

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
H04N 19/436 (2023.01); H04N 19/52 (2023.01)

(21)(22) Заявка: 2020143566, 10.07.2019

(24) Дата начала отсчета срока действия патента:
10.07.2019Дата регистрации:
14.07.2023

Приоритет(ы):

(30) Конвенционный приоритет:
10.07.2018 US 62/696,281;
02.08.2018 US 62/713,944;
09.07.2019 US 16/506,720

(43) Дата публикации заявки: 10.08.2022 Бюл. № 22

(45) Опубликовано: 14.07.2023 Бюл. № 20

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 10.02.2021(86) Заявка РСТ:
US 2019/041241 (10.07.2019)(87) Публикация заявки РСТ:
WO 2020/014389 (16.01.2020)Адрес для переписки:
129090, Москва, ул. Б.Спасская, 25, строение 3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

ФАМ ВАН, Луонг (US),
ЧИЭНЬ, Вэй-Цзюнь (US),
СЕРЕГИН, Вадим (US),
КАРЧЕВИЧ, Марта (US),
ХУАН, Хань (US)

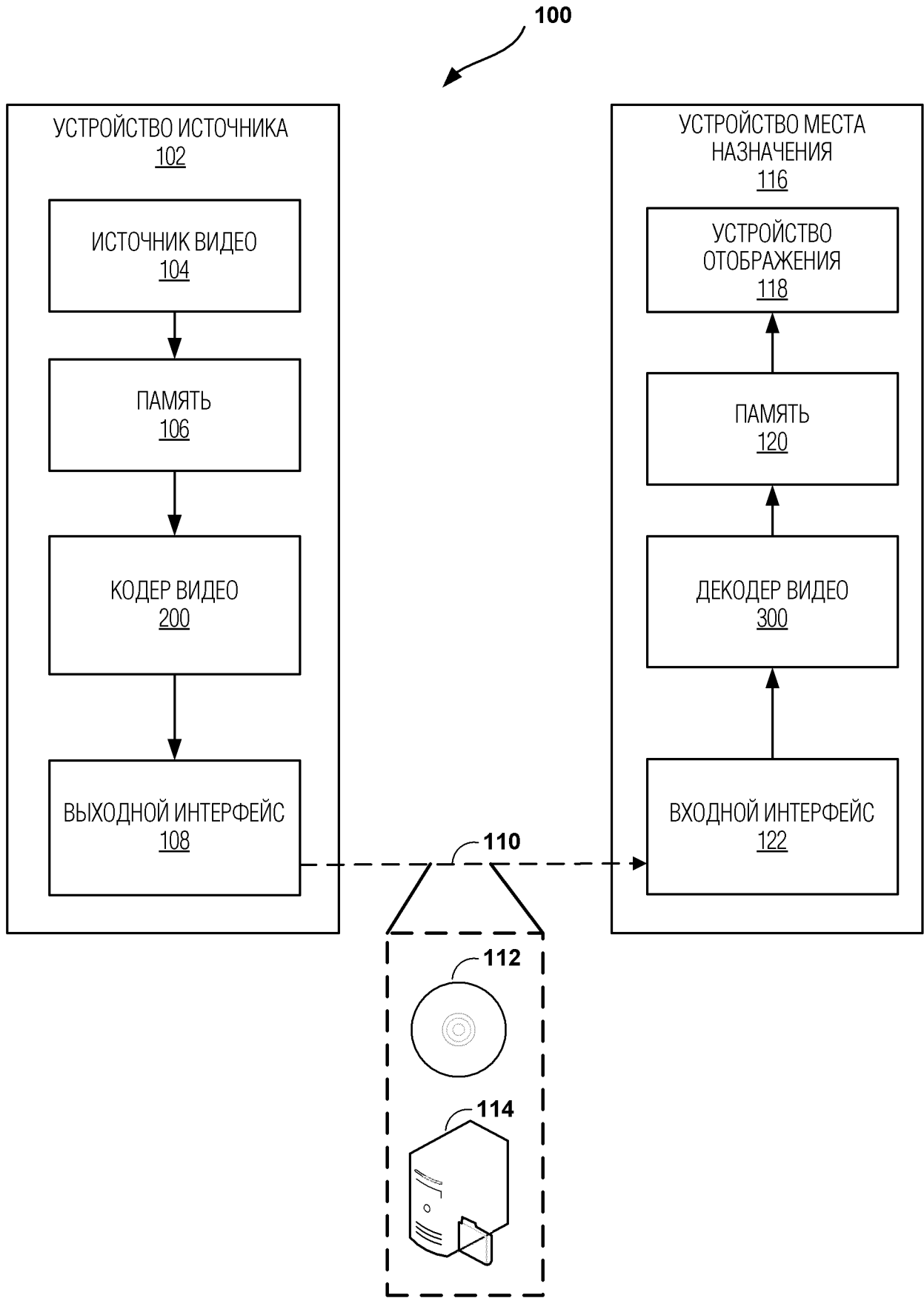
(73) Патентообладатель(и):

КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

(56) Список документов, цитированных в отчете
о поиске: Fan Wang et al., OpenCL based high-
quality HEVC motion estimation on GPU, 2014
IEEE International Conference on Image
Processing, 29.01.2015, 5 с.. US 2007/0211801 A1,
13.09.2007. US 2013/0336405 A1, 19.12.2013. Li
Zhang et al., CE4-related: History-based Motion
Vector Prediction, Joint Video Experts Team
(JVET) of ITU-T SG 16 WP 3 and ISO/IEC (см.
прод.)(54) МНОЖЕСТВО ОСНОВАННЫХ НА ИСТОРИИ НЕСМЕЖНЫХ MVP ДЛЯ ОБРАБОТКИ
ВОЛНОВОГО ФРОНТА КОДИРОВАНИЯ ВИДЕО

(57) Реферат:

Изобретение относится к средствам
кодирования данных видео. Технический
результат – повышение эффективности
кодирования данных видео. Сохраняют
информацию движения для первой строки единиц
дерева кодирования (СТУ) картинки в первом
буфере памяти предсказателя вектора движения(MVP) истории. Сбрасывают второй буфер памяти
MVP истории. После сброса второго буфера MVP
истории, сохраняют информацию движения для
второй строки СТУ картинки во втором буфере
MVP истории, причем вторая строка СТУ
отличается от первой строки СТУ. 4 н. и 54 з.п.
ф-лы, 21 ил.



ФИГ. 1

(56) (продолжение):

JTC 1/SC 29/WG 11, 11th Meeting: Ljubljana, SI, 10-18 July 2018, [Найдено 01.02.2023] в Интернет URL http://phenix.int-evry.fr/jvet/doc_end_user/documents/11_Ljubljana/wg11/JVET-K0104-v1.zip, 03.07.2018, 5 с.. WO 2017/197126 A1, 16.11.2017. US 2013/0170549 A1, 04.07.2013. Stefan Radicke et al., A Multi-Threaded Full-feature HEVC Encoder Based on Wavefront Parallel Processing, In Proceedings of the 11th International Conference on Signal Processing and Multimedia Applications (SIGMAP-2014), с. 90-98, 28.08.2014, 9 с.. RU 2573744 C2, 27.01.2016.

R U 2 8 0 0 0 9 C 2

R U 2 8 0 0 0 9 C 2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.
H04N 19/436 (2014.01)
H04N 19/52 (2014.01)

(12) **ABSTRACT OF INVENTION**

(52) CPC
H04N 19/436 (2023.01); H04N 19/52 (2023.01)

(21)(22) Application: **2020143566, 10.07.2019**

(24) Effective date for property rights:
10.07.2019

Registration date:
14.07.2023

Priority:

(30) Convention priority:
10.07.2018 US 62/696,281;
02.08.2018 US 62/713,944;
09.07.2019 US 16/506,720

(43) Application published: **10.08.2022 Bull. № 22**

(45) Date of publication: **14.07.2023 Bull. № 20**

(85) Commencement of national phase: **10.02.2021**

(86) PCT application:
US 2019/041241 (10.07.2019)

(87) PCT publication:
WO 2020/014389 (16.01.2020)

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

(72) Inventor(s):

PHAM VAN, Luong (US),
CHIEN, Wei-Jung (US),
SEREGIN, Vadim (US),
KARCZEWICZ, Marta (US),
HUANG, Han (US)

(73) Proprietor(s):

QUALCOMM INCORPORATED (US)

(54) **SET OF HISTORY-BASED NON-ADJACENT MVP FOR VIDEO ENCODING WAVE-FRONT PROCESSING**

(57) Abstract:

FIELD: video data encoding.

SUBSTANCE: motion information for the first row of coding tree units (CTU) of the picture is stored in the first history motion vector predictor (MVP) memory buffer. Flush the second history MVP memory buffer. After flushing the second history MVP buffer, the motion information for the second line of the picture

