блоки выборок PU на основе фактических или интерполированных выборок в опорном местоположении, указываемом посредством вектора движения PU.

Если PU находится в серии последовательных B-макроблоков, модуль оценки движения может выполнять унипрогнозирование или бипрогнозирование для PU. Чтобы выполнять унипрогнозирование для PU, модуль оценки движения может выполнять поиск в опорных изображениях RefPicList0 или второго списка опорных изображений (RefPicList1) на предмет опорной области для PU. Модуль оценки движе-

ния может выводить в качестве информации движения PU опорный индекс, который указывает позицию в RefPicList0 или RefPicList1 опорного изображения, которое содержит опорную область, MV, который указывает пространственное смещение между блоком выборок PU и опорным местоположением, ассоциированным с опорной областью, и один или более индикаторов направления прогнозирования, которые указывают то, находится опорное изображение в RefPicList0 или в RefPicList1. Модуль компенсации движения модуля 120 обработки взаимного прогнозирования может формировать прогнозирующие блоки выборок PU, по меньшей мере, частично на основе фактических или интерполированных выборок в опорной области, указываемой посредством вектора движения PU.

Чтобы выполнять двунаправленное взаимное прогнозирование для PU, модуль оценки движения может выполнять поиск в опорных изображениях RefPicList0 на предмет опорной области для PU, а также может выполнять поиск в опорных изображениях RefPicList1 на предмет другой опорной области для PU. Модуль оценки движения может формировать индексы опорных изображений, которые указывают позиции в RefPicList0 и RefPicList1 опорных изображений, которые содержат опорные области. Помимо этого, модуль оценки движения может формировать MV, которые указывают пространственные смещения между опорным местоположением, ассоциированным с опорными областями, и блоком выборок PU. Информация движения PU может включать в себя опорные индексы и MV PU. Модуль компенсации движения может формировать прогнозирующие блоки выборок PU, по меньшей мере, частично на основе фактических или интерполированных выборок в опорной области, указываемой посредством вектора движения PU.

В соответствии с различными примерами этого раскрытия сущности видеокодер 20 может быть выполнен с возможностью осуществлять кодирование на основе палитр. Относительно инфраструктуры HEVC в качестве примера, технологии кодирования на основе палитр могут быть выполнены с возможностью использоваться в качестве CU-режима. В других примерах технологии кодирования на основе палитр могут быть выполнены с возможностью использоваться в качестве PU-режима в инфраструктуре HEVC. Соответственно все раскрытые процессы, описанные в данном документе (в ходе этого раскрытия сущности) в контексте CU-режима, дополнительно или альтернативно могут применяться к PU-режиму. Тем не менее, эти примеры на основе HEVC не должны считаться ограничением или недочетом технологий кодирования на основе палитр, описанных в данном документе, поскольку технологии в данном документе могут применяться, чтобы работать независимо либо в качестве части других существующих или будущих разрабатываемых систем/стандартов. В этих случаях единица для палитрового кодирования может представлять собой квадратные блоки, прямоугольные блоки или даже области непрямоугольной формы.

Модуль 122 кодирования на основе палитр, например, может выполнять кодирование на основе палитр, когда режим кодирования на основе палитр выбирается, например, для CU или PU. Например, модуль 122 кодирования на основе палитр может быть выполнен с возможностью формировать палитру, имеющую записи, указывающие пиксельные значения, выбирать пиксельные значения в палитре, чтобы представлять пиксельные значения, по меньшей мере, некоторых позиций блока видеоданных и передавать в служебных сигналах информацию, такую как значения индекса, ассоциирующую, по меньшей мере, некоторые позиции блока видеоданных с записями в палитре, соответствующими надлежащим образом выбранным пиксельным значениям. Хотя различные функции описываются как выполняемые посредством модуля 122 кодирования на основе палитр, некоторые или все такие функции могут выполняться посредством других модулей обработки или комбинации различных модулей обработки.

Модуль 122 кодирования на основе палитр может быть выполнен с возможностью формировать любой из различных синтаксических элементов, описанных в данном документе. Соответственно видеокодер 20 может быть выполнен с возможностью кодировать блоки видеоданных с использованием режимов кодирования на основе палитр, как описано в этом раскрытии сущности. Видеокодер 20 может избирательно кодировать блок видеоданных с использованием режима палитрового кодирования или кодировать блок видеоданных с использованием другого режима, например такого режима взаимного прогнозирующего или внутреннего прогнозирующего HEVC-кодирования. Блок видеоданных, например, может представлять собой CU или PU, сформированную согласно процессу HEVC-кодирования. Видеокодер 20 может кодировать некоторые блоки с помощью режимов взаимного прогнозирующего временного прогнозирования или внутреннего прогнозирующего пространственного кодирования и декодировать другие блоки с помощью режима кодирования на основе палитр.

Модуль 126 обработки внутреннего прогнозирования может формировать прогнозирующие данные для PU посредством выполнения внутреннего прогнозирования для PU. Прогнозирующие данные для PU могут включать в себя прогнозирующие блоки выборок для PU и различные синтаксические элементы. Модуль 126 обработки внутреннего прогнозирования может выполнять внутреннее прогнозирование для PU в сериях последовательных I-макроблоков, сериях последовательных P-макроблоков и сериях последовательных B-макроблоков.

Чтобы выполнять внутреннее прогнозирование для PU, модуль 126 обработки внутреннего прогнозирования может использовать несколько режимов внутреннего прогнозирования для того, чтобы формировать несколько наборов прогнозирующих данных для PU. При использовании некоторых режимов

внутреннего прогнозирования для того, чтобы формировать набор прогнозирующих данных для PU, модуль 126 обработки внутреннего прогнозирования может расширять значения выборок из блоков выборок соседних PU через прогнозирующие блоки PU в направлениях, ассоциированных с режимами внутреннего прогнозирования. Соседние PU могут располагаться выше, выше и справа, выше и слева или слева от PU при условии порядка кодирования слева направо, сверху вниз для PU, CU и STU. Модуль 126 обработки внутреннего прогнозирования может использовать различные числа режимов внутреннего прогнозирования, например 33 режима направленного внутреннего прогнозирования. В некоторых примерах число режимов внутреннего прогнозирования может зависеть от размера области, ассоциированной с PU.

Модуль 100 обработки прогнозирования может выбирать прогнозирующие данные для PU CU из числа прогнозирующих данных, сформированных посредством модуля 120 обработки взаимного прогнозирования для PU, или прогнозирующих данных, сформированных посредством модуля 126 обработки внутреннего прогнозирования для PU. В некоторых примерах модуль 100 обработки прогнозирования выбирает прогнозирующие данные для PU CU на основе показателей искажения в зависимости от скорости передачи наборов прогнозирующих данных. Прогнозирующие блоки выборок для выбранных прогнозирующих данных могут упоминаться в данном документе как выбранные прогнозирующие блоки выборок.

Модуль 102 формирования остатков может формировать, на основе блоков кодирования (к примеру, блоков кодирования сигналов яркости, Сb-блоков кодирования и Сr-блоков кодирования) CU и выбранных прогнозирующих блоков выборок (к примеру, блоков сигналов яркости, прогнозирующих Сb-блоков и прогнозирующих Сr-блоков) PU CU, остаточные блоки (к примеру, остаточные блоки сигналов яркости, остаточные Сb-блоки и остаточные Сr-блоки) CU. Например, модуль 102 формирования остатков может формировать остаточные блоки CU таким образом, что каждая выборка в остаточных блоках имеет значение, равное разности между выборкой в блоке кодирования CU и соответствующей выборкой в соответствующем выбранном прогнозирующем блоке выборок PU CU.

Модуль 104 обработки преобразования может выполнять сегментацию на дерево квадрантов для того, чтобы сегментировать остаточные блоки, ассоциированные с CU, на блоки преобразования, ассоциированные с TU CU. Таким образом, в некоторых примерах, TU может быть ассоциирована с блоком преобразования сигналов яркости и двумя блоками преобразования сигналов цветности. Размеры и позиции блоков преобразования сигналов яркости и сигналов цветности TU CU могут быть основаны либо могут не быть основаны на размерах и позициях прогнозных блоков PU CU. Структура в виде дерева квадрантов, известная как "остаточное дерево квадрантов" (RQT), может включать в себя узлы, ассоциированные с каждой из областей. TU CU могут соответствовать концевым узлам RQT.

Модуль 104 обработки преобразования может формировать блоки коэффициентов преобразования для каждой TU CU посредством применения одного или более преобразований к блокам преобразования TU. Модуль 104 обработки преобразования может применять различные преобразования к блоку преобразования, ассоциированному с TU. Например, модуль 104 обработки преобразования может применять дискретное косинусное преобразование (DCT), направленное преобразование или концептуально аналогичное преобразование к блоку преобразования. В некоторых примерах модуль 104 обработки преобразования не применяет преобразования к блоку преобразования. В таких примерах блок преобразования может трактоваться в качестве блока коэффициентов преобразования.

Модуль 106 квантования может квантовать коэффициенты преобразования в блоке коэффициентов. Процесс квантования может уменьшать битовую глубину, ассоциированную с некоторыми или всеми коэффициентами преобразования. Например, n-битовый коэффициент преобразования может округляться в меньшую сторону до m-битового коэффициента преобразования во время квантования, где n превышает m. Модуль 106 квантования может квантовать блок коэффициентов, ассоциированный с TU CU, на основе значения параметра квантования (QP), ассоциированного с CU. Видеокодер 20 может регулировать степень квантования, применяемого к блокам коэффициентов, ассоциированным с CU, посредством регулирования QP-значения, ассоциированного с CU. Квантование может вводить потери информации, в силу чего квантованные коэффициенты преобразования могут иметь меньшую точность по сравнению с исходными коэффициентами преобразования.

Модуль 108 обратного квантования и модуль 110 обработки обратного преобразования могут применять обратное квантование и обратные преобразования к блоку коэффициентов соответственно для того, чтобы восстанавливать остаточный блок из блока коэффициентов. Модуль 112 восстановления может суммировать восстановленный остаточный блок с соответствующими выборками из одного или более прогнозирующих блоков выборок, сформированных посредством модуля 100 обработки прогнозирования для того, чтобы формировать восстановленный блок преобразования, ассоциированный с TU. Посредством восстановления блоков преобразования для каждой TU CU таким способом видеокодер 20 может восстанавливать блоки кодирования CU.