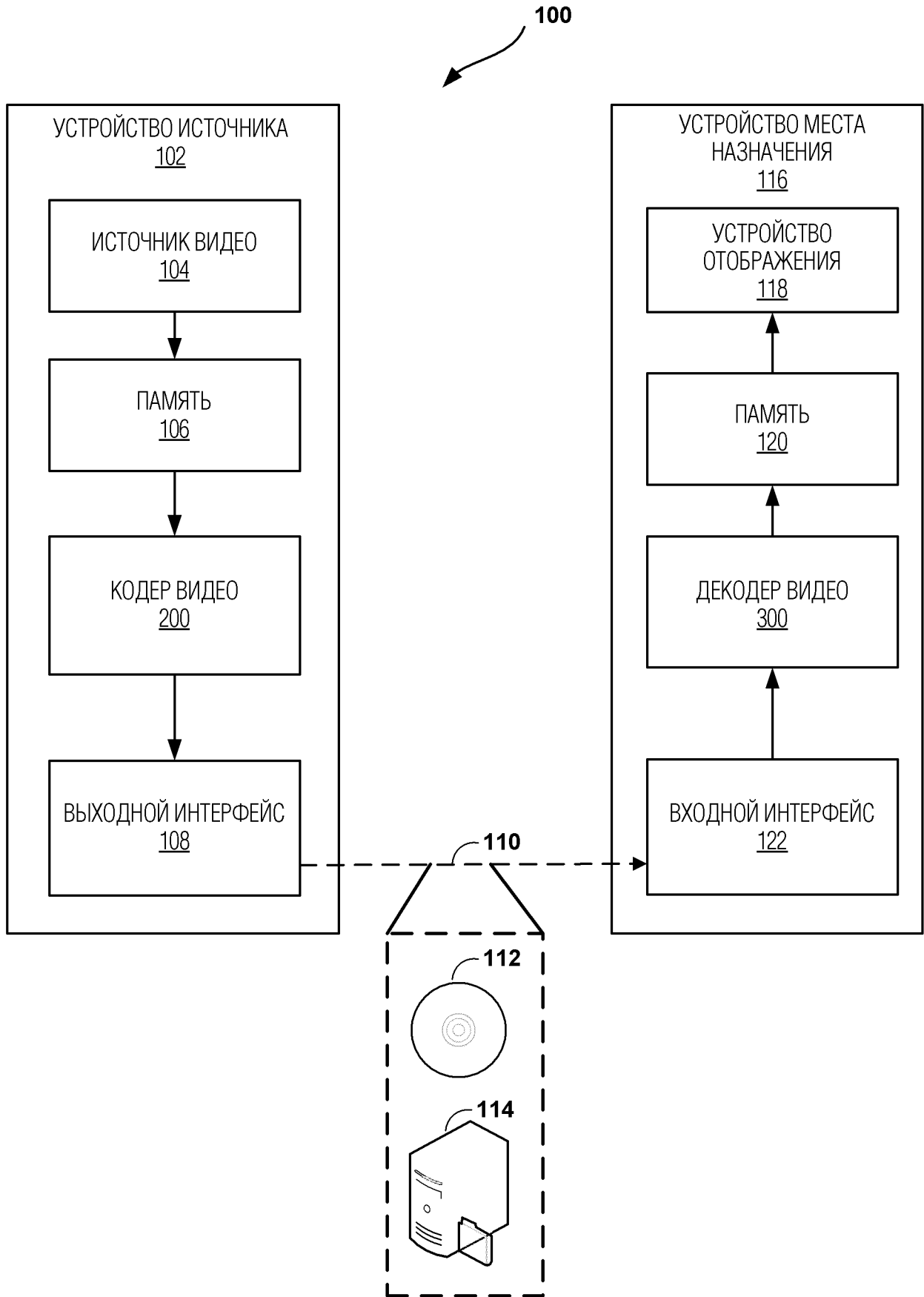
CTU is stored in the second history MVP buffer, the second line of the CTU being different from the first line of the CTU.

EFFECT: increased efficiency of video data encoding.

58 cl, 21 dwg

RU 2 800 009 C2

RU 2 800 009 C2



ФИГ. 1

[0001] Настоящая заявка испрашивает приоритет предварительной заявки США № 62/696,281, поданной 10 июля 2018, предварительной заявки США № 62/713,944, поданной 2 августа 2018 и заявки США № 16/506,720 поданной 9 июля 2019, все содержание каждой из которых включено в настоящий документ посредством ссылки.

ОБЛАСТЬ ТЕХНИКИ

[0002] Настоящее раскрытие относится к кодированию видео, включая кодирование видео и декодирование видео.

Предшествующий уровень техники

[0003] Функциональные возможности цифрового видео могут быть включены в широкий диапазон устройств, в том числе, цифровые телевизоры, цифровые системы прямого вещания, системы беспроводного вещания, персональные цифровые ассистенты (PDA), ноутбуки или стационарные компьютеры, планшетные компьютеры, электронные книги, цифровые камеры, цифровые записывающие устройства, цифровые медиаплееры, видеоигровые устройства, видеоигровые консоли, сотовые или спутниковые радиотелефоны, так называемые “смартфоны”, устройства видео-телеконференций, устройства потоковой передачи (стриминга) видео и тому подобное. Цифровые видеоустройства реализуют методы кодирования видео, такие как методы, описанные в стандартах, определенных MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, часть 10, Расширенное кодирование видео (AVC), стандарте Высокоэффективное кодирование видео (HEVC), ITU-T H.265/Высокоэффективное кодирование видео (HEVC) и расширениях таких стандартов. Видеоустройства могут передавать, принимать, кодировать, декодировать и/или хранить информацию цифрового видео более эффективно путем реализации таких методов кодирования видео.

[0004] Методы кодирования видео выполняют пространственное (внутри изображения, внутрикадровое) предсказание и/или временное (между изображениями, межкадровое) предсказание, чтобы уменьшить или удалить избыточность, присущую последовательностям видео. Для кодирования видео на блочной основе, вырезка (слайс) видео (т.е. изображение видео или часть изображения видео) может разбиваться на блоки видео, которые могут также упоминаться как единицы дерева кодирования (CTU), единицы кодирования (CU) и/или узлы кодирования. Блоки видео в интра-кодированной (I) вырезке изображения кодируются с использованием пространственного предсказания по отношению к опорным выборкам в соседних блоках в том же самом изображении. Блоки видео в интер-кодированной (P или B) вырезке изображения могут использовать пространственное предсказание по отношению к опорным выборкам в соседних блоках в том же самом изображении или временное предсказание по отношению к опорным выборкам в других опорных изображениях. Изображения могут упоминаться как кадры, и опорные изображения могут упоминаться как опорные кадры.

Краткое описание сущности изобретения

[0005] В общем, настоящее раскрытие описывает технологии для кодирования информации движения блоков данных видео. Эти технологии могут использоваться во время параллельной обработки волнового фронта. Информация движения может включать в себя вектора движения, которые предсказываются из основывающихся на истории предсказателей (предикторов) векторов движения (HMVP). Потенциально подходящий (кандидат) HMVP может соответствовать информации движения ранее кодированного блока. Устройство кодирования видео (кодер или декодер) может поддерживать таблицу с множеством кандидатов HMVP во время процесса кодирования (кодирования и декодирования). Устройство кодирования видео может опустошать таблицу, когда появляется новая вырезка. Когда имеется интер-кодированный блок,

устройство кодирования видео может присоединять информацию движения, ассоциированную с интер-кодированным блоком, к таблице.

5 [0006] В одном примере, способ кодирования (кодирования или декодирования) данных видео включает в себя сохранение информации движения для первой строки единиц дерева кодирования (СТУ) изображения в первом буфере предиктора векторов движения, основывающегося на истории (MVP истории), в памяти; сброс второго буфера MVP истории в памяти; и, после сброса второго буфера MVP истории, сохранение информации движения для второй строки СТУ изображения во втором буфере MVP истории, причем вторая строка СТУ отличается от первой строки СТУ. В некоторых 10 примерах, первый подпроцесс (тред) процесса кодирования видео может кодировать первую строку СТУ, и второй подпроцесс процесса кодирования видео, отличающийся от первого подпроцесса, может кодировать вторую строку СТУ.

[0007] В другом примере, устройство для кодирования данных видео включает в себя память, сконфигурированную, чтобы хранить данные видео; и один или более 15 модулей обработки, реализованных в схемах и сконфигурированных, чтобы: сохранять информацию движения для первой строки единиц дерева кодирования (СТУ) изображения в первом буфере предиктора векторов движения, основывающегося на истории (MVP истории), в памяти; сбрасывать второй буфер памяти MVP истории; и, после сброса второго буфера MVP истории, сохранять информацию движения для 20 второй строки СТУ изображения во втором буфере MVP истории, причем вторая строка СТУ отличается от первой строки СТУ.

[0008] В другом примере, считываемый компьютером носитель данных имеет сохраненные в нем инструкции, которые, при их исполнении, предписывают процессору: 25 сохранять информацию движения для первой строки единиц дерева кодирования (СТУ) изображения в первом буфере предиктора векторов движения, основывающегося на истории (MVP истории), в памяти; сбрасывать второй буфер MVP истории в памяти; и, после сброса второго буфера MVP истории, сохранять информацию движения для второй строки СТУ изображения во втором буфере MVP истории, причем вторая строка СТУ отличается от первой строки СТУ.

30 [0009] В другом примере, устройство для кодирования данных видео включает в себя память, сконфигурированную, чтобы хранить данные видео; средство для сохранения информации движения для первой строки единиц дерева кодирования (СТУ) изображения в первом буфере предиктора векторов движения, основывающегося на истории (MVP истории), в памяти; средство для сброса второго буфера MVP истории 35 в памяти; и средство для сохранения информации движения для второй строки СТУ изображения во втором буфере MVP истории после сброса второго буфера MVP истории, причем вторая строка СТУ отличается от первой строки СТУ.

[0010] Детали одного или более примеров представлены на приложенных чертежах и в описании ниже. Другие признаки, задачи и преимущества будут очевидны из описания 40 и чертежей, а также из формулы изобретения.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

[0011] Фиг. 1 представляет собой блок-схему, иллюстрирующую примерную систему кодирования и декодирования видео, которая может выполнять методы настоящего раскрытия.

45 [0012] Фиг. 2А и 2В представляют собой концептуальные диаграммы, иллюстрирующие примерную структуру квадродерева-двоичного дерева (QTBT) и соответствующую единицу дерева кодирования (СТУ).

[0013] Фиг. 3 представляет собой блок-схему, иллюстрирующую примерный процесс

для кодирования информации движения с использованием предсказателей векторов движения, основывающихся на истории (HMVP).

[0014] Фиг. 4 представляет собой концептуальную диаграмму, иллюстрирующую пример обновления таблицы HMVP.

5 [0015] Фиг. 5 представляет собой концептуальную диаграмму, иллюстрирующую примерный выбор несмежных блоков для кодирования информации движения.

[0016] Фиг. 6 представляет собой концептуальную диаграмму, иллюстрирующую примерный выбор несмежных блоков на основе родительского (порождающего) блока.

10 [0017] Фиг. 7 представляет собой концептуальную диаграмму, иллюстрирующую пример желательной обработки волнового фронта единиц дерева кодирования (STU).

[0018] Фиг. 8 представляет собой концептуальную диаграмму, иллюстрирующую пример информации движения, используемой для HMVP.

15 [0019] Фиг. 9 представляет собой концептуальную диаграмму, иллюстрирующую пример изображения, разделенной на множество строк единиц дерева кодирования (STU).

[0020] Фиг. 10А и 10В представляют собой блочные диаграммы, иллюстрирующие примерные пространственные соседние векторы движения, являющиеся кандидатами для режимов слияния и расширенного предсказания вектора движения (AMVP).

20 [0021] Фиг. 11А и 11В представляют собой концептуальные диаграммы для иллюстрации кандидатов временного предсказания векторов движения (TMPV).

[0022] Фиг. 12 представляет собой блочную схему, иллюстрирующую пример единицы дерева кодирования (STU) и соседних блоков.

[0023] Фиг. 13 представляет собой блочную схему, иллюстрирующую текущую CU в текущей STU.

25 [0024] Фиг. 14 представляет собой блок-схему, иллюстрирующую примерный кодер видео, который может выполнять методы настоящего раскрытия.

[0025] Фиг. 15 представляет собой блок-схему, иллюстрирующую примерный декодер видео, который может выполнять методы настоящего раскрытия.

30 [0026] Фиг. 16 представляет собой блок-схему последовательности операций, иллюстрирующую примерный способ для кодирования текущего блока данных видео в соответствии с методами настоящего раскрытия.

[0027] Фиг. 17 представляет собой блок-схему последовательности операций, иллюстрирующую примерный способ для декодирования текущего блока данных видео в соответствии с методами настоящего раскрытия.

35 [0028] Фиг. 18 представляет собой блок-схему последовательности операций, иллюстрирующую примерный способ кодирования (кодирования или декодирования) данных видео в соответствии с методами данного раскрытия.

ДЕТАЛЬНОЕ ОПИСАНИЕ

40 [0029] Фиг. 1 представляет собой блок-схему, иллюстрирующую примерную систему 100 кодирования и декодирования видео, которая может выполнять методики согласно настоящему раскрытию. Методики по настоящему раскрытию, в основном, направлены на кодирование (кодирование и/или декодирование) данных видео. В принципе, данные видео включают в себя любые данные для обработки видео. Таким образом, данные видео могут включать необработанное, некодированное видео, кодированное видео, 45 декодированное (например, восстановленное) видео и метаданные видео, такие как данные сигнализации.

[0030] Как показано на фиг. 1, система 100 включает в себя устройство 102 источника, которое обеспечивает кодированные данные видео, подлежащие декодированию и

отображению устройством 116 места назначения (получателем), в данном примере. В частности, устройство 102 источника предоставляет данные видео на устройство 116 места назначения посредством считываемого компьютером носителя 110. Устройство 102 источника и устройство 116 места назначения могут содержать любое из широкого спектра устройств, включая настольные компьютеры, ноутбуки, планшетные компьютеры, телевизионные приставки, телефоны, такие как смартфоны, телевизоры, камеры, устройства отображения, цифровые медиаплееры, видеоигровые консоли, устройства потоковой передачи видео и тому подобное. В некоторых случаях, устройство 102 источника и устройство 116 места назначения могут быть оснащены для беспроводной связи и, таким образом, могут упоминаться как устройства беспроводной связи.

[0031] В примере на фиг. 1, устройство 102 источника включает в себя источник 104 видео, память 106, кодер 200 видео и выходной интерфейс 108. Устройство 116 места назначения включает в себя входной интерфейс 122, декодер 300 видео, память 120 и устройство 118 отображения. В соответствии с настоящим раскрытием, кодер 200 видео устройства 102 источника и декодер 300 видео устройства 116 места назначения могут быть сконфигурированы, чтобы применять методы для кодирования информации движения. Таким образом, устройство 102 источника представляет собой пример устройства кодирования видео, в то время как устройство 116 места назначения представляет собой пример устройства декодирования видео. В других примерах, устройство источника и устройство места назначения могут включать в себя другие компоненты или устройства. Например, устройство 102 источника может принимать данные видео от внешнего источника видео, такого как внешняя камера. Аналогично, устройство 116 места назначения может взаимодействовать с внешним устройством отображения вместо того, чтобы включать в себя встроенное устройство отображения.

[0032] Система 100, как показано на фиг. 1, является лишь одним примером. В принципе, любое цифровое устройство кодирования и/или декодирования видео может выполнять методы для кодирования информации движения. Устройство 102 источника и устройство 116 места назначения являются лишь примерами таких кодирующих устройств, в которых устройство 102 источника генерирует закодированные данные видео для передачи на устройство 116 места назначения. Настоящее раскрытие ссылается на “кодирующее” устройство как на устройство, которое выполняет кодирование (кодирование и/или декодирование) данных. Таким образом, кодер 200 видео и декодер 300 видео представляют собой примеры кодирующих устройств, в частности, кодер видео и декодер видео, соответственно. В некоторых примерах, устройство 102 источника и устройство 116 места назначения могут работать по существу симметрично, так что каждое из устройства 102 источника и устройства 116 места назначения включает в себя компоненты кодирования и декодирования видео. Следовательно, система 100 может поддерживать однонаправленную или двунаправленную передачу видео между устройством 102 источника и устройством 116 места назначения, например, для потоковой передачи видео, воспроизведения видео, видеовещания или видеотелефонии.

[0033] В общем случае, источник 104 видео представляет источник данных видео (т.е. необработанных, некодированных данных видео) и предоставляет последовательность картинок (также упоминаемых как “кадры”) данных видео на кодер 200 видео, который кодирует данные для картинок. Источник 104 видео устройства 102 источника может включать в себя устройство захвата видео, такое как видеокамера, видеоархив, содержащий ранее захваченное необработанное видео, и/или интерфейс подачи видео для приема видео от поставщика видеоконтента. В качестве

дополнительной альтернативы, источник 104 видео может генерировать данные на основе компьютерной графики в качестве исходного видео или комбинацию живого видео, архивного видео и генерируемого компьютером видео. В каждом случае, кодер 200 видео кодирует захваченные, предварительно захваченные или сгенерированные компьютером данные видео. Кодер 200 видео может переупорядочивать изображения из принятого порядка (иногда называемого “порядком отображения”) в порядок кодирования для кодирования. Кодер 200 видео может генерировать битовый поток, включающий закодированные данные видео. Устройство 102 источника может затем выводить закодированные данные видео через выходной интерфейс 108 на считываемый компьютером носитель 110 для приема и/или извлечения данных, например, входным интерфейсом 122 устройства 116 места назначения.

[0034] Память 106 устройства 102 источника и память 120 устройства 116 места назначения представляют собой память общего назначения. В некоторых примерах, памяти 106 и 120 могут хранить необработанные данные видео, например необработанное видео из источника 104 видео и необработанные декодированные данные видео из декодера 300 видео. Дополнительно или альтернативно, памяти 106 и 120 могут хранить программные инструкции, исполняемые, например, кодером 200 видео и декодером 300 видео соответственно. Хотя показано отдельно от кодера 200 видео и декодера 300 видео, следует понимать, что кодер 200 видео и декодер 300 видео также могут включать в себя внутреннюю память для функционально сходных или эквивалентных целей. Кроме того, в памяти 106 и 120 могут храниться закодированные данные видео, например, выход с кодера 200 видео и вход в декодер 300 видео. В некоторых примерах, части памяти 106 и 120 могут быть выделены в качестве одного или более буферов видео, например, для хранения необработанных, декодированных и/или закодированных данных видео.

[0035] Считываемый компьютером носитель 110 может представлять собой любой тип носителя или устройства, способного транспортировать закодированные данные видео от устройства 102 источника к устройству 116 места назначения. В одном примере, считываемый компьютером носитель 110 представляет коммуникационную среду, позволяющую устройству 102 источника передавать закодированные данные видео непосредственно на устройство 116 места назначения в реальном времени, например, через радиочастотную сеть или компьютерную сеть. Выходной интерфейс 108 может модулировать сигнал передачи, включающий в себя закодированные данные видео, а входной интерфейс 122 может демодулировать принятый сигнал передачи в соответствии со стандартом связи, таким как протокол беспроводной связи. Коммуникационная среда может содержать любую беспроводную или проводную коммуникационную среду, такую как радиочастотный (RF) спектр или одна или более физических линий передачи. Коммуникационная среда может формировать часть пакетной сети, такой как локальная сеть, сеть широкого охвата или глобальная сеть, такая как Интернет. Коммуникационная среда может включать в себя маршрутизаторы, коммутаторы, базовые станции или любое другое оборудование, которое может быть полезно для облегчения связи от устройства 102 источника к устройству 116 места назначения.

[0036] В некоторых примерах, устройство 102 источника может выводить закодированные данные из выходного интерфейса 108 в устройство 112 хранения. Аналогично, устройство 116 места назначения может получать доступ к закодированным данным из устройства 112 хранения через входной интерфейс 122. Устройство 112 хранения может включать в себя любое из множества распределенных или локально доступных носителей хранения данных, таких как накопитель на жестком диске, Blu-

гау диски, DVD, CD-ROM, флэш-память, энергозависимая или энергонезависимая память или любые другие подходящие цифровые носители хранения для хранения закодированных данных видео.

[0037] В некоторых примерах, устройство 102 источника может выводить кодированные данные видео на файловый сервер 114 или другое промежуточное устройство хранения, которое может хранить закодированное видео, сгенерированное устройством 102 источника. Устройство 116 места назначения может получать доступ к сохраненным данным видео с файлового сервера 114 посредством потоковой передачи или загрузки. Файловый сервер 114 может быть любым типом серверного устройства, способного хранить закодированные данные видео и передавать эти закодированные данные видео на устройство 116 места назначения. Файловый сервер 114 может представлять собой веб-сервер (например, для веб-сайта), сервер протокола передачи файлов (FTP), сетевое устройство доставки контента или сетевое устройство хранения (NAS). Устройство 116 места назначения может получать доступ к закодированным данным видео с файлового сервера 114 через любое стандартное соединение передачи данных, включая Интернет-соединение. Это может включать в себя беспроводной канал (например, соединение Wi-Fi), проводное соединение (например, цифровую абонентскую линию (DSL), кабельный модем и т.д.) или комбинацию обоих, которая подходит для доступа к закодированным данным видео, хранящимся на файловом сервере 114. Файловый сервер 114 и входной интерфейс 122 могут быть сконфигурированы для работы в соответствии с протоколом потоковой передачи, протоколом передачи загрузки или их комбинацией.