Модуль 114 фильтрации может выполнять одну или более операций удаления блочности, чтобы уменьшать артефакты блочности в блоках кодирования, ассоциированных с CU. Буфер 116 декодированных изображений может сохранять восстановленные блоки кодирования после того, как модуль 114

фильтрации выполняет одну или более операций удаления блочности для восстановленных блоков кодирования. Модуль 120 обработки взаимного прогнозирования может использовать опорное изображение, которое содержит восстановленные блоки кодирования, для того чтобы выполнять взаимное прогнозирование для PU других изображений. Помимо этого, модуль 126 обработки внутреннего прогнозирования может использовать восстановленные блоки кодирования в буфере 116 декодированных изображений для того, чтобы выполнять внутреннее прогнозирование для других PU в изображении, идентичном изображению CU.

Модуль 118 энтропийного кодирования может принимать данные из других функциональных компонентов видеокодера 20. Например, модуль 118 энтропийного кодирования может принимать блоки коэффициентов из модуля 106 квантования и может принимать синтаксические элементы из модуля 100 обработки прогнозирования. Модуль 118 энтропийного кодирования может выполнять одну или более операций энтропийного кодирования для данных, чтобы формировать энтропийно кодированные данные. Например, модуль 118 энтропийного кодирования может выполнять CABAC-операцию, операцию контекстно-адаптивного кодирования переменной длины (CAVLC), операцию кодирования переменно-переменной (V2V) длины, операцию синтаксического контекстно-адаптивного двоичного арифметического кодирования (SBAC), операцию энтропийного кодирования на основе сегментирования на интервалы вероятности (PIPE), операцию кодирования экспоненциальным кодом Голомба или другой тип операции энтропийного кодирования для данных. Видеокодер 20 может выводить поток битов, который включает в себя энтропийно кодированные данные, сформированные посредством модуля 118 энтропийного кодирования. Например, поток битов может включать в себя данные, которые представляют RQT для CU.

В некоторых примерах, остаточное кодирование не выполняется при палитровом кодировании. Соответственно видеокодер 20 не может выполнять преобразование или квантование при кодировании с использованием режима палитрового кодирования. Помимо этого, видеокодер 20 может энтропийно кодировать данные, сформированные с использованием режима палитрового кодирования, отдельно от остаточных данных.

Согласно одной или более технологиям этого раскрытия сущности видеокодер 20, и в частности модуль 122 кодирования на основе палитр, может выполнять кодирование видео на основе палитр прогнозных видеоблоков. Как описано выше, палитра, сформированная посредством видеокодера 20, может явно кодироваться, прогнозироваться из предыдущих записей палитры, прогнозироваться из предыдущих пиксельных значений либо как комбинация вышеозначенного.

В частности, модуль 100 обработки прогнозирования может определять режим кодирования для блока (например, CU или PU) видеоданных, к примеру режим взаимного прогнозирования, режим внутреннего прогнозирования или палитровый режим. При условии, что выбирается палитровый режим, модуль 122 кодирования на основе палитр может формировать палитру для блока на основе статистики пиксельных значений для блока. Для каждого пикселя блока модуль 122 кодирования на основе палитр может определять то, имеет ли нет пиксел соответствующее значение в палитре, и если да, передавать в служебных сигналах индекс в палитре в соответствующее значение для пикселя. Модуль 122 кодирования на основе палитр также может передавать в служебных сигналах значение серии, представляющее число пикселей, имеющих значение, идентичное значению предыдущего пикселя.

Альтернативно, если последовательность пикселей имеет значения, равные соседним сверху пикселям, модуль 122 кодирования на основе палитр может передавать в служебных сигналах значение серии для режима "с копированием сверху", причем серия представляет число пикселей, имеющих значения, равные соседним сверху пикселям.

Если ни индексный режим, ни режим с копированием сверху адекватно не представляют значение текущего пикселя кодированного в палитровом режиме блока, модуль 122 кодирования на основе палитр может использовать технологии этого раскрытия сущности, чтобы кодировать пиксел в качестве пикселя с управляющим кодом. В соответствии с технологиями этого раскрытия сущности видеокодер 20 может кодировать значение пикселя с управляющим кодом в двух частях, первой части и второй части. Первая и вторая части при конкатенации могут представлять значение пикселя с управляющим кодом. Модуль 118 энтропийного кодирования может кодировать первую часть с использованием кодового слова Голомба-Райса и вторую часть с использованием кодового слова с экспоненциальным кодом Голомба с параметром 3. В еще одних других примерах видеокодер 20 и/или видеокодер 30 могут кодировать (кодировать или декодировать соответственно) только одно значение с использованием кодирования экспоненциальным кодом Голомба с параметром 3, причем это значение представляет полное значение для пикселя с управляющим кодом. В еще одних других примерах модуль 118 энтропийного кодирования может кодировать только одно значение с использованием кодирования экспоненциальным кодом Голомба с параметром 3, причем это значение представляет полное значение для пикселя с управляющим кодом.

В некоторых примерах модуль 118 энтропийного кодирования может кодировать вторую часть с использованием кодирования усеченным экспоненциальным кодом Голомба. Кодирование усеченным экспоненциальным кодом Голомба может быть основано на определенном максимальном квантованном

пиксельном значении с управляющим кодом. Видеокодер 20 (более конкретно, модуль 118 энтропийного кодирования) может определять максимальное квантованное пиксельное значение с управляющим кодом с использованием любой из различных технологий, описанных в данном документе.

Например, модуль 118 энтропийного кодирования может определять значение размера шага квантования на основе параметра квантования для блока согласно вышеприведенной формуле (2) и затем использовать размер шага квантования для того, чтобы определять максимальное квантованное пиксельное значение с управляющим кодом, например, согласно вышеприведенному процессу (3). В частности, в процессе (3) временное значение инициализируется равным значению квантованного шага, определенному с использованием формулы (2). Затем, в то время как временное значение меньше $(1 \ll \text{bitDepth}) - 1$, т.е. значения в единицу, сдвинутого влево на битовую глубину, минус 1, временное значение увеличивается на определенное значение квантованного шага и максимальное квантованное пиксельное значение с управляющим кодом (инициализированное равным единице) увеличивается на единицу. Альтернативно модуль 118 энтропийного кодирования может определять максимальное квантованное пиксельное значение с управляющим кодом согласно вышеприведенному процессу (4).

В качестве другого примера видеокодер 20 может передавать в служебных сигналах значения для определенных переменных кодера, такие как значение (`quantScale`) масштабирования при квантовании, значение (`rightShiftParam`) параметра сдвига вправо и значение (`rightShiftOffset`) смещения со сдвигом вправо. Типично эти значения должны быть доступными только для видеокодера 20, но в этом примере видеокодер 20 может передавать в служебных сигналах значения для этих элементов в потоке битов и также использовать эти значения для того, чтобы определять максимальное квантованное пиксельное значение с управляющим кодом. Видеокодер 20 может передавать в служебных сигналах значения для этих элементов с использованием одной или более таблиц поиска. Кроме того, модуль 118 энтропийного кодирования может вычислять максимальное квантованное пиксельное значение с управляющим кодом с использованием вышеприведенных процессов (5), (6) и (7). Как можно видеть, процесс (7) включает в себя вычисление максимального квантованного пиксельного значения `sMax` с управляющим кодом с использованием `quantScale`, `rightShiftOffset` и `rightShiftParam`. Альтернативно модуль 118 энтропийного кодирования может вычислять максимальное квантованное пиксельное значение с управляющим кодом с использованием процесса (8), который использует элементы `quantScale`, `rightShiftOffset` и `rightShiftParam` и опускает операции отсечения и деления.

В качестве еще одного другого примера модуль 118 энтропийного кодирования может определять максимальное квантованное пиксельное значение с управляющим кодом с использованием вышеприведенного процесса (9), который использует таблицу поиска на основе параметра (`qP`) квантования и битовой глубины. Таблица поиска может задаваться согласно либо табл. 3 или 4 либо другим таблицам поиска.

В качестве другого примера модуль 118 энтропийного кодирования может определять максимальный квантованный пиксел с управляющим кодом с использованием вышеприведенных процессов (10)-(13). Иными словами, модуль 118 энтропийного кодирования может определять значение `qPBase` (с использованием процесса (10)), которое может быть предварительно определенным значением. Модуль 118 энтропийного кодирования затем может определять значение `quantBits` из параметра (`qP`) квантования и значения `qPBase` с использованием процесса (11), затем определять значение `maxBit` согласно процессу (12) из битовой глубины и определенного значения `quantBits`. В завершение, модуль 118 энтропийного кодирования может определять максимальное квантованное пиксельное значение (`sMax`) с управляющим кодом согласно процессу (13) в качестве значения в 1, сдвинутого влево на значение `maxBit`, минус 1.

Как отмечено выше, любая из этих различных примерных технологий может использоваться для того, чтобы определять максимальное квантованное пиксельное значение с управляющим кодом, которое модуль 118 энтропийного кодирования может использовать для того, чтобы выполнять кодирование усеченным экспоненциальным кодом Голомба второй части пиксельного значения с управляющим кодом. Альтернативно модуль 118 энтропийного кодирования может использовать любую из этих различных технологий для того, чтобы выполнять усеченное двоичное кодирование пиксельного значения с управляющим кодом без выполнения комбинации кодирования кодом Голомба-Райса и кодирования экспоненциальным кодом Голомба с параметром 3.

Таким образом, видеокодер 20 представляет пример видеокодера, выполненного с возможностью определять значение для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных и кодировать экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пиксела с управляющим кодом, при этом для того, чтобы кодировать экспоненциальное кодовое слово Голомба, видеокодер выполнен с возможностью использовать кодирование экспоненциальным кодом Голомба с параметром 3.

Фиг. 3 является блок-схемой, иллюстрирующей примерный видеодекодер 30, который выполнен с возможностью реализовывать технологии этого раскрытия сущности. Фиг. 3 предоставляется для целей пояснения и не является ограничением технологий, как проиллюстрировано и описано в общих чертах в этом раскрытии сущности. Для целей пояснения это раскрытие сущности описывает видеодекодер 30 в контексте HEVC-кодирования. Тем не менее, технологии этого раскрытия сущности могут быть применимыми к другим стандартам или способам кодирования.

В примере по фиг. 3 видеодекoder 30 включает в себя запоминающее устройство 148 видеоданных, модуль 150 энтропийного декодирования, модуль 152 обработки прогнозирования, модуль 154 обратного квантования, модуль 156 обработки обратного преобразования, модуль 158 восстановления, модуль 160 фильтрации и буфер 162 декодированных изображений. Модуль 152 обработки прогнозирования включает в себя модуль 164 компенсации движения и модуль 166 обработки внутреннего прогнозирования. Видеодекoder 30 также включает в себя модуль 165 декодирования на основе палитр, выполненный с возможностью осуществлять различные аспекты технологий кодирования на основе палитр, описанных в этом раскрытии сущности. В других примерах видеодекoder 30 может включать в себя большее, меньшее число или другие функциональные компоненты.