[0038] Выходной интерфейс 108 и входной интерфейс 122 могут представлять собой беспроводные передатчики/приемники, модемы, проводные сетевые компоненты (например, Ethernet-карты), компоненты беспроводной связи, работающие в соответствии с любым из множества стандартов IEEE 802.11, или другие физические компоненты. В примерах, где выходной интерфейс 108 и входной интерфейс 122 содержат беспроводные компоненты, выходной интерфейс 108 и входной интерфейс 122 могут быть сконфигурированы для передачи данных, таких как закодированные данные видео, в соответствии со стандартом сотовой связи, таким как 4G, 4G-LTE (Долгосрочное развитие), LTE-Advanced (Расширенное LTE), 5G или тому подобное. В некоторых примерах, если выходной интерфейс 108 содержит беспроводной передатчик, выходной интерфейс 108 и входной интерфейс 122 могут быть сконфигурированы для передачи данных, таких как закодированные данные видео, в соответствии с другими беспроводными стандартами, такими как спецификация IEEE 802.11, спецификация IEEE 802.15 (например, ZigBee™), стандарт Bluetooth™ или тому подобное. В некоторых примерах, устройство 102 источника и/или устройство 116 места назначения могут включать в себя соответствующие однокристальные (SoC) устройства. Например, устройство 102 источника может включать в себя SoC-устройство для выполнения функциональности, относящейся к кодированию видео и/или выходному интерфейсу 108, и устройство 116 места назначения может включать в себя SoC-устройство для выполнения функциональности, относящейся к декодированию видео и/или входному интерфейсу 122.

[0039] Технологии согласно настоящему раскрытию могут быть применены к кодированию видео при поддержке любого из множества мультимедийных приложений, таких как эфирные телевизионные передачи, передачи кабельного телевидения, передачи спутникового телевидения, потоковые передачи видео через Интернет, такие как динамическая адаптивная потоковая передача по HTTP (DASH), цифровое видео,

которое закодировано на носителе хранения данных, декодирование цифрового видео, сохраненного на носителе хранения данных, или других приложений.

[0040] Входной интерфейс 122 устройства 116 места назначения принимает закодированный битовый поток видео со считываемого компьютером носителя 110 (например, устройства 112 хранения, файлового сервера 114 или тому подобного). Закодированный битовый поток видео может включать в себя информацию сигнализации, определенную кодером 200 видео, которая также используется декодером 300 видео, такую как синтаксические элементы, имеющие значения, которые описывают характеристики и/или обработку блоков видео или других кодированных единиц (например, вырезок, картинок, групп картинок, последовательностей или тому подобное). Устройство 118 отображения отображает пользователю декодированные изображения декодированных данных видео. Устройство 118 отображения может представлять собой любое из множества устройств отображения, таких как электронно-лучевая трубка (CRT), жидкокристаллический дисплей (LCD), плазменный дисплей, дисплей на органических светоизлучающих диодах (OLED) или другой тип устройства отображения.

[0041] Хотя не показано на фиг. 1, в некоторых примерах, кодер 200 видео и декодер 300 видео могут быть интегрированы с кодером аудио и/или декодером аудио и могут включать соответствующие модули мультиплексирования-демультиплексирования (MUX-DEMUX) или другие аппаратные средства и/или программное обеспечение для обработки мультиплексированных потоков, включающих в себя как аудио, так и видео, в общем потоке данных. Если применимо, модули MUX-DEMUX могут соответствовать протоколу мультиплексора ITU H.223 или другим протоколам, таким как протокол пользовательских дейтаграмм (UDP).

[0042] Как кодер 200 видео, так и декодер 300 видео могут быть реализованы как любая из множества подходящих схем кодера и/или декодера, таких как один или более микропроцессоров, цифровых сигнальных процессоров (DSP), специализированных интегральных схем (ASIC), программируемых вентильных матриц (FPGA), дискретной логики, программного обеспечения, аппаратных средств, микропрограммного обеспечения или любых их комбинаций. Когда методы частично реализованы в программном обеспечении, устройство может хранить инструкции для программного обеспечения на подходящем, не-временном считываемом компьютером носителе и исполнять инструкции в аппаратных средствах с использованием одного или более процессоров для выполнения методов настоящего раскрытия. Каждый из кодера 200 видео и декодера 300 видео может быть включен в один или более кодеров или декодеров, любой из которых может быть интегрирован как часть комбинированного кодера/декодера (CODEC) в соответствующем устройстве. Устройство, включающее в себя кодер 200 видео и/или декодер 300 видео, может содержать интегральную схему, микропроцессор и/или устройство беспроводной связи, такое как сотовый телефон.

[0043] Кодер 200 видео и декодер 300 видео могут работать в соответствии со стандартом кодирования видео, таким как ITU-T H. 265, также упоминаемым как Высокоэффективное кодирование видео (HEVC), или его расширениями, такими как расширения многовидового и/или масштабируемого кодирования видео. В качестве альтернативы, кодер 200 видео и декодер 300 видео могут работать в соответствии с другими проприетарными или отраслевыми стандартами, такими как Тестовая модель совместного исследования (Joint Exploration Test Model, JEM) или ITU-T H.266, также упоминаемыми как Многоцелевое кодирование видео (Versatile Video Coding, VVC). Недавний проект стандарта VVC описан в Bross, et al. "Versatile Video Coding (Draft 5)",

Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14th Meeting: Geneva, CH, 19-27 March 2019, JVET-N1001-v3 (далее “VVC Draft 5”). Однако методы настоящего раскрытия не ограничиваются каким-либо конкретным стандартом кодирования.

5 [0044] В общем, кодер 200 видео и декодер 300 видео могут выполнять блочное кодирование картинок. Термин “блок” обычно относится к структуре, включающей
данные, подлежащие обработке (например, кодированию, декодированию или
использованию иным образом в процессе кодирования и/или декодирования). Например,
10 блок может включать в себя двумерную матрицу выборок данных яркости и/или
цветности. В общем, кодер 200 видео и декодер 300 видео могут кодировать данные
видео, представленные в формате YUV (например, Y, Cb, Cr). То есть вместо
кодирования данных красного, зеленого и синего (RGB) для выборок изображения,
кодер 200 видео и декодер 300 видео может кодировать компоненты в себя яркости и
15 цветности, где компоненты цветности могут включать компоненты цветности как
красного оттенка, так и синего оттенка. В некоторых примерах, кодер 200 видео
преобразует принятые данные в формате RGB в представление YUV перед
кодированием, а декодер 300 видео преобразует представление YUV в формат RGB.
Альтернативно, эти преобразования могут выполняться модулями пред- и пост-
обработки (не показаны).

20 [0045] Настоящее раскрытие может, в общем, ссылаться на кодирование (например,
кодирование и декодирование) картинок, включающее в себя процесс кодирования или
декодирования данных изображения. Аналогично, настоящее раскрытие может
ссылаться на кодирование блоков картинок, включающее в себя процесс кодирования
или декодирования данных для блоков, например, кодирование предсказания и/или
25 остатка. Кодированный битовый поток видео обычно включает в себя
последовательность значений для синтаксических элементов, представляющих решения
кодирования (например, режимы кодирования) и разбиение картинок на блоки. Таким
образом, ссылки на кодирование изображения или блока, в общем, следует понимать
как кодирование значений для синтаксических элементов, образующих изображение
30 или блок.

[0046] HEVC определяет различные блоки, включая единицы кодирования (CU),
единицы предсказания (PU) и единицы преобразования (TU). Согласно HEVC, устройство
кодирования видео (такое как кодер 200 видео) разбивает единицу дерева кодирования
(CTU) на CU в соответствии со структурой квадродерева. То есть устройство
35 кодирования видео разбивает CTU и CU на четыре равных, непересекающихся квадрата,
и каждый узел квадродерева имеет либо ноль, либо четыре дочерних узла. Узлы без
дочерних узлов могут упоминаться как “листовые узлы”, и CU таких листовых узлов
могут включать одну или более PU и/или одну или более TU. Устройство кодирования
40 видео может дополнительно разбивать PU и TU. Например, в HEVC, остаточное
квадродерево (RQT) представляет собой разбиение TU. В HEVC, PU представляют
данные интер-предсказания, в то время как TU представляют остаточные данные. CU,
которые являются интра-предсказанными, включают информацию интра-предсказания,
такую как указание интра-режима.

45 [0047] В качестве другого примера, кодер 200 видео и декодер 300 видео могут быть
skonфигурированы для работы в соответствии с JEM или VVC. Согласно JEM или VVC,
устройство кодирования видео (такое как кодер 200 видео) разбивает изображение на
множество единиц дерева кодирования (CTU). Кодер 200 видео может разбивать CTU
в соответствии со структурой дерева, такой как структура квадродерева-двоичного

дерева (QTBT) или структура дерева множества типов (МТТ). Структура QTBT исключает концепции множества типов разбиения, такие как разделение между CU, PU и TU в HEVC. Структура QTBT может включать в себя два уровня: первый уровень, разделенный в соответствии с разбиением квадродерева, и второй уровень, разделенный в соответствии с разбиением двоичного дерева. Корневой узел структуры QTBT соответствует STU. Листовые узлы двоичных деревьев соответствуют единицам кодирования (CU).

[0048] В структуре разбиения МТТ, блоки могут быть разделены с использованием разбиения квадродерева (QT), разбиения двоичного дерева (BT) и одного или более типов разбиений троичного дерева (TT). Разбиение троичного дерева представляет собой разбиение, когда блок разделяется на три подблока. В некоторых примерах, разбиение троичного дерева делит блок на три подблока без деления исходного блока через центр. Типы разбиения в МТТ (например, QT, BT и TT) могут быть симметричными или асимметричными.

[0049] В некоторых примерах, кодер 200 видео и декодер 300 видео могут использовать одну структуру QTBT или МТТ для представления компонентов как яркости, так и цветности, в то время как в других примерах, кодер 200 видео и декодер 300 видео могут использовать две или более структур QTBT или МТТ, например, одну структуру QTBT или МТТ для компонента яркости и другую структуру QTBT или МТТ для обоих компонентов цветности (или две структуры QTBT или МТТ для соответствующих компонентов цветности).

[0050] Кодер 200 видео и декодер 300 видео могут быть сконфигурированы для использования разбиения квадродерева в HEVC, разбиения QTBT в соответствии с JEM или других структур разбиения. В целях пояснения, описание методов настоящего раскрытия представлено в отношении разбиения QTBT. Однако следует понимать, что методы настоящего раскрытия также могут быть применены к устройствам кодирования видео, сконфигурированным для использования разбиения квадродерева или других типов разбиения.

[0051] Настоящее раскрытие может использовать “N×N” и “N на N” взаимозаменяемым образом для ссылки на размеры выборки блока (такого как CU или другой блок видео), в терминах вертикального и горизонтального размеров, например, 16×16 выборок или 16 на 16 выборок. В общем случае, CU размером 16×16 будет иметь 16 выборок в вертикальном направлении (y=16) и 16 выборок в горизонтальном направлении (x=16). Аналогично, CU размером N×N обычно имеет N выборок в вертикальном направлении и N выборок в горизонтальном направлении, где N представляет неотрицательное целое значение. Выборки в CU могут быть расположены в строках и столбцах. Кроме того, CU не обязательно должны иметь одинаковое количество выборок в горизонтальном направлении и в вертикальном. Например, CU может содержать N×M выборок, где M не обязательно равно N.

[0052] Кодер 200 видео кодирует данные видео для CU, представляющих информацию предсказания и/или остаточную информацию и другую информацию. Информация предсказания указывает, как должна предсказываться CU, чтобы формировать блок предсказания для CU. Остаточная информация обычно представляет собой разности от выборки к выборке между выборками CU перед кодированием и блоком предсказания.

[0053] Для предсказания CU, кодер 200 видео может обычно формировать блок предсказания для CU посредством интер-предсказания или интра-предсказания. Интер-предсказание обычно относится к предсказанию CU из данных ранее кодированной

изображения, в то время как интра-предсказание обычно относится к предсказанию CU из ранее кодированных данных той же самой изображения. Чтобы выполнить интер-предсказание, кодер 200 видео может генерировать блок предсказания с использованием одного или более векторов движения. Кодер 200 видео обычно может выполнять поиск движения для идентификации опорного блока, который близко согласуется с CU, например, с точки зрения разностей между CU и опорным блоком. Кодер 200 видео может вычислить метрику разности с использованием суммы абсолютных разностей (SAD), суммы квадратов разностей (SSD), средней абсолютной разности (MAD), средне-квадратичных разностей (MSD) или других подобных разностных вычислений для определения, является ли опорный блок близко согласованным с текущей CU. В некоторых примерах, кодер 200 видео может предсказать текущую CU с использованием однонаправленного предсказания или двунаправленного предсказания.

[0054] JEM также обеспечивает режим аффинной компенсации движения, который может рассматриваться как режим интер-предсказания. В режиме аффинной компенсации движения, кодер 200 видео может определять два или более вектора движения, которые представляют непоступательное движение, такое как увеличение или уменьшение масштаба, вращение, движение в перспективе или другие типы нерегулярного движения.

[0055] Для выполнения интра-предсказания, кодер 200 видео может выбрать режим интра-предсказания для генерации блока предсказания. JEM обеспечивает шестьдесят семь режимов интра-предсказания, включая различные направленные режимы, а также планарный режим и режим DC. В общем случае, кодер 200 видео выбирает режим интра-предсказания, который описывает соседние выборки для текущего блока (например, блока CU), из которых следует предсказывать выборки текущего блока. Такие выборки обычно могут находиться сверху, сверху и слева или слева от текущего блока в том же изображении, где и текущий блок, в предположении, что кодер 200 видео кодирует STU и CU в порядке растрового сканирования (слева направо, сверху вниз).

[0056] Кодер 200 видео кодирует данные, представляющие режим предсказания для текущего блока. Например, для режимов интер-предсказания, кодер 200 видео может кодировать данные, представляющие, какой из различных доступных режимов интер-предсказания используется, а также информацию движения для соответствующего режима. Например, для однонаправленного или двунаправленного интер-предсказания, кодер 200 видео может кодировать векторы движения с использованием расширенного предсказания векторов движения (AMVP) или режима слияния. Кодер 200 видео может использовать аналогичные режимы для кодирования векторов движения для режима аффинной компенсации движения.

[0057] После предсказания, такого как интра-предсказание или интер-предсказание блока, кодер 200 видео может вычислить остаточные данные для блока. Остаточные данные, такие как остаточный блок, представляют разности от выборки к выборке между данным блоком и блоком предсказания для данного блока, сформированным с использованием соответствующего режима предсказания. Кодер 200 видео может применять одно или более преобразований к остаточному блоку для получения преобразованных данных в области преобразования вместо области выборок. Например, кодер 200 видео может применять дискретное косинусное преобразование (DCT), целочисленное преобразование, вейвлет-преобразование или концептуально аналогичное преобразование к остаточным данным видео. Кроме того, кодер 200 видео может применять вторичное преобразование после первого преобразования, такое как зависимое от режима неделимое вторичное преобразование (MDNSST), зависимое от сигнала преобразование, преобразование Карунена-Лоэва (KLT) и т.п. Кодер 200 видео

формирует коэффициенты преобразования после применения одного или более преобразований.

[0058] Как отмечалось выше, после любых преобразований для получения коэффициентов преобразования, кодер 200 видео может выполнять квантование коэффициентов преобразования. Квантование обычно относится к процессу, в котором коэффициенты преобразования квантуются, чтобы по возможности уменьшить объем данных, используемых для представления коэффициентов, обеспечивая дальнейшее сжатие. Путем выполнения процесса квантования, кодер 200 видео может уменьшить битовую глубину, ассоциированную с некоторыми или всеми из коэффициентов. Например, кодер 200 видео может округлять n -битовое значение до m -битового значения во время квантования, где n больше, чем m . В некоторых примерах, чтобы выполнить квантование, кодер 200 видео может выполнить побитовое смещение вправо значения, подлежащего квантованию.

[0059] После квантования, кодер 200 видео может сканировать коэффициенты преобразования, формируя одномерный вектор из двумерной матрицы, включающей квантованные коэффициенты преобразования. Сканирование может быть спроектировано так, чтобы помещать коэффициенты преобразования более высокой энергии (и, следовательно, более низкой частоты) в начале вектора и помещать коэффициенты преобразования более низкой энергии (и, следовательно, более высокой частоты) в конце вектора. В некоторых примерах, кодер 200 видео может использовать predetermined порядок сканирования для сканирования квантованных коэффициентов преобразования для получения преобразованного в последовательную форму вектора, а затем энтропийно кодировать квантованные коэффициенты преобразования вектора. В других примерах, кодер 200 видео может выполнять адаптивное сканирование. После сканирования квантованных коэффициентов преобразования для формирования одномерного вектора, кодер 200 видео может энтропийно кодировать одномерный вектор, например, в соответствии с контекстно-адаптивным двоичным арифметическим кодированием (САВАС). Кодер 200 видео может также энтропийно кодировать значения для синтаксических элементов, описывающих метаданные, ассоциированные с кодированными данными видео, для использования декодером 300 видео при декодировании данных видео.

[0060] Для выполнения САВАС, кодер 200 видео может назначать контекст в рамках контекстной модели символу, который должен быть передан. Контекст может относиться, например, к тому, являются ли соседние значения символа нулевыми или нет. Определение вероятности может основываться на контексте, назначенном символу.

[0061] Кодер 200 видео может дополнительно генерировать синтаксические данные, такие как синтаксические данные на блочной основе, синтаксические данные на основе изображения и синтаксические данные на основе последовательности, для декодера 300 видео, например, в заголовке изображения, заголовке блока, заголовке вырезки, или другие синтаксические данные, такие как набор параметров последовательности (SPS), набор параметров изображения (PPS) или набор параметров видео (VPS). Декодер 300 видео может также декодировать такие синтаксические данные, чтобы определить, как декодировать соответствующие данные видео.

[0062] Таким образом, кодер 200 видео может генерировать битовый поток, включающий в себя кодированные данные видео, например, синтаксические элементы, описывающие разбиение изображения на блоки (например, CU), и информацию предсказания и/или остаточную информацию для блоков. В конечном счете, декодер 300 видео может принимать битовый поток и декодировать закодированные данные

видео.

[0063] В принципе, декодер 300 видео выполняет процесс, обратный процессу, выполняемому кодером 200 видео, для декодирования закодированных данных видео битового потока. Например, декодер 300 видео может декодировать значения для синтаксических элементов битового потока с использованием САВАС, по существу аналогично, хотя и обратным образом, по отношению к процессу кодирования САВАС кодера 200 видео. Синтаксические элементы могут определять информацию разбиения изображения на СТU и разбиения каждой СТU согласно соответствующей структуре разбиения, такой как структура QTBT, для определения CU в СТU. Синтаксические элементы могут дополнительно определять предсказание и остаточную информацию для блоков (например, CU) данных видео.

[0064] Остаточная информация может быть представлена, например, квантованными коэффициентами преобразования. Декодер 300 видео может обратно квантовать и обратно преобразовывать квантованные коэффициенты преобразования блока для восстановления остаточного блока для данного блока. Декодер 300 видео использует сигнализированный режим предсказания (интра- или интер-предсказания) и связанную информацию предсказания (например, информацию движения для интер-предсказания) для формирования блока предсказания для данного блока. Декодер 300 видео может затем объединять блок предсказания и остаточный блок (на основе от выборки к выборке) для восстановления исходного блока. Декодер 300 видео может выполнять дополнительную обработку, такую как выполнение процесса устранения блочности для уменьшения визуальных артефактов вдоль границ блока.