Запоминающее устройство 148 видеоданных может сохранять видеоданные, такие как кодированный поток видеобитов, которые должны декодироваться посредством компонентов видеодекодера 30. Видеоданные, сохраненные в запоминающем устройстве 148 видеоданных, могут получаться, например, из машиночитаемого носителя 16, например, из локального видеоисточника, такого как камера, через передачу по проводной или беспроводной сети видеоданных, либо посредством осуществления доступа к физическим носителям хранения данных. Запоминающее устройство 148 видеоданных может формировать буфер кодированных изображений (CPB), который сохраняет кодированные видеоданные из кодированного потока видеобитов. Буфер 162 декодированных изображений может представлять собой запоминающее устройство опорных изображений, которое сохраняет опорные видеоданные для использования при декодировании видеоданных посредством видеодекодера 30, например в режимах внутреннего или взаимного кодирования. Запоминающее устройство 148 видеоданных и буфер 162 декодированных изображений могут формироваться посредством любого из множества запоминающих устройств, к примеру как динамическое оперативное запоминающее устройство (DRAM), включающее в себя синхронное DRAM (SDRAM), магниторезистивное RAM (MRAM), резистивное RAM (RRAM) или другие типы запоминающих устройств. Запоминающее устройство 148 видеоданных и буфер 162 декодированных изображений могут предоставляться посредством идентичного запоминающего устройства или отдельных запоминающих устройств. В различных примерах запоминающее устройство 148 видеоданных может быть внутрикристалльным с другими компонентами видеодекодера 30 или внекристалльным относительно этих компонентов.

Запоминающее устройство 148 видеоданных, т.е. CPB, может принимать и сохранять кодированные видеоданные (например, NAL-единицы) потока битов. Модуль 150 энтропийного декодирования может принимать кодированные видеоданные (например, NAL-единицы) из запоминающего устройства 148 видеоданных и может синтаксически анализировать NAL-единицы, чтобы декодировать синтаксические элементы. Модуль 150 энтропийного декодирования может энтропийно декодировать энтропийно кодированные синтаксические элементы в NAL-единицах. Модуль 152 обработки прогнозирования, модуль 154 обратного квантования, модуль 156 обработки обратного преобразования, модуль 158 восстановления и модуль 160 фильтрации могут формировать декодированные видеоданные на основе синтаксических элементов, полученных (к примеру, извлеченных) из потока битов.

NAL-единицы потока битов могут включать в себя NAL-единицы кодированных серий последовательных макроблоков. В качестве части декодирования потока битов модуль 150 энтропийного декодирования может извлекать и энтропийно декодировать синтаксические элементы из NAL-единиц кодированных серий последовательных макроблоков. Каждая из кодированных серий последовательных макроблоков может включать в себя заголовок серии последовательных макроблоков и данные серии последовательных макроблоков. Заголовок серии последовательных макроблоков может содержать синтаксические элементы, связанные с серией последовательных макроблоков. Синтаксические элементы в заголовке серии последовательных макроблоков могут включать в себя синтаксический элемент, который идентифицирует PPS, ассоциированный с изображением, которое содержит серию последовательных макроблоков.

Помимо декодирования синтаксических элементов из потока битов, видеодекoder 30 может выполнять операцию восстановления для несегментированной CU. Чтобы выполнять операцию восстановления для несегментированной CU, видеодекoder 30 может выполнять операцию восстановления для каждой TU CU. Посредством выполнения операции восстановления для каждой TU CU видеодекoder 30 может восстанавливать остаточные блоки, ассоциированные с CU.

В качестве части выполнения операции восстановления для TU CU модуль 154 обратного квантования может обратно квантовать, т.е. деквантовать, блоки коэффициентов, ассоциированные с TU. Модуль 154 обратного квантования может использовать QR-значение, ассоциированное с CU TU, для того, чтобы определять степень квантования и аналогично степень обратного квантования для модуля 154 обратного квантования для применения. Иными словами, коэффициент сжатия, т.е. соотношение числа битов, используемых для того, чтобы представлять исходную последовательность и сжатую последовательность, может управляться посредством регулирования значения QR, используемого при квантовании коэффициентов преобразования. Коэффициент сжатия также может зависеть от используемого способа энтропийного кодирования.

После того как модуль 154 обратного квантования обратно квантует блок коэффициентов, модуль

156 обработки обратного преобразования может применять одно или более обратных преобразований к блоку коэффициентов, чтобы формировать остаточный блок, ассоциированный с TU. Например, модуль 156 обработки обратного преобразования может применять обратное DCT, обратное целочисленное преобразование, обратное преобразование Карунена-Лоэва (KLT), обратное вращательное преобразование, обратное направленное преобразование или другое обратное преобразование к блоку коэффициентов.

Если PU кодируется с использованием внутреннего прогнозирования, модуль 166 обработки внутреннего прогнозирования может выполнять внутреннее прогнозирование для того, чтобы формировать прогнозирующие блоки для PU. Модуль 166 обработки внутреннего прогнозирования может использовать режим внутреннего прогнозирования для того, чтобы формировать прогнозирующие блоки сигналов яркости, прогнозирующие Cb-блоки и прогнозирующие Cr-блоки для PU на основе прогнозных блоков пространственно соседних PU. Модуль 166 обработки внутреннего прогнозирования может определять режим внутреннего прогнозирования для PU на основе одного или более синтаксических элементов, декодированных из потока битов.

Модуль 152 обработки прогнозирования может составлять первый список опорных изображений (RefPicList0) и второй список опорных изображений (RefPicList1) на основе синтаксических элементов, извлеченных из потока битов. Кроме того, если PU кодируется с использованием взаимного прогнозирования, модуль 150 энтропийного декодирования может извлекать информацию движения для PU. Модуль 164 компенсации движения может определять на основе информации движения PU одну или более опорных областей для PU. Модуль 164 компенсации движения может формировать на основе блоков выборок в одном или более опорных блоков для PU прогнозирующие блоки (прогнозирующие блоки сигналов яркости, прогнозирующие Cb-блоки и прогнозирующие Cr-блоки) для PU.

Модуль 158 восстановления может использовать блоки преобразования (например, блоки преобразования сигналов яркости, Cb-блоки преобразования и Cr-блоки преобразования), ассоциированные с TU CU, и прогнозирующие блоки (например, блоки сигналов яркости, Cb-блоки и Cr-блоки) PU CU, т.е. либо данные внутреннего прогнозирования, либо данные взаимного прогнозирования, при соответствующих условиях для того, чтобы восстанавливать блоки кодирования (например, блоки кодирования сигналов яркости, Cb-блоки кодирования и Cr-блоки кодирования) CU. Например, модуль 158 восстановления может суммировать выборки блоков преобразования (например, блоков преобразования сигналов яркости, Cb-блоков преобразования и Cr-блоков преобразования) с соответствующими выборками прогнозирующих блоков (например, прогнозирующих блоков сигналов яркости, прогнозирующих Cb-блоков и прогнозирующих Cr-блоков), чтобы восстанавливать блоки кодирования (например, блоки кодирования сигналов яркости, Cb-блоки кодирования и Cr-блоки кодирования) CU.

Модуль 160 фильтрации может выполнять операцию удаления блочности, чтобы уменьшать артефакты блочности, ассоциированные с блоками кодирования (к примеру, с блоками кодирования сигналов яркости, Cb-блоками кодирования и Cr-блоками кодирования) CU. Видеодекодер 30 может сохранять блоки кодирования (к примеру, блоки кодирования сигналов яркости, Cb-блоки кодирования и Cr-блоки кодирования) CU в буфере 162 декодированных изображений. Буфер 162 декодированных изображений может предоставлять опорные изображения для последующей компенсации движения, внутреннего прогнозирования и представления на устройстве отображения, к примеру на устройстве 32 отображения по фиг. 1. Например, видеодекодер 30 может выполнять на основе блоков (к примеру, блоков сигналов яркости, Cb-блоков и Cr-блоков) в буфере 162 декодированных изображений операции внутреннего прогнозирования или взаимного прогнозирования для PU других CU. Таким образом, видеодекодер 30 может извлекать из потока битов уровни коэффициентов преобразования блока значимых коэффициентов, обратно квантовать уровни коэффициентов преобразования, применять преобразование к уровням коэффициентов преобразования для того, чтобы формировать блок преобразования, формировать, по меньшей мере, частично на основе блока преобразования блок кодирования и выводить блок кодирования для отображения.

В соответствии с различными примерами этого раскрытия сущности видеодекодер 30 может быть выполнен с возможностью осуществлять кодирование на основе палитр. В частности, видеодекодер 30 включает в себя модуль 165 декодирования на основе палитр, который может выполнять кодирование на основе палитр. Например, модуль 165 декодирования на основе палитр может быть выполнен с возможностью формировать палитру, имеющую записи, указывающие пиксельные значения. Кроме того, в этом примере модуль 165 декодирования на основе палитр может принимать информацию, ассоциирующую, по меньшей мере, некоторые позиции блока видеоданных с записями в палитре. В этом примере модуль 165 декодирования на основе палитр может выбирать пиксельные значения в палитре на основе информации. Дополнительно в этом примере модуль 165 декодирования на основе палитр может восстанавливать пиксельные значения блока на основе выбранных пиксельных значений. Хотя различные функции описываются как выполняемые посредством модуля 165 декодирования на основе палитр, некоторые или все такие функции могут выполняться посредством других модулей обработки или комбинации различных модулей обработки.

Модуль 165 декодирования на основе палитр может принимать информацию режима палитрового кодирования и выполнять вышеуказанные операции, когда информация режима палитрового кодирова-

ния указывает то, что режим палитрового кодирования применяется к блоку. Когда информация режима палитрового кодирования указывает то, что режим палитрового кодирования не применяется к блоку, либо когда информация других режимов указывает использование другого режима, модуль 165 декодирования на основе палитр декодирует блок видеоданных с использованием режима кодирования не на основе палитр, например, такого режима взаимного прогнозирующего или внутреннего прогнозирующего HEVC-кодирования, когда информация режима палитрового кодирования указывает то, что режим палитрового кодирования не применяется к блоку. Блок видеоданных, например, может представлять собой CU или PU, сформированную согласно процессу HEVC-кодирования. Видеодекодер 30 может декодировать некоторые блоки с помощью режимов взаимного прогнозирующего временного прогнозирования или внутреннего прогнозирующего пространственного кодирования и декодировать другие блоки с помощью режима кодирования на основе палитр. Режим кодирования на основе палитр может содержать один из множества различных режимов кодирования на основе палитр, либо может быть предусмотрен один режим кодирования на основе палитр.

Согласно одной или более технологий этого раскрытия сущности видеодекодер 30, и в частности модуль 165 декодирования на основе палитр, может выполнять декодирование видео на основе палитр кодированных в палитровом режиме видеоблоков. Как описано выше, палитра, декодированная посредством видеодекодера 30, может явно кодироваться и передаваться в служебных сигналах, восстанавливаться посредством видеодекодера 30 относительно принимаемого кодированного в палитровом режиме блока, прогнозироваться из предыдущих записей палитры, прогнозироваться прогнозированных из предыдущих пиксельных значений либо как комбинация вышеозначенного.

В частности, модуль 150 энтропийного декодирования может декодировать информацию, указывающую то, что блок (например, PU или CU) видеоданных кодируется с использованием палитрового режима. Модуль 150 энтропийного декодирования дополнительно может декодировать информацию, представляющую то, как каждый пиксел блока кодируется с использованием палитрового режима. Например, модуль 150 энтропийного декодирования может декодировать значение индекса и значение серии в качестве пары {индекс, серия}. Значение индекса представляет запись палитры для блока, причем запись указывает пиксельное значение, и значение серии указывает число пикселей в дополнение к текущему пикселу, которые имеют идентичное значение.

Если значение индекса равно размеру палитры, модуль 150 энтропийного декодирования может определять то, что текущий пиксел представляет собой пиксел с управляющим кодом. Таким образом, в соответствии с технологиями этого раскрытия сущности модуль 150 энтропийного декодирования может декодировать два последующих значения потока битов: кодированное кодом Голомба-Райса первое значение и кодированное экспоненциальным кодом Голомба с параметром 3 второе значение. Первое значение представляет первую часть значения пиксела с управляющим кодом, в то время как второе значение представляет вторую часть значения пиксела с управляющим кодом. Таким образом, после декодирования первого значения и второго значения (формирующих первую часть и вторую часть соответственно) модуль 150 энтропийного декодирования может конкатенировать первую часть и вторую часть для того, чтобы воспроизводить значение пиксела с управляющим кодом и предоставлять значение пиксела с управляющим кодом в модуль 165 декодирования на основе палитр. Альтернативно модуль 150 энтропийного декодирования может предоставлять первую часть и вторую часть в модуль 165 декодирования на основе палитр, который может конкатенировать первую часть и вторую часть для того, чтобы воспроизводить значение пиксела с управляющим кодом. В еще одних других примерах модуль 150 энтропийного декодирования может декодировать только одно значение с использованием кодирования экспоненциальным кодом Голомба с параметром 3, причем это значение представляет полное значение для пиксела с управляющим кодом.

В некоторых примерах модуль 150 энтропийного декодирования может декодировать вторую часть с использованием кодирования усеченным экспоненциальным кодом Голомба. Кодирование усеченным экспоненциальным кодом Голомба может быть основано на определенном максимальном квантованном пиксельном значении с управляющим кодом. Видеодекодер 30 (более конкретно, модуль 150 энтропийного декодирования) может определять максимальное квантованное пиксельное значение с управляющим кодом с использованием любой из различных технологий, описанных в данном документе.

Например, модуль 150 энтропийного декодирования может определять значение размера шага квантования на основе параметра квантования для блока согласно вышеприведенной формуле (2) и затем использовать размер шага квантования для того, чтобы определять максимальное квантованное пиксельное значение с управляющим кодом, например, согласно вышеприведенному процессу (3). В частности, в процессе (3) временное значение инициализируется равным значению квантованного шага, определенному с использованием формулы (2). Затем, в то время как временное значение меньше $(1 \ll \text{bitDepth}) - 1$, т.е. значения в единицу, сдвинутого влево на битовую глубину, минус 1, временное значение увеличивается на определенное значение квантованного шага, и максимальное квантованное пиксельное значение с управляющим кодом (инициализированное равным единице) увеличивается на единицу. Альтернативно модуль 150 энтропийного декодирования может определять максимальное квантованное пиксельное значение с управляющим кодом согласно вышеприведенному процессу (4).

В качестве другого примера видеодекoder 30 может принимать значения для определенных переменных кодера, такие как значение (quantScale) масштабирования при квантовании, значение (rightShiftParam) параметра сдвига вправо и значение (rightShiftOffset) смещения со сдвигом вправо. Типично эти значения должны быть доступными только для видеокодера, но в этом примере видеодекoder 30 может декодировать значения для этих элементов из потока битов и также использовать эти значения для того, чтобы определять максимальное квантованное пиксельное значение с управляющим кодом. Видеодекoder 30 может декодировать индексы в одну или более таблиц поиска, которые представляют эти переменные кодеры. Кроме того, модуль 150 энтропийного декодирования может вычислять максимальное квантованное пиксельное значение с управляющим кодом с использованием вышеприведенных процессов (5), (6) и (7). Как можно видеть, процесс (7) включает в себя вычисление максимального квантованного пиксельного значения сMax с управляющим кодом с использованием quantScale, rightShiftOffset и rightShiftParam. Альтернативно модуль 150 энтропийного декодирования может вычислять максимальное квантованное пиксельное значение с управляющим кодом с использованием процесса (8), который использует элементы quantScale, rightShiftOffset и rightShiftParam и опускает операции отсечения и деления.

В качестве еще одного другого примера модуль 150 энтропийного декодирования может определять максимальное квантованное пиксельное значение с управляющим кодом с использованием вышеприведенного процесса (9), который использует таблицу поиска на основе параметра (qP) квантования и битовой глубины. Таблица поиска может задаваться согласно либо табл. 3 или 4, либо другим таблицам поиска.

В качестве другого примера модуль 150 энтропийного декодирования может определять максимальный квантованный пиксел с управляющим кодом с использованием вышеприведенных процессов (10)-(13). Иными словами, модуль 150 энтропийного декодирования может определять значение qPBase (с использованием процесса (10)), которое может быть предварительно определенным значением. Модуль 150 энтропийного декодирования затем может определять значение quantBits из параметра (qP) квантования и значения qPBase с использованием процесса (11), затем определять значение maxBit согласно процессу (12) из битовой глубины и определенного значения quantBits. В завершение, модуль 150 энтропийного декодирования может определять максимальное квантованное пиксельное значение (сMax) с управляющим кодом согласно процессу (13) в качестве значения в 1, сдвинутого влево на значение maxBit, минус 1.

Как отмечено выше, любая из этих различных примерных технологий может использоваться для того, чтобы определять максимальное квантованное пиксельное значение с управляющим кодом, которое модуль 150 энтропийного декодирования может использовать для того, чтобы выполнять кодирование усеченным экспоненциальным кодом Голомба второй части пиксельного значения с управляющим кодом. Альтернативно модуль 150 энтропийного декодирования может использовать любую из этих различных технологий для того, чтобы выполнять усеченное двоичное кодирование пиксельного значения с управляющим кодом без выполнения комбинации кодирования кодом Голомба-Райса и кодирования экспоненциальным кодом Голомба с параметром 3.

С другой стороны, модуль 150 энтропийного декодирования может выполнять вышеприведенные процессы для того, чтобы декодировать пиксельное значение с управляющим кодом, передаваемое в служебных сигналах посредством значения индекса палитры, которое равно размеру палитры. Альтернативно модуль 150 энтропийного декодирования может декодировать только значение серии. Значение серии обозначает число пикселей, кодированных с использованием режима с копированием сверху. Модуль 150 энтропийного декодирования может предоставлять декодированное значение серии в модуль 165 декодирования на основе палитр. Таким образом, модуль 165 декодирования на основе палитр может извлекать значения для каждого из этих пикселей из соответствующих соседних сверху пикселей.

Таким образом, видеодекoder 30 представляет пример видеодекодера, выполненного с возможностью декодировать экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пиксела с управляющим кодом кодированного в палитровом режиме блока, при этом видеодекoder выполнен с возможностью декодировать экспоненциальное кодовое слово Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3 и декодировать блок с использованием значения для пиксела с управляющим кодом.

Фиг. 4 является блок-схемой последовательности операций, иллюстрирующей примерный способ для кодирования блока видеоданных в соответствии с технологиями этого раскрытия сущности. Способ по фиг. 4 может осуществляться посредством видеокодера 20 и его компонентов (например, проиллюстрированных на фиг. 2).

В этом примере модуль 100 обработки прогнозирования первоначально принимает блок видеоданных (200). Блок, например, может представлять собой единицу прогнозирования (PU) или единицу кодирования (CU). Модуль 100 обработки прогнозирования затем определяет режим кодирования для блока (202). Например, модуль 100 обработки прогнозирования может тестировать различные режимы кодирования и сравнивать режимы с использованием процесса оптимизации искажения в зависимости от скорости передачи (RDO). Кроме того, модуль 100 обработки прогнозирования также может сравнивать различные размеры блоков и схемы сегментации на блоки с использованием RDO-процесса.

Модуль 100 обработки прогнозирования может выбирать режим внутреннего или взаимного прогнозирования, причем в этом случае модуль 120 обработки взаимного прогнозирования или модуль 126 обработки внутреннего прогнозирования может прогнозировать пиксели блока с использованием внутреннего прогнозирования или взаимного прогнозирования (204) соответственно, формируя прогнозированный блок. Видеокодер 20 затем может формировать и обрабатывать остаточные значения блока (206). Например, модуль 102 формирования остатков может вычитать исходный блок из прогнозированного блока попиксельно, формируя остаточный блок. Модуль 104 обработки преобразования затем может преобразовывать остаточный блок с использованием преобразования, такого как, например, DCT, формируя блок преобразования. Модуль 106 квантования затем может квантовать коэффициенты преобразования блока преобразования и предоставлять квантованные коэффициенты преобразования в модуль 118 энтропийного кодирования. Модуль 100 обработки прогнозирования также предоставляет информацию, представляющую режим прогнозирования (например, внутренний/взаимный, выбранный внутренний режим, если используется внутреннее прогнозирование, или параметры движения, если используется взаимное прогнозирование), в модуль 118 энтропийного кодирования. Таким образом, модуль 118 энтропийного кодирования энтропийно кодирует информацию прогнозирования и остаточные значения (т.е. квантованные коэффициенты преобразования) (208).