[0065] Настоящее раскрытие может, в общем, ссылаться на “сигнализацию” определенной информации, такой как синтаксические элементы. Термин “сигнализация” может, в общем, относиться к сообщению значений для синтаксических элементов и/или других данных, используемых для декодирования закодированных данных видео. То есть, кодер 200 видео может сигнализировать значения для синтаксических элементов в битовом потоке. В общем, сигнализация относится к генерации значения в битовом потоке. Как отмечалось выше, устройство 102 источника может транспортировать битовый поток к устройству 116 места назначения по существу в реальном времени или не в реальном времени, как может происходить при сохранении синтаксических элементов в устройстве 112 хранения для последующего извлечения устройством 116 места назначения.

[0066] В соответствии с методами настоящего раскрытия, кодер 200 видео и декодер 300 видео могут быть сконфигурированы для выполнения параллельной обработки волнового фронта при кодировании изображения данных видео. В общем, параллельная обработка волнового фронта может включать кодирование отдельных строк единиц дерева кодирования (СТU) с использованием отдельных подпроцессов обработки. Например, первый подпроцесс, исполняемый кодером 200 видео или декодером 300 видео, может обрабатывать первую строку СТU, второй подпроцесс может обрабатывать вторую строку СТU и т.д. Кодирование СТU включает в себя, среди прочего, кодирование информации движения для единиц кодирования (CU) с предсказанием движения в СТU, которые могут ссылаться на информацию движения в пределах той же самой СТU или ранее закодированной СТU (например, левой и/или верхней соседней СТU). Такая информация движения может сохраняться в буфере предсказателя вектора движения (MVP). В соответствии с методами настоящего раскрытия, кодер 200 видео и декодер 300 видео могут быть сконфигурированы, чтобы сбрасывать буфер MVP для текущей строки СТU перед кодированием данных видео текущей строки СТU. Буфер

MVP может быть отдельным буфером MVP для текущей строки STU, или общий буфер MVP может использоваться для множества строк STU.

5 [0067] В некоторых примерах, при сохранении информации движения в буфере MVP, кодер 200 видео и декодер 300 видео могут сохранять только уникальную информацию движения в буфере MVP. Например, кодер 200 видео и декодер 300 видео могут кодировать текущую СУ с использованием текущего вектора движения, определять, сохранен ли в текущее время вектор движения в буфере MVP для текущей СУ, и если да, запрещать сохранение вектора движения в буфере MVP, а если нет, то сохранять вектор движения в буфере MVP.

10 [0068] В некоторых примерах, кодер 200 видео и декодер 300 видео могут использовать правило “первым пришел - первым обслужен” (FIFO) для удаления векторов движения из буфера MVP, когда буфер MVP становится заполненным. То есть для того, чтобы добавить новый вектор движения в буфер MVP, кодер 200 видео и декодер 300 видео могут удалить самый ранний введенный вектор движения из буфера MVP и ввести новый вектор движения в буфер MVP. Таким образом, буфер MVP может реализовывать режим, подобный очереди.

15 [0069] В некоторых примерах, кодер 200 видео и декодер 300 видео могут поддерживать отдельные буферы MVP для каждого из множества типов различных моделей движения. Например, кодер 200 видео и декодер 300 видео могут поддерживать буфер аффинного MVP для аффинной модели движения, буфер MVP внутриблочного копирования для информации движения режима внутриблочного копирования, буфер MVP компенсации освещения для информации движения компенсации локального освещения, буфер подблочного MVP для подблочного MVP и/или буфер временного MVP для временного предсказания движения.

25 [0070] В некоторых примерах, кодер 200 видео и декодер 300 видео могут генерировать синтетический MVP из двух или более MVP в одном или более буферах MVP и вводить синтетический MVP в один из буферов MVP. Два или более MVP могут соответствовать одной и той же или различным моделям движения (то есть иметь разные типы информации движения).

30 [0071] Фиг. 2А и 2В представляют собой концептуальную диаграмму, иллюстрирующую примерную структуру 130 квадродерева-двоичного дерева (QTBT) и соответствующую единицу дерева кодирования (STU) 132. Сплошные линии представляют разбиение квадродерева, а пунктирные линии указывают разбиение двоичного дерева. В каждом разделенном (т.е. нелистовом) узле двоичного дерева 35 сигнализируется один флаг, указывающий, какой тип разделения (т.е. горизонтальное или вертикальное) используется, где 0 указывает горизонтальное разделение, а 1 указывает вертикальное разделение в данном примере. Для разделения квадродерева, нет необходимости указывать тип разделения, так как узлы квадродерева разделяют блок по горизонтали и вертикали на 4 подблока одинакового размера. Соответственно, 40 кодер 200 видео может кодировать, а декодер 300 видео может декодировать синтаксические элементы (такие как информация разделения) для уровня дерева области структуры QTBT 130 (т.е. сплошные линии) и синтаксические элементы (такие как информация разделения) для уровня дерева предсказания структуры QTBT 130 (т.е. пунктирные линии). Кодер 200 видео может кодировать, а декодер 300 видео может декодировать данные видео, такие как данные предсказания и преобразования, для 45 СУ, представленных конечными листовыми узлами структуры QTBT 130.

[0072] В общем, STU 132 на фиг. 2В может ассоциироваться с параметрами, определяющими размеры блоков, соответствующих узлам структуры QTBT 130 на

первом и втором уровнях (например, уровне дерева области и уровне дерева предсказания). Эти параметры могут включать в себя размер STU (представляющий размер STU 132 в выборках), минимальный размер квадродерева (MinQTSize, представляющий минимально допустимый размер листового узла квадродерева),

5 максимальный размер двоичного дерева (MaxBTSize, представляющий максимально допустимый размер корневого узла двоичного дерева), максимальную глубину двоичного дерева (MaxBTDepth, представляющий максимально допустимую глубину двоичного дерева) и минимальный размер двоичного дерева (MinBTSize, представляющий минимально допустимый размер листового узла двоичного дерева).

10 [0073] Корневой узел структуры QTBT, соответствующий STU, может иметь четыре дочерних узла на первом уровне структуры QTBT, каждый из которых может разбиваться в соответствии с разбиением квадродерева. То есть узлы первого уровня являются либо листовыми узлами (не имеющими дочерних узлов), либо имеют четыре дочерних узла. Пример структуры QTBT 130 представляет такие узлы, как включающие

15 родительский узел и дочерние узлы, имеющие сплошные линии для ветвей. Если узлы первого уровня не больше, чем максимально допустимый размер корневого узла двоичного дерева (MaxBTSize), то узлы могут дополнительно разбиваться на соответствующие двоичные деревья. Разделение двоичного дерева одного узла может повторяться до тех пор, пока узлы, полученные в результате разделения, не достигнут

20 минимально допустимого размера листового узла двоичного дерева (MinBTSize) или максимально допустимой глубины двоичного дерева (MaxBTDepth). Пример структуры QTBT 130 представляет такие узлы как имеющие пунктирные линии для ветвей. Листовой узел двоичного дерева упоминается как единица кодирования (CU), которая используется для предсказания (например, предсказания внутри изображения или между

25 изображениями) и преобразования, без какого-либо дальнейшего разбиения. Как обсуждалось выше, CU также могут упоминаться как “блоки видео” или “блоки”.

[0074] В одном примере структуры разбиения QTBT, размер STU устанавливается как 128×128 (выборки яркости и две соответствующие 64×64 выборки цветности), MinQTSize устанавливается как 16×16, MaxBTSize устанавливается как 64×64, MinBTSize

30 (как для ширины, так и для высоты) устанавливается как 4, и MaxBTDepth устанавливается как 4. Разбиение квадродерева применяется к STU сначала, чтобы сформировать листовые узлы квадродерева. Листовые узлы квадродерева могут иметь размер от 16×16 (т.е. MinQTSize) до 128×128 (т.е. размер STU). Если листовой узел квадродерева равен 128×128, он не будет далее разделен на двоичное дерево, так как

35 размер превышает MaxBTSize (т.е. 64×64, в данном примере). В противном случае, листовой узел квадродерева будет дополнительно разбиваться на двоичное дерево. Таким образом, листовой узел квадродерева также является корневым узлом для двоичного дерева и имеет глубину двоичного дерева, равную 0. Когда глубина двоичного дерева достигает MaxBTDepth (4, в данном примере), дальнейшее разделение не

40 допускается. Когда узел двоичного дерева имеет ширину, равную MinBTSize (4, в данном примере), это означает, что дальнейшее вертикальное разделение не допускается. Аналогично, узел двоичного дерева, имеющий высоту, равную MinBTSize, подразумевает, что дальнейшее горизонтальное разделение для этого узла двоичного дерева не допускается. Как отмечено выше, листовые узлы двоичного дерева

45 упоминаются как CU и далее обрабатываются в соответствии с предсказанием и преобразованием без дальнейшего разбиения.

[0075] Фиг. 3 представляет собой блок-схему последовательности операций, иллюстрирующую примерный процесс для кодирования информации движения с

использованием предсказателей векторов движения истории (HMVP). Первоначально устройство кодирования видео, такое как кодер 200 видео или декодер 300 видео, загружает таблицу с кандидатами HMVP (140). Затем устройство кодирования видео кодирует блок данных видео с использованием кандидатов HMVP (142). Затем
 5 устройство кодирования видео обновляет таблицу информацией движения кодированного блока (144).

[0076] Фиг. 4 представляет собой концептуальную диаграмму, иллюстрирующую пример обновления таблицы HMVP. В JVET-K0104, размер таблицы установлен равным 16, и применяется правило “первым пришел - первым обслужен” (FIFO). Фиг. 4
 10 изображает пример, в котором правило FIFO применяется для удаления кандидата HMVP и добавления нового в таблицу, используемую в примерах методов настоящего раскрытия.

[0077] Устройство кодирования видео, такое как кодер 200 видео или декодер 300 видео, может вводить кандидатов HMVP из последней записи в первую запись в таблице
 15 после кандидата временного предсказания вектора движения (TMVP) в список кандидатов. Устройство кодирования видео может применить обрезку (исключение) к кандидатам HMVP. Устройство кодирования видео может прекратить процесс обрезки, когда общее число доступных кандидатов слияния достигнет сигнализированного максимально допустимого числа кандидатов слияния.

[0078] В примере на фиг. 4, таблица перед обновлением содержит от MVP_0 истории (HMVP₀) до MVP_{L-1} истории (HMVP_{L-1}), где нижние индексы от 0 до L-1 представляют порядок, в котором добавляются MVP истории. C_{L-1} представляет новый MVP истории, подлежащий добавлению в таблицу. Таким образом, согласно правилу FIFO, HMVP₀
 20 удаляется из таблицы перед добавлением C_{L-1} , в данном примере.

[0079] Фиг. 5 представляет собой концептуальную диаграмму, иллюстрирующую примерный выбор несмежных блоков для кодирования информации движения. В примере на фиг. 5, текущий блок, обозначенный “Curr”, представляет текущую единицу кодирования (CU), для которой информация движения может быть закодирована с
 30 использованием смежных и/или несмежных соседних блоков, обозначенных A_i , B_j и N_{A_k} . Предсказание несмежного вектора движения описано, например, в заявке США № 16/003,269, поданной 8 июня 2018. Устройство кодирования видео может применять правило FIFO и максимальный размер буфера кандидатов движения для несмежных блоков.

[0080] Фиг. 6 представляет собой концептуальную диаграмму, иллюстрирующую примерный выбор несмежных блоков на основе родительского блока. То есть, родительский блок представляет собой блок, разбитый на подблоки, включая текущий блок. Например, родительский блок может представлять собой STU или подблок, на
 40 который была разделена STU. Подобно фиг. 5, на фиг. 6 текущая CU обозначена как “Curr”, и несмежные блоки, из которых информация движения может извлекаться и использоваться для предсказания информации движения текущей CU, обозначены как “ $NA_{i,j}$ ”.

[0081] Векторы движения смежных пространственных блоков совмещенного блока
 45 могут быть использованы в качестве кандидатов предсказания вектора движения (MVP) для режима слияния, в дополнение к векторам движения Н и С (то есть векторам движения в центре и снизу-справа от совмещенного блока).

[0082] Методы настоящего раскрытия могут быть использованы для улучшения

предсказания вектора движения, например, путем добавления кандидатов, используемых для режимов AMVP и/или кодирования со слиянием, где добавленные кандидаты могут быть взяты из несмежных блоков. Например, добавленные кандидаты могут соответствовать любым из $NA_{1,1}$ до $NA_{1,9}$ на фиг. 6.

5 [0083] Фиг. 7 представляет собой концептуальную диаграмму, иллюстрирующую пример желательной обработки волнового фронта единиц дерева кодирования (СТУ). Как показано на фиг. 7, для обработки различных строк СТУ, могут быть назначены различные подпроцессы. То есть устройство кодирования видео, такое как кодер 200 видео или декодер 300 видео, может исполнять множество различных подпроцессов, 10 например, для параллельной обработки волнового фронта (WPP) при кодировании различных строк СТУ. В некоторых примерах, определенные вероятности, используемые для контекстного кодирования (например, кодирования САВАС) информации движения интер-предсказанных блоков, не могли бы определяться, если бы вероятности надо было определять из последнего блока предыдущей строки СТУ, например, в 15 предположении, что последний блок еще не был кодирован. Таким образом, в соответствии с методами настоящего раскрытия, кодер 200 видео и декодер 300 видео могут сбросить буфер СТУ для строки СТУ перед кодированием строки СТУ, чтобы гарантировать, что строки СТУ могут обрабатываться корректным образом.

[0084] Фиг. 8 представляет собой концептуальную диаграмму, иллюстрирующую 20 пример информации движения, используемой для HMVP. Фиг. 8 иллюстрирует, как использование FIFO может удалять векторы движения блоков, более близких к текущему блоку, из списка кандидатов, в то время как векторы движения последующих блоков могут приниматься во внимание. В частности, на фиг. 8, X представляет информацию движения, кодируемую в текущее время, и MV заштрихованных блоков находятся в 25 буфере истории. Настоящее раскрытие признает, что традиционные методы HMVP не полностью используют векторы движения несмежных блоков, по меньшей мере частично, ввиду использования правила FIFO, как показано на фиг. 8.

[0085] В частности, при кодировании блока X, информация движения несмежных 30 блоков (TL0, T0, T1, TR0, TR1, TR2, TR3) верхней-левой СТУ, верхней СТУ и верхней-правой СТУ была удалена из буфера истории. Поэтому информация движения этих блоков не рассматривается для добавления в список кандидатов, даже если несмежные блоки находятся ближе к X, чем, например, СТУ LL0, СТУ LL1 и СТУ F0-F3, векторы движения которых находятся в буфере истории.

[0086] Настоящее раскрытие также признает, что один буфер для HMVP не применим 35 для параллельной обработки волнового фронта. Если используется только один буфер, размер буфера должен быть очень большим для вмещения потенциальных пространственных кандидатов для блоков, обрабатываемых в каждом подпроцессе (например, строк СТУ). Например, если четыре подпроцесса настроены, чтобы 40 исполняться параллельно, размер буфера может достигать 64. В результате, потребуется больше битов, чтобы сигнализировать индекс MVP декодеру 300 видео. Аналогичным образом, могут возникать избыточные записи. То есть, запись в буфере истории может быть потенциально полезной для блока в этой строке, но может быть бесполезной для 45 блоков в других строках (например, X и F на фиг. 8). Следовательно, может быть трудным найти оптимального кандидата для блока.

[0087] Фиг. 9 представляет собой концептуальную диаграмму, иллюстрирующую 50 пример изображения, разбитой на множество строк единиц дерева кодирования (СТУ). В частности, в примере фиг. 9, изображение 150 включает в себя строки СТУ 152A-152E (строки СТУ 152). Каждая из строк СТУ 152 включает в себя соответствующий набор

СТУ: строка СТУ 152А включает в себя СТУ 154А-154J, строка СТУ 152В включает в себя СТУ 156А-156J, строка СТУ 152С включает в себя СТУ 158А-158J, строка СТУ 152D включает в себя СТУ 160А-160J, и строка СТУ 152Е включает в себя СТУ 162А-162J.

5 [0088] Кодер 200 видео и декодер 300 видео могут быть сконфигурированы в соответствии с методами настоящего раскрытия для использования множества буферов для основанного на истории MVP. В некоторых примерах, кодер 200 видео и декодер 300 видео могут поддерживать отдельные буферы MVP истории для каждой из строк СТУ 152 (каждая из которых может обрабатываться отдельным соответствующим
10 подпроцессом обработки), или может иметься единственный буфер, который сбрасывается в начале каждой строки СТУ при применении параллельной обработки волнового фронта.

[0089] В одном примере, СТУ 158С может представлять собой текущую СТУ. Информация движения СТУ 154F-154F, 156А-156D, 158А и 158В (показана с
15 использованием серого затенения на фиг. 9) может быть доступна в одном или более соответствующих буферах MVP истории для использования при кодировании информации движения СТУ 158С.

[0090] Дополнительно или альтернативно, кодер 200 видео и декодер 300 видео могут выполнять инициализацию буфера MVP истории с использованием любого или всех из
20 следующих методов, отдельно или в комбинации. Кодер 200 видео и декодер 300 видео могут сбросить буфер MVP истории каждой строки СТУ для очистки. Кодер 200 видео и декодер 300 видео могут предварительно заполнять буфер MVP истории каждой строки СТУ нулевыми векторами движения с различным индексом опорного кадра и/или направлениями интер-предсказания или другой предопределенной или выведенной
25 информации движения. Кодер 200 видео и декодер 300 видео могут предварительно заполнить буфер MVP истории каждой строки СТУ информацией движения из кодированных кадров (картинок) в том же временном слое или нижних временных слоях (опорных картинок, доступных для текущего кадра/изображения).

[0091] Кодер 200 видео и декодер 300 видео могут масштабировать информацию
30 движения, например, на основе временного расстояния, или обрабатывать/модифицировать информацию движения, например, комбинировать эту информацию движения с другим MV. В общем, кодер 200 видео и декодер 300 видео могут комбинировать информацию движения с информацией движения из буфера MVP предыдущей истории в кодированных кадрах/изображениях или информацией движения
35 из совмещенной области (может быть СТУ или больше, чем размер определенного блока, например, блока 4×4) в кодированных кадрах/изображениях. Кодер 200 видео и декодер 300 видео могут предварительно заполнить буфер MVP истории верхней строки СТУ, когда кодируется верхняя-правая СТУ от текущей СТУ. Кодер 200 видео и декодер 300 видео могут использовать нулевые векторы движения с различными
40 индексами опорных кадров и/или направлениями интер-предсказания, или другую предопределенную или выведенную информацию движения.

[0092] Дополнительно или альтернативно, кодер 200 видео и декодер 300 видео могут, когда кодируется (кодируется или декодируется) некоторая СТУ из строк СТУ, использовать ассоциированный буфер MVP истории для инициализации или модификации
45 буферов истории строки СТУ ниже текущей строки СТУ.

[0093] Дополнительно или альтернативно, кодер 200 видео и декодер 300 видео могут применять правило FIFO для удаления записи из ассоциированного буфера MVP истории.

[0094] Дополнительно или альтернативно, кодер 200 видео и декодер 300 видео могут

очистить буфер MVP истории, когда ассоциированная строка STU полностью закодирована/декодирована.

[0095] Кодер 200 видео и декодер 300 видео могут поддерживать размер буфера MVP, который больше, чем списки кандидатов AMVP/слияния или других режимов. Любой MV или несколько MV из буфера могут выбираться в качестве кандидата(ов) MV для списка кандидатов, используемого в определенном режиме, например AMPV, режиме слияния, аффинном или любом другом интер-режиме. Правило может определять, как выбрать MV из буфера, например, взять N последних MV, добавленных в буфер, или взять несколько от начала буфера и/или несколько от середины буфера и/или несколько от конца буфера. Альтернативно, сигнализация может применяться для указания того, какие MV выбираются (например, кодер 200 видео может кодировать сигнализируемые данные, и декодер 300 видео может декодировать сигнализируемые данные). Размер буфера MVP может сигнализироваться в любом наборе параметров (например, наборе параметров видео, наборе параметров последовательности, наборе параметров изображения и т.п.), заголовке вырезки или еще где-либо. Буфер MVP может быть ассоциирован с вырезкой, изображением и/или последовательностью видео.