Альтернативно модуль 100 обработки прогнозирования может выбирать палитровый режим для того, чтобы кодировать блок, причем в этом случае модуль 122 кодирования на основе палитр анализирует пиксельную статистику для блока (210). Например, модуль 122 кодирования на основе палитр может определять часто используемые пиксельные значения. Модуль 122 кодирования на основе палитр затем формирует палитру для блока на основе статистики (212). Хотя не показано на фиг. 4, модуль 118 энтропийного кодирования может энтропийно кодировать данные для палитры. Например, палитра может прогнозирующе кодироваться относительно ранее используемой палитры, например, как пояснено выше относительно табл. 1 и 2.

Модуль 122 кодирования на основе палитр затем может сканировать пиксели блока (214), чтобы определять то, как кодировать пиксели. Например, модуль 122 кодирования на основе палитр может определять то, включено или нет текущее пиксельное значение в палитру. Если пиксельное значение включено в палитру, модуль 122 кодирования на основе палитр может предоставлять индекс из палитры, которая соответствует пиксельному значению, в модуль 118 энтропийного кодирования, который может энтропийно кодировать значение индекса (216). Кроме того, модуль 122 кодирования на основе палитр может определять число пикселей в строке после предыдущего пикселя, которые имеют идентичное значение, и предоставлять значение "серии" в модуль 118 энтропийного кодирования, который может энтропийно кодировать значение серии (218).

Альтернативно, если текущий пиксел не имеет значения в палитре, модуль 122 кодирования на основе палитр может определять то, является или нет значение для пикселя идентичным значению соседнего сверху пикселя. В некоторых примерах это определение может выполняться до определения того, соответствует или нет пиксельное значение значению в палитре. В любом случае, если текущий пиксел имеет значение, которое равно значению соседнего сверху пикселя, модуль 122 кодирования на основе палитр может предоставлять значение серии, описывающее число пикселей, которые имеют значения, равные их соседним сверху пикселям, в модуль 118 энтропийного кодирования, который может энтропийно кодировать значение серии (220).

Если текущий пиксел не соответствует значению в палитре и не имеет значения, равного соседнему сверху пикселу, модуль 122 кодирования на основе палитр может кодировать пиксел в качестве пикселя с управляющим кодом. В частности, модуль 122 кодирования на основе палитр может предоставлять значение индекса, равное размеру палитры, в модуль 118 энтропийного кодирования, который может энтропийно кодировать значение индекса в качестве размера палитры (222). Значение индекса, равное размеру палитры, может передавать в служебных сигналах то, что текущий пиксел кодирован как пиксел с управляющим кодом. Модуль 122 кодирования на основе палитр дополнительно может предоставлять значение пикселя с управляющим кодом в модуль 118 энтропийного кодирования.

В соответствии с технологиями этого раскрытия сущности модуль 118 энтропийного кодирования может кодировать значение пикселя с управляющим кодом в двух частях. Более конкретно, модуль 118 энтропийного кодирования может разбивать значение пикселя с управляющим кодом на первую часть и вторую часть. Модуль 118 энтропийного кодирования может кодировать кодом Голомба-Райса первую часть пиксельного значения (224) и кодировать экспоненциальным кодом Голомба с параметром 3 вторую часть пиксельного значения (226). Хотя не показано на фиг. 4, модуль 118 энтропийного кодирования может кодировать вторую часть с использованием усеченного экспоненциального кода Голомба с параметром 3 на основе определенного максимального квантованного пиксельного значения с управляющим кодом.

Видеокодер 20 может выполнять этот процесс (например, одно из последовательности этапов 216 и 218, этапа 220 или последовательности этапов 222-226) для каждого пикселя кодированного в палитровом режиме блока.

Таким образом, способ по фиг. 4 представляет пример способа, включающего в себя определение

значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных и кодирование экспоненциального кодового слова Голомба, представляющего по меньшей мере часть значения для пиксела с управляющим кодом, при этом кодирование экспоненциального кодового слова Голомба содержит кодирование экспоненциального кодового слова Голомба с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

Хотя способ по фиг. 4 включает в себя кодирование как первой части значения пиксела с управляющим кодом с использованием кодирования кодом Голомба-Райса, так и второй части значения пиксела с управляющим кодом с использованием кодирования экспоненциальным кодом Голомба, следует понимать, что в других примерах полное значение для пиксела с управляющим кодом может кодироваться с использованием только кодирования экспоненциальным кодом Голомба (например, с использованием параметра 3 для кодирования экспоненциальным кодом Голомба).

Фиг. 5 является блок-схемой последовательности операций, иллюстрирующей другой примерный способ для кодирования блока видеоданных в соответствии с технологиями этого раскрытия сущности. Способ по фиг. 5 может осуществляться посредством видеокодера 20 и его компонентов (например, проиллюстрированных на фиг. 2). В общем, способ по фиг. 5 фактически соответствует способу по фиг. 4. Тем не менее, на фиг. 5, когда текущий пиксел не соответствует значению в палитре и не имеет значения, равного соседнему сверху пикселу, модуль 122 кодирования на основе палитр может кодировать пиксел в качестве пиксела с управляющим кодом, и в частности кодировать значение для пиксела с управляющим кодом с использованием кодирования экспоненциальным кодом Голомба с параметром 3 (228).

Таким образом, способ по фиг. 5 представляет другой пример способа, включающего в себя определение значения для пиксела с управляющим кодом кодированного в палитровом режиме блока видеоданных и кодирование экспоненциального кодового слова Голомба, представляющего по меньшей мере часть значения для пиксела с управляющим кодом, при этом кодирование экспоненциального кодового слова Голомба содержит кодирование экспоненциального кодового слова Голомба с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

Фиг. 6 является блок-схемой последовательности операций, иллюстрирующей примерный способ для декодирования блока видеоданных в соответствии с технологиями этого раскрытия сущности. Способ по фиг. 6 может осуществляться посредством видеодекодера 30 и его компонентов (например, проиллюстрированных на фиг. 3).

Первоначально модуль 150 энтропийного декодирования может декодировать данные, указывающие режим кодирования для блока видеоданных (250). Модуль 150 энтропийного декодирования может использовать эти данные, чтобы определять режим кодирования для блока (252), например один из режима внутреннего прогнозирования, режима взаимного прогнозирования или палитрового режима.

В случае если режим кодирования представляет собой внутреннее прогнозирование или взаимное прогнозирование, модуль 150 энтропийного декодирования может декодировать информацию прогнозирования (например, внутренний режим или параметры движения) и предоставлять информацию прогнозирования в модуль 152 обработки прогнозирования. Модуль 152 обработки прогнозирования может использовать информацию прогнозирования для того, чтобы прогнозировать пикселы блока с использованием внутреннего прогнозирования или взаимного прогнозирования (254). Например, модуль 166 обработки внутреннего прогнозирования может использовать режим внутреннего прогнозирования для того, чтобы составлять прогнозный блок из соседних пикселов относительно блока. В качестве другого примера модуль 164 компенсации движения может использовать параметры движения для того, чтобы извлекать (и потенциально обрабатывать, например, фильтровать) опорный блок из ранее декодированного изображения буфера 162 декодированных изображений.

Помимо этого, видеодекодер 30 может декодировать и обрабатывать остаточные значения блока (256). Например, модуль 154 обратного квантования может обратно квантовать квантованные коэффициенты преобразования, и модуль 156 обработки обратного преобразования может обратно преобразовывать коэффициенты преобразования, чтобы восстанавливать остаточный блок. Модуль 158 восстановления затем может комбинировать остаточные значения остаточного блока и прогнозируемые значения прогнозируемого блока (258), чтобы восстанавливать исходный блок.

Альтернативно модуль 150 энтропийного декодирования может определять то, что режим кодирования для блока представляет собой палитровый режим. В этом случае модуль 150 энтропийного декодирования может декодировать палитру для блока (260). Как пояснено выше относительно табл. 1 и 2, палитра может прогнозирующе кодироваться относительно предыдущей палитры. Таким образом, модуль 150 энтропийного декодирования может предоставлять декодированные данные для палитры в модуль 165 декодирования на основе палитр, который может восстанавливать палитру для блока с использованием декодированных данных.

Модуль 150 энтропийного декодирования также может декодировать данные для пикселов блока (264). Например, декодированные данные могут соответствовать значению индекса, которое меньше размера палитры. В этом случае модуль 150 энтропийного декодирования также может декодировать значение серии (266) и предоставлять индекс и значение серии в модуль 165 декодирования на основе палитр. Модуль 165 декодирования на основе палитр может задавать значение пиксела и каждого из пик-

слов в серии равным пиксельному значению палитры, которая соответствует значению индекса (268).

В качестве другого примера, декодированные данные могут быть значением серии без значения индекса. Такое значение серии без значения индекса может указывать число пикселей, кодированных с использованием режима с копированием сверху. В этом случае модуль 150 энтропийного декодирования может предоставлять значение серии в модуль 165 декодирования на основе палитр, который может задавать значения для каждого из пикселей в серии равными значениям соответствующих значений соседнего сверху пикселя.

В качестве другого примера декодированные данные могут быть значением индекса, которое равно размеру палитры. В этом случае модуль 150 энтропийного декодирования может декодировать кодом Голомба-Райса первое значение, соответствующее первой части пиксельного значения (272). Модуль 150 энтропийного декодирования также может декодировать экспоненциальным кодом Голомба с параметром 3 второе значение, соответствующее второй части пиксельного значения (274). Как пояснено выше, модуль 150 энтропийного декодирования альтернативно может определять максимальное квантованное пиксельное значение с управляющим кодом, чтобы выполнять декодирование усеченным экспоненциальным кодом Голомба с параметром 3 второго значения. Модуль 150 энтропийного декодирования (или модуль 165 декодирования на основе палитр) затем может конкатенировать первую часть и вторую часть для того, чтобы воспроизводить пиксельное значение (276).

Видеодекодер 30 может выполнять этот процесс (например, одно из последовательности этапов 266 и 268, этапа 270 или последовательности этапов 272-276) для каждого пикселя кодированного в палитровом режиме блока.

Таким образом, способ по фиг. 6 представляет пример способа, включающего в себя декодирование экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пикселя с управляющим кодом кодированного в палитровом режиме блока видеоданных, при этом декодирование экспоненциального кодового слова Голомба содержит декодирование экспоненциального кодового слова Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3 и декодирование блока с использованием значения для пикселя с управляющим кодом.

Хотя способ по фиг. 6 включает в себя декодирование как первой части значения пикселя с управляющим кодом с использованием кодирования кодом Голомба-Райса, так и второй части значения пикселя с управляющим кодом с использованием кодирования экспоненциальным кодом Голомба, следует понимать, что в других примерах полное значение для пикселя с управляющим кодом может кодироваться с использованием только кодирования экспоненциальным кодом Голомба (например, с использованием параметра 3 для кодирования экспоненциальным кодом Голомба).

Фиг. 7 является блок-схемой последовательности операций, иллюстрирующей другой примерный способ для декодирования блока видеоданных в соответствии с технологиями этого раскрытия сущности. Способ по фиг. 7 может осуществляться посредством видеодекодера 30 и его компонентов (например, проиллюстрированных на фиг. 3). В общем, способ по фиг. 7 фактически соответствует способу по фиг. 6. Тем не менее, на фиг. 7, когда значение индекса равно размеру палитры, модуль 165 декодирования на основе палитр может декодировать пиксел в качестве пикселя с управляющим кодом, и в частности декодировать значение для пикселя с управляющим кодом с использованием кодирования экспоненциальным кодом Голомба с параметром 3 (278).

Таким образом, способ по фиг. 7 представляет пример способа, включающего в себя декодирование экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пикселя с управляющим кодом кодированного в палитровом режиме блока видеоданных, при этом декодирование экспоненциального кодового слова Голомба содержит декодирование экспоненциального кодового слова Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3 и декодирование блока с использованием значения для пикселя с управляющим кодом.

Следует признавать, что в зависимости от примера определенные этапы или события любой из технологий, описанных в данном документе, могут выполняться в другой последовательности, могут добавляться, объединяться или вообще исключаться (например, не все описанные этапы или события требуются для практической реализации технологий). Кроме того, в определенных примерах этапы или события могут выполняться одновременно, например посредством многопоточной обработки, обработки прерывания или посредством нескольких процессоров, а не последовательно. Помимо этого, хотя конкретные аспекты этого раскрытия сущности описываются как выполняемые посредством одного модуля или блока для понятности, следует понимать, что технологии этого раскрытия сущности могут выполняться посредством комбинации блоков или модулей, ассоциированных с видеокодером.

Конкретные аспекты этого раскрытия сущности описаны относительно разрабатываемого HEVC-стандарта для целей иллюстрации. Тем не менее, технологии, описанные в этом раскрытии сущности, могут быть полезными для других процессов кодирования видео, включающих в себя другие стандартные или собственные процессы кодирования видео, еще не разработанные.

Технологии, описанные выше, могут выполняться посредством видеокодера 20 (фиг. 1 и 2) и/или видеодекодера 30 (фиг. 1 и 3), оба из которых в общем могут упоминаться в качестве видеокодера. Аналогично кодирование видео может означать кодирование видео или декодирование видео при соответст-

вующих условиях.

Хотя выше описываются конкретные комбинации различных аспектов технологий, эти комбинации предоставляются просто для того, чтобы иллюстрировать примеры технологий, описанных в этом раскрытии сущности. Соответственно технологии этого раскрытия сущности не должны быть ограничены этими примерными комбинациями и могут охватывать любую возможную комбинацию различных аспектов технологий, описанных в этом раскрытии сущности.

В одном или более примеров описанные функции могут быть реализованы в аппаратных средствах, программном обеспечении, микропрограммном обеспечении или любой комбинации вышеозначенного. При реализации в программном обеспечении функции могут сохраняться или передаваться в качестве одной или более инструкций или кода по машиночитаемому носителю и выполняться посредством аппаратного модуля обработки.

Машиночитаемые носители могут включать в себя машиночитаемые носители хранения данных, которые соответствуют материальному носителю, такие как носители хранения данных, или среды связи, включающие в себя любой носитель, который упрощает перенос компьютерной программы из одного места в другое, например, согласно протоколу связи. Таким образом, машиночитаемые носители в общем могут соответствовать (1) материальному машиночитаемому носителю хранения данных, который является энергонезависимым, или (2) среде связи, такой как сигнал или несущая. Носители хранения данных могут представлять собой любые доступные носители, к которым может осуществляться доступ посредством одного или более компьютеров или одного или более процессоров, с тем чтобы извлекать инструкции, код и/или структуры данных для реализации технологий, описанных в этом раскрытии сущности. Компьютерный программный продукт может включать в себя машиночитаемый носитель.

В качестве примера, а не ограничения, эти машиночитаемые носители хранения данных могут содержать RAM, ROM, EEPROM, CD-ROM или другое устройство хранения данных на оптических дисках, устройство хранения данных на магнитных дисках или другие магнитные устройства хранения, флэш-память либо любой другой носитель, который может быть использован для того, чтобы сохранять требуемый программный код в форме инструкций или структур данных, и к которому можно осуществлять доступ посредством компьютера. Также любое подключение корректно называть машиночитаемым носителем. Например, если инструкции передаются из веб-узла, сервера или другого удаленного источника с помощью коаксиального кабеля, оптоволоконного кабеля, "витой пары", цифровой абонентской линии (DSL) или беспроводных технологий, таких как инфракрасные, радиопередающие и микроволновые среды, то коаксиальный кабель, оптоволоконный кабель, "витая пара", DSL или беспроводные технологии, такие как инфракрасные, радиопередающие и микроволновые среды, включаются в определение носителя. Тем не менее, следует понимать, что машиночитаемые носители хранения данных и носители хранения данных не включают в себя соединения, несущие, сигналы или другие энергозависимые носители, а вместо этого направлены на энергонезависимые материальные носители хранения данных. Диск (disk) и диск (disc) при использовании в данном документе включают в себя компакт-диск (CD), лазерный диск, оптический диск, универсальный цифровой диск (DVD), гибкий диск и диск Blu-Ray, при этом диски (disk) обычно воспроизводят данные магнитно, тогда как диски (disc) обычно воспроизводят данные оптически с помощью лазеров. Комбинации вышеперечисленного также следует включать в число машиночитаемых носителей.

Инструкции могут выполняться посредством одного или более процессоров, например одного или более процессоров цифровых сигналов (DSP), микропроцессоров общего назначения, специализированных интегральных схем (ASIC), программируемых пользователем вентильных матриц (FPGA) либо других эквивалентных интегральных или дискретных логических схем. Соответственно термин "процессор" при использовании в данном документе может означать любую вышеуказанную структуру или другую структуру, подходящую для реализации технологий, описанных в данном документе. Помимо этого, в некоторых аспектах функциональность, описанная в данном документе, может быть предоставлена в рамках специализированных программных и/или аппаратных модулей, выполненных с возможностью кодирования или декодирования либо встроенных в комбинированный кодек. Кроме того, технологии могут быть полностью реализованы в одной или более схем или логических элементов.

Технологии этого раскрытия сущности могут быть реализованы в широком спектре устройств или приборов, в том числе в беспроводном переносном телефоне, в интегральной схеме (IC) или в наборе IC (к примеру, в наборе микросхем). Различные компоненты, модули или блоки описываются в этом раскрытии сущности для того, чтобы подчеркивать функциональные аспекты устройств, выполненных с возможностью осуществлять раскрытые технологии, но необязательно требуют реализации посредством различных аппаратных модулей. Наоборот, как описано выше, различные блоки могут быть комбинированы в аппаратный модуль кодека или предоставлены посредством набора взаимодействующих аппаратных модулей, включающих в себя один или более процессоров, как описано выше, в сочетании с надлежащим программным обеспечением и/или микропрограммным обеспечением.

Описаны различные примеры. Эти и другие примеры находятся в пределах объема прилагаемой формулы изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ декодирования видеоданных, которые были кодированы в палитровом режиме, при этом способ содержит этапы, на которых

формируют палитру, которая включает в себя записи, представляющие пиксельные значения для блока видеоданных, при этом палитра содержит наиболее доминирующие пиксельные значения в текущей единице кодирования (CU), при этом для каждого пиксельного местоположения в CU указывается, включено или нет пиксельное значение в палитру;

определяют то, что блок видеоданных включает в себя один или более пикселей, кодированных в качестве пикселей с управляющим кодом, причем пиксели с управляющим кодом представляют собой пиксели, которые не существуют в палитре, и этим пикселям назначается индекс и фактическое пиксельное значение, переданное для данного пиксельного местоположения в единице кодирования (CU), и при этом, когда пиксел кодируется в качестве пиксела с управляющим кодом, значение пиксела передается в служебных сигналах непосредственно;

декодируют для каждого из одного или более пикселей, кодированных в качестве пикселей с управляющим кодом, экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пиксела, при этом декодирование экспоненциального кодового слова Голомба содержит этап, на котором декодируют экспоненциальное кодовое слово Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3;

декодируют блок с использованием значения для каждого из одного или более пикселей, кодированных в качестве пикселей с управляющим кодом.

2. Способ по п.1, дополнительно содержащий этапы, на которых

определяют значение размера шага квантования для кодированного в палитровом режиме блока с использованием параметра квантования для кодированного в палитровом режиме блока; и

определяют максимальное квантованное значение для пиксела с управляющим кодом кодированного в палитровом режиме блока на основе значения размера шага квантования,

при этом декодирование экспоненциального кодового слова Голомба содержит этап, на котором декодируют экспоненциальное кодовое слово Голомба с использованием максимального квантованного значения.

3. Способ по п.2, в котором определение максимального квантованного значения содержит этап, на котором определяют максимальное квантованное значение только на основе значения размера шага квантования.

4. Способ по п.2, в котором определение максимального квантованного значения содержит этап, на котором определяют максимальное квантованное значение на основе значения размера шага квантования, значения масштабирования при квантовании, значения смещения со сдвигом вправо и значения параметра сдвига вправо, причем способ дополнительно содержит этап, на котором определяют значение масштабирования при квантовании, значение смещения со сдвигом вправо и значение параметра сдвига вправо из таблицы поиска.

5. Способ по п.2, в котором определение максимального квантованного значения содержит этап, на котором извлекают максимальное квантованное значение из таблицы поиска, индексированной посредством параметра квантования.

6. Способ по п.5, дополнительно содержащий этап, на котором определяют параметр квантования на основе цветового компонента, который включает в себя пиксел с управляющим кодом.

7. Способ по п.1, в котором по меньшей мере часть значения содержит вторую часть значения, причем способ дополнительно содержит этапы, на которых

декодируют кодовое слово Голомба-Райса, представляющее первую часть значения для пиксела с управляющим кодом;

комбинируют первую часть и вторую часть для того, чтобы формировать значение для пиксела с управляющим кодом.

8. Устройство для декодирования видеоданных, которые были кодированы в палитровом режиме, при этом устройство содержит

средство для формирования палитры, которая включает в себя записи, представляющие пиксельные значения для блока видеоданных, при этом палитра содержит наиболее доминирующие пиксельные значения в текущей единице кодирования (CU), при этом для каждого пиксельного местоположения в CU указывается, включено или нет пиксельное значение в палитру;

средство для определения того, что блок видеоданных включает в себя один или более пикселей, кодированных в качестве пикселей с управляющим кодом, причем пиксели с управляющим кодом представляют собой пиксели, которые не существуют в палитре, и этим пикселям назначается индекс и фактическое пиксельное значение, переданное для данного пиксельного местоположения в единице кодирования (CU), и при этом, когда пиксел кодируется в качестве пиксела с управляющим кодом, значение пиксела передается в служебных сигналах непосредственно;

средство для декодирования для каждого из одного или более пикселей, кодированных в качестве

пикселей с управляющим кодом, экспоненциального кодового слова Голомба, представляющего по меньшей мере часть значения для пикселя, при этом средство для декодирования экспоненциального кодового слова Голомба содержит средство для декодирования экспоненциального кодового слова Голомба с использованием декодирования экспоненциальным кодом Голомба с параметром 3;

средство для декодирования блока с использованием значения для каждого из одного или более пикселей, кодированных в качестве пикселей с управляющим кодом.

9. Устройство по п.8, дополнительно содержащее

средство для определения значения размера шага квантования для кодированного в палитровом режиме блока с использованием параметра квантования для кодированного в палитровом режиме блока;

средство для определения максимального квантованного значения для пикселя с управляющим кодом кодированного в палитровом режиме блока на основе значения размера шага квантования,

при этом средство для декодирования экспоненциального кодового слова Голомба содержит средство для декодирования экспоненциального кодового слова Голомба с использованием максимального квантованного значения.

10. Устройство по п.8, дополнительно содержащее

средство для декодирования кодового слова Голомба-Райса, представляющего первую часть значения для пикселя с управляющим кодом;

средство для комбинирования первой части и второй части для того, чтобы формировать значение для пикселя с управляющим кодом.

11. Способ кодирования видеоданных в палитровом режиме, при этом способ содержит этапы, на которых

формируют палитру, которая включает в себя записи, представляющие пиксельные значения для блока видеоданных, при этом палитра содержит наиболее доминирующие пиксельные значения в текущей единице кодирования (CU), при этом для каждого пиксельного местоположения в CU указывается, включено или нет пиксельное значение в палитру;

определяют значение для каждого из одного или более пикселей блока видеоданных, подлежащего кодированию в качестве пикселя с управляющим кодом, при этом пиксели с управляющим кодом представляют собой пиксели, которые не существуют в палитре, и этим пикселям назначается индекс и фактическое пиксельное значение, переданное для данного пиксельного местоположения в единице кодирования (CU), и при этом, когда пиксел кодируется в качестве пикселя с управляющим кодом, значение пикселя передается в служебных сигналах непосредственно;

кодируют для каждого из одного или более пикселей, подлежащих кодированию в качестве пикселей с управляющим кодом, экспоненциальное кодовое слово Голомба, представляющее по меньшей мере часть значения для пикселя, при этом кодирование экспоненциального кодового слова Голомба содержит этап, на котором кодируют экспоненциальным кодом Голомба с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

12. Способ по п.11, дополнительно содержащий этапы, на которых

определяют значение размера шага квантования для кодированного в палитровом режиме блока с использованием параметра квантования для кодированного в палитровом режиме блока;

определяют максимальное квантованное значение для пикселя с управляющим кодом кодированного в палитровом режиме блока на основе значения размера шага квантования,

при этом кодирование экспоненциального кодового слова Голомба содержит этап, на котором кодируют экспоненциальное кодовое слово Голомба с использованием максимального квантованного значения.

13. Способ по п.12, в котором определение максимального квантованного значения содержит этап, на котором определяют максимальное квантованное значение только на основе значения размера шага квантования.

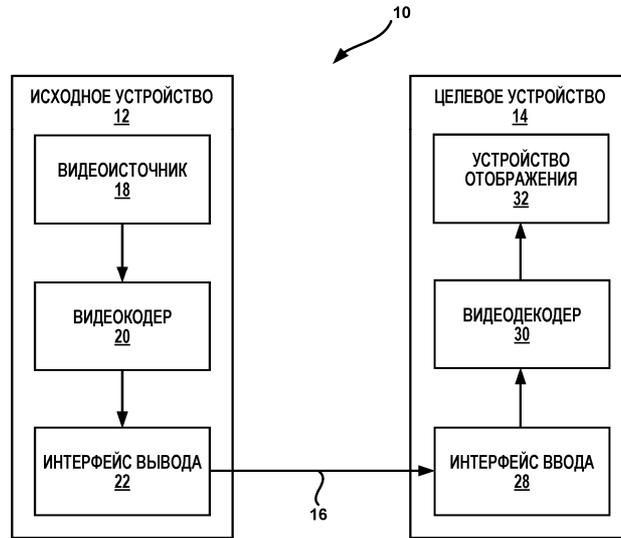
14. Устройство для кодирования видеоданных в палитровом режиме, при этом устройство содержит

средство для формирования палитры, которая включает в себя записи, представляющие пиксельные значения для блока видеоданных, при этом палитра содержит наиболее доминирующие пиксельные значения в текущей единице кодирования (CU), при этом для каждого пиксельного местоположения в CU указывается, включено или нет пиксельное значение в палитру;

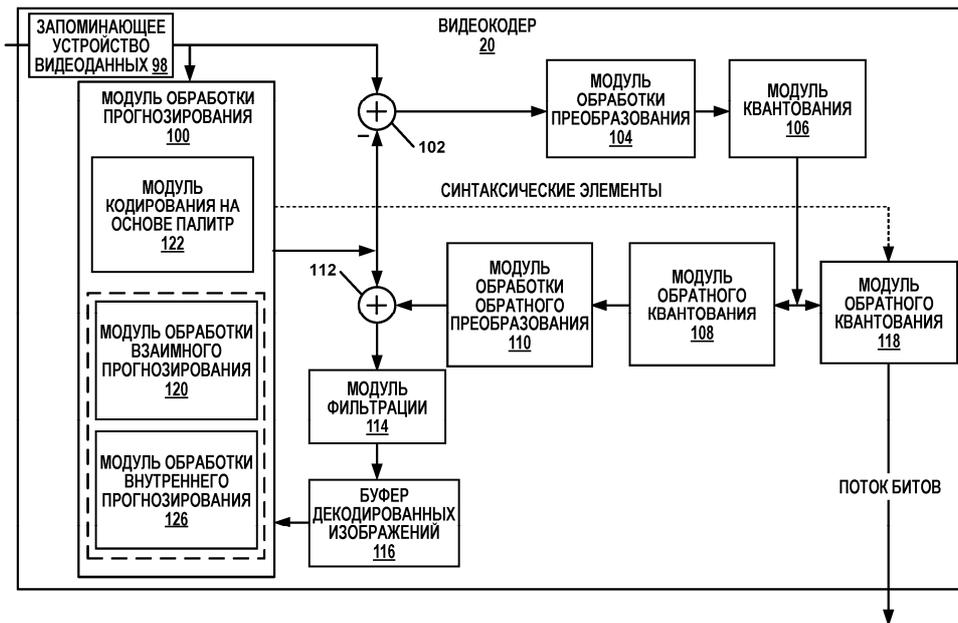
средство для определения значения для каждого из одного или более пикселей блока видеоданных, подлежащих кодированию в качестве пикселей с управляющим кодом, при этом пиксели с управляющим кодом представляют собой пиксели, которые не существуют в палитре, и этим пикселям назначается индекс и фактическое пиксельное значение, переданное для данного пиксельного местоположения в единице кодирования (CU), и при этом, когда пиксел кодируется в качестве пикселя с управляющим кодом, значение пикселя передается в служебных сигналах непосредственно;

средство для кодирования для каждого из одного или более пикселей, подлежащих кодированию в качестве пикселей с управляющим кодом, экспоненциального кодового слова Голомба, представляющего вторую часть значения для пикселя с управляющим кодом, при этом средство для кодирования экспоненциального кодового слова Голомба содержит средство для кодирования экспоненциального кодового слова Голомба с использованием кодирования экспоненциальным кодом Голомба с параметром 3.

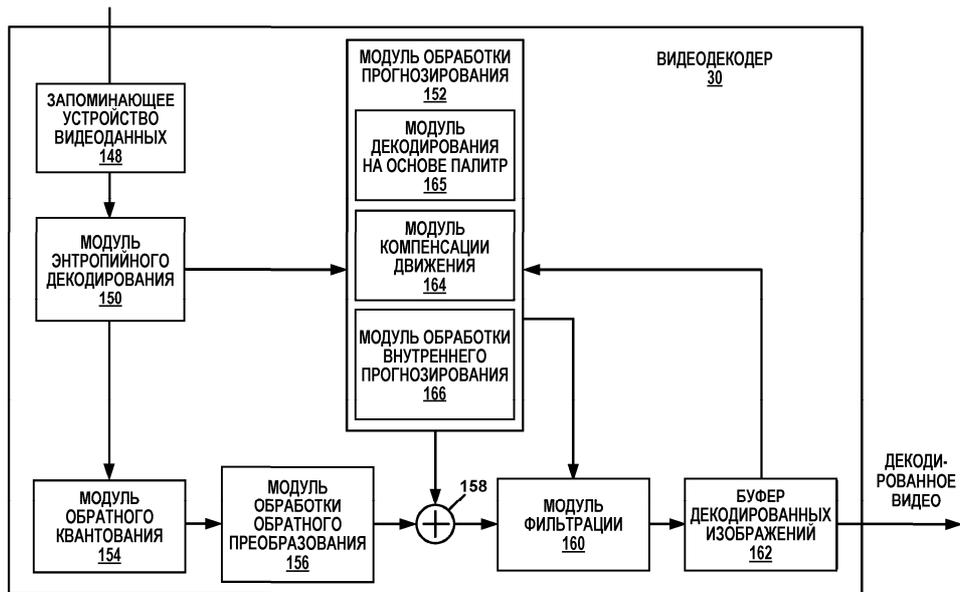
15. Машиночитаемый носитель хранения данных с программным кодом, который содержит инструкции, для одного или более процессоров устройства из пп.8, 14 кодирования видео для осуществления операций способа по любому из пп.1-7 или 11-13.



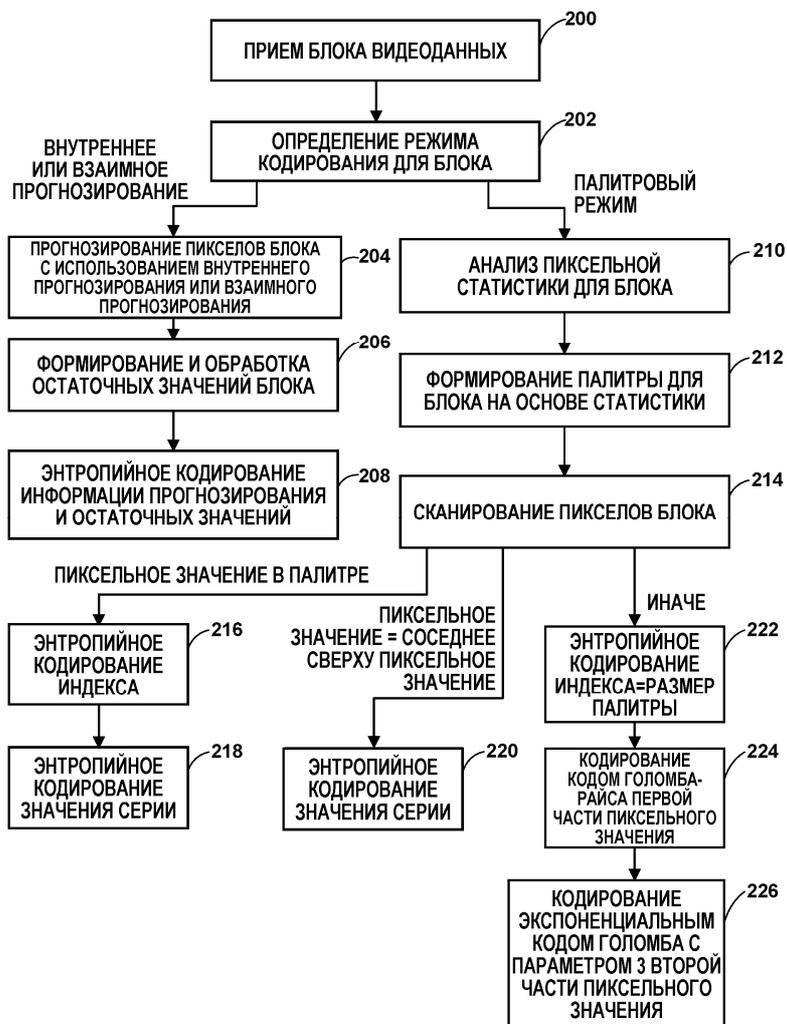
Фиг. 1



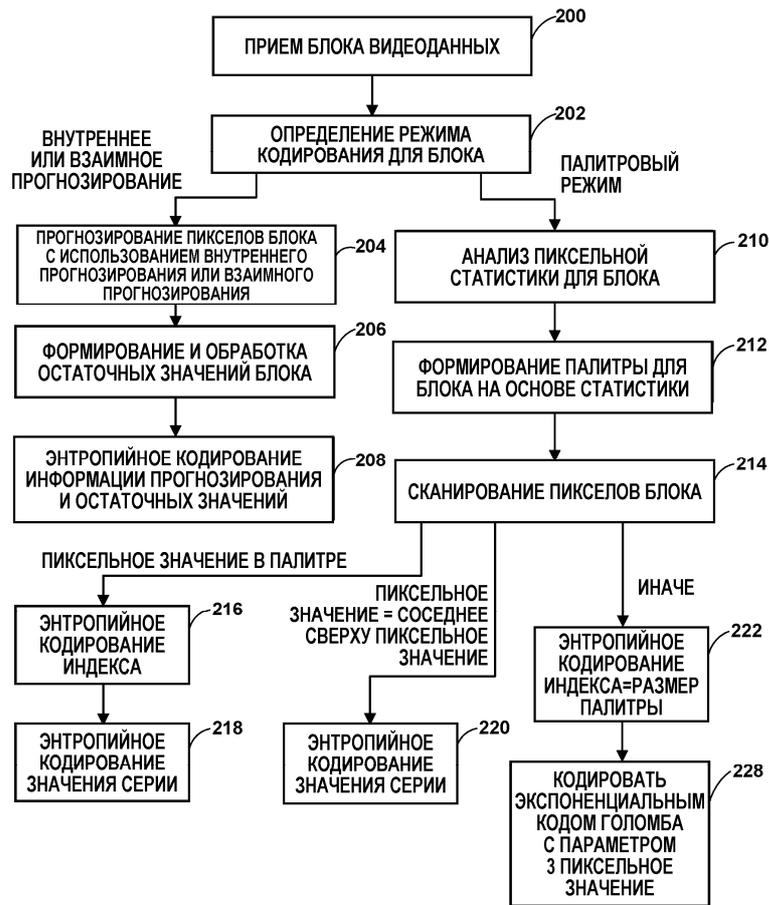
Фиг. 2



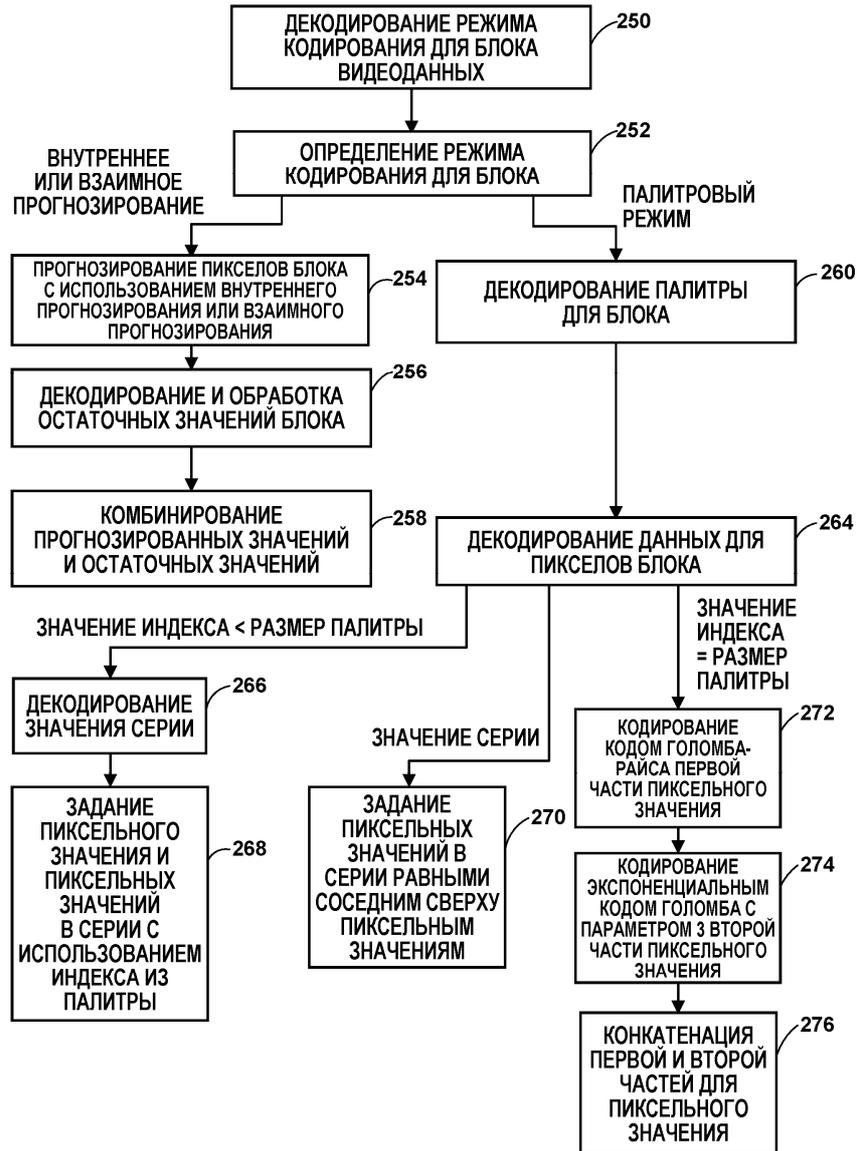
Фиг. 3



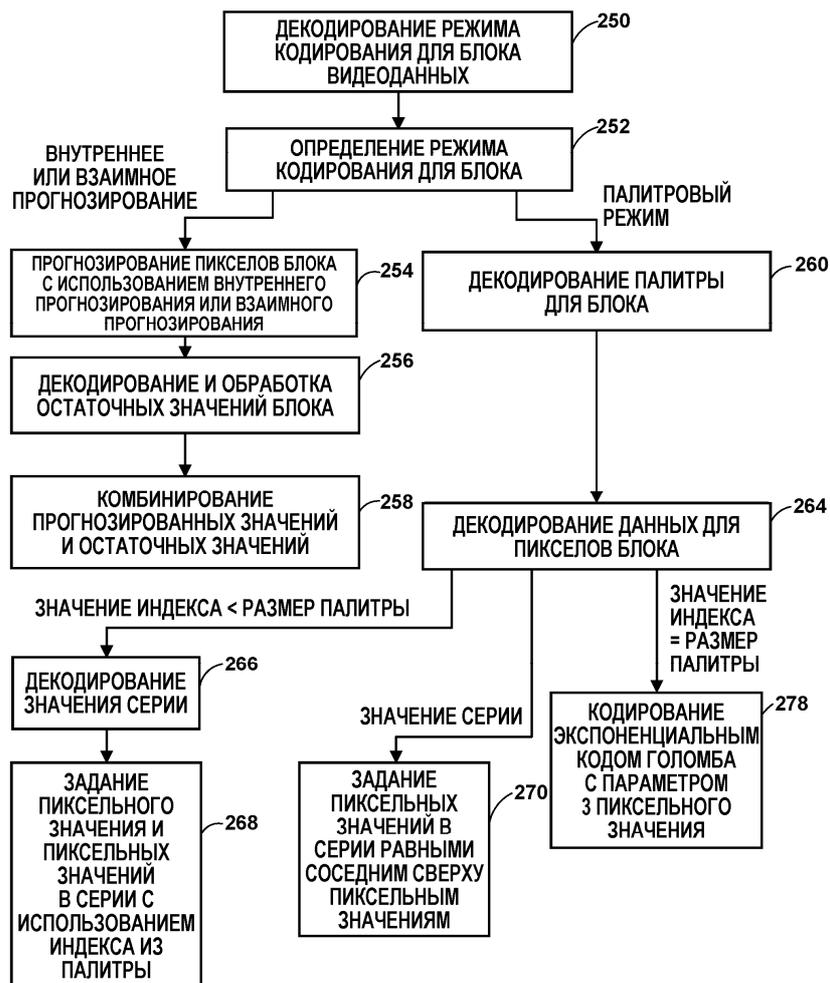
Фиг. 4



Фиг. 5



Фиг. 6



Фиг. 7