[0096] Когда кодер 200 видео и декодер 300 видео обрабатывают интер-кодированный блок, MV, использованные в блоке, могут добавляться в буфер MVP, и только уникальные MV могут сохраняться в буфере. Когда буфер заполнен, более старые MV могут быть удалены из буфера, когда добавляются новые MV. Может иметься правило, согласно которому MV могут добавляться в буфер, например, могут добавляться только сигнализированные MV, как в режиме AMVP, и если блок кодирован в режиме слияния, то MV блока не добавляется в буфер.

[0097] Кодер 200 видео и декодер 300 видео могут добавлять MV к уже существующему MV или нескольким MV в буфере. Например, если уже существующие MV в буфере являются однонаправленными, то при добавлении нового MV эти существующие MV могут быть модифицированы как двунаправленные путем добавления нового MV.

[0098] Некоторая обработка MV может быть применена при добавлении нового MV. Например, если новый MV близок к уже существующим MV в буфере, то эти близкие MV могут быть удалены. Близкие может означать близость при сравнении значений компонентов MV (например, x- и y-компонентов). В некоторых примерах, в буфер могут быть добавлены только MV, которые отличаются от уже существующих MV в буфере на пороговое значение. То же самое пороговое значение может быть сконфигурировано для разных буферов.

[0099] Векторы движения в буфере могут быть однонаправленными (L0 или L1), двунаправленными или представлять собой MV любой другой модели движения.

[0100] Информация о режиме может быть ассоциирована с MV в буфере, и если индекс MV в буфере сигнализируется в блоке или применяется другое правило о получении MV из буфера, то информация о режиме может быть получена из данных, ассоциированных с этой информацией MV. Например, если этой информацией является режим слияния, то блок кодируется в режиме слияния с указанным MV.

[0101] Настоящее раскрытие далее признает, что традиционные, основанные на истории MVP содержат только регулярные предсказатели движения и используются только для регулярного предсказания движения без модифицирования информации движения. В соответствии с методами настоящего раскрытия, кодер 200 видео и декодер 300 видео могут использовать по меньшей мере один буфер MVP истории, который содержит не только закодированную информацию движения, но и другие типы

предсказателей движения, таких как для аффинной модели движения, информацию движения режима внутриблочного копирования, информацию движения компенсации локального освещения, подблочный MVP или временной предсказатель движения.

5 [0102] Дополнительно или альтернативно, кодер 200 видео и декодер 300 видео могут использовать множество буферов MVP истории для различных моделей движения, таких как аффинная модель движения, информация движения режима внутриблочного копирования, информация движения компенсации локального освещения, подблочный MVP или временной предсказатель движения.

10 [0103] Дополнительно или альтернативно, синтетический вектор движения, основанный на текущем MVP и другом предсказателе движения, таком как другой пространственный MVP или временной MVP, также может быть добавлен в список кандидатов.

15 [0104] Дополнительно или альтернативно, кодер 200 видео и декодер 300 видео могут генерировать синтетический MVP из двух или более MVP в буфере MVP истории или одного или более MVP в буфере MVP истории с другими типами MVP, такими, как пространственные или временные MVP.

20 [0105] Кодер 200 видео и декодер 300 видео могут реализовывать схему блочного разбиения. В HEVC, изображения делятся на последовательность единиц дерева кодирования (CTU). Для изображения, которая имеет три массива выборок, CTU включает в себя блок $N \times N$ выборок яркости вместе с двумя соответствующими блоками выборок цветности. CTU разделяется на единицы кодирования (CU) с использованием древовидной структуры. Каждая листовая CU может дополнительно разделяться на одну, две или четыре единицы предсказания (PU) в соответствии с типом разделения PU. После получения остаточного блока, путем применения процесса предсказания, 25 основанного на типе разделения PU, листовая CU может быть разделена на единицы преобразования (TU).

[0106] В VVC, структура сегментации квадродерева с вложенным деревом множественного типа, использующим двоичные и троичные разделения, заменяет концепции множественных типов единиц разбиения, т.е. разбиение вложенного дерева 30 множественного типа удаляет концепции разделения CU, PU и TU, за исключением необходимых для CU, которые имеют слишком большой размер для максимальной длины преобразования и поддерживают большую гибкость для форм разделения CU. В структуре дерева кодирования, CU может иметь квадратную или прямоугольную форму.

35 [0107] Кодер 200 видео и декодер 300 видео могут использовать информацию движения для предсказания блока данных видео. Для каждого блока, может быть доступен набор информации движения. Набор информации движения содержит информацию движения для прямого и обратного направлений предсказания. Здесь прямое и обратное направления предсказания представляют собой два направления 40 предсказания, соответствующие списку опорных картинок 0 (RefPicList0) и списку опорных картинок 1 (RefPicList1) текущей изображения или вырезки. Термины “прямое” и “обратное” не обязательно имеют геометрическое значение. Вместо этого, они используются, чтобы различать, на каком опорном изображении основан вектор движения. Прямое предсказание означает предсказание, сформированное на основе опорного списка 0, в то время как обратное предсказание означает предсказание, сформированное на основе опорного списка 1. В тех случаях, когда как опорный список 0, так и опорный список 1 используются для формирования предсказания для данного блока, это называется двунаправленным предсказанием.

[0108] Для данной изображения или вырезки, если используется только один список опорных картинок, каждый блок внутри изображения или вырезки предсказывается в прямом направлении. Если оба списка опорных картинок используются для данной изображения или вырезки, блок внутри изображения или вырезки может быть

5 предсказываемым в прямом направлении или предсказываемым в обратном направлении или двунаправленно предсказываемым.

[0109] Для каждого направления предсказания, информация движения содержит опорный индекс и вектор движения. Опорный индекс используется для идентификации опорной изображения в соответствующем списке опорных картинок (например,

10 RefPicList0 или RefPicList1). Вектор движения имеет как горизонтальную, так и вертикальную компоненту, причем каждая из них указывает значение смещения вдоль горизонтального и вертикального направлений соответственно. В некоторых описаниях, для простоты, слово “вектор движения” может использоваться взаимозаменяемо с информацией движения, чтобы указывать как вектор движения, так и ассоциированный

15 с ним опорный индекс.

[0110] Отсчет порядка картинок (POC) широко используется в стандартах кодирования видео для идентификации порядка отображения изображения. Хотя бывают случаи, когда два изображения в одной кодированной последовательности видео могут иметь одинаковое значение POC, обычно это не происходит в пределах

20 кодированной последовательности видео. Когда в битовом потоке присутствует несколько кодированных последовательностей видео, изображения с одинаковым значением POC могут быть ближе друг к другу с точки зрения порядка декодирования.

[0111] В HEVC, имеются два режима интер-предсказания, называемые режимом слияния (причем пропуск считается специальным случаем режима слияния) и режимом

25 расширенного предсказания вектора движения (AMVP), соответственно для PU.

[0112] В режиме AMVP или слияния, список кандидатов вектора движения (MV) поддерживается для множества предсказателей векторов движения. Вектор(ы) движения, как и опорные индексы в режиме слияния, текущей PU генерируются взятием одного кандидата из списка кандидатов MV.

[0113] Список кандидатов MV содержит до 5 кандидатов для режима слияния и только два кандидата для режима AMVP. Кандидат слияния может содержать набор информации движения, например, векторов движения, соответствующих как спискам опорных картинок (список 0 и список 1), так и опорным индексам. Если кандидат слияния идентифицируется индексом слияния, то опорные изображения используются

35 для предсказания текущих блоков, а также определяются ассоциированные векторы движения. Однако в режиме AMVP для каждого потенциального направления предсказания из списка 0 или списка 1 опорный индекс явно сигнализируется, вместе с индексом MVP в список кандидатов MV, поскольку кандидат AMVP содержит только вектор движения. В режиме AMVP, предсказанные векторы движения могут

40 дополнительно уточняться.

[0114] Как можно видеть из описанного выше, кандидат слияния соответствует полному набору информации движения, в то время как кандидат AMVP содержит только один вектор движения для конкретного направления предсказания и опорного индекса. Кандидаты для обоих режимов могут быть выведены аналогично из одних и

45 тех же пространственных и временных соседних блоков.

[0115] Фиг. 10А и 10В представляют собой блочные диаграммы, иллюстрирующие примерные пространственные соседние кандидаты векторов движения для режимов слияния и расширенного предсказания вектора движения (AMVP). Фиг. 10А показывает

пример пространственных соседних кандидатов MV для режима слияния, а фиг. 10B показывает пример пространственных соседних кандидатов MV режима AMVP. Пространственные кандидаты MV выводятся из соседних блоков, как показано на фиг. 10A и 10B. Для конкретного PU (PU0), способы для генерации кандидатов из блоков различаются для режимов слияния и AMVP.

[0116] В режиме слияния, можно вывести до четырех пространственных кандидатов MV в порядке, показанном на фиг. 10A. В частности, порядок является следующим: слева (0), сверху (1), сверху справа (2), снизу слева (3) и сверху слева (4), как показано на фиг. 10A.

[0117] В режиме AVMP, соседние блоки делятся на две группы. Первой группой является левая группа, которая включает в себя блоки 0 и 1. Второй группой является верхняя группа, которая включает в себя блоки 2, 3 и 4, как показано на фиг. 10B. Для каждой группы, потенциальный кандидат в соседнем блоке, ссылающийся на то же самое опорное изображение, что и указанная сигнализированным опорным индексом, имеет наивысший приоритет, подлежащий выбору для формирования окончательного кандидата группы. Возможно, что все соседние блоки не содержат вектор движения, указывающий на одно и то же опорное изображение. Поэтому, если такой кандидат не может быть найден, первый доступный кандидат будет масштабироваться для формирования окончательного кандидата, таким образом, различия во временном расстоянии могут быть скомпенсированы.

[0118] Фиг. 11A и 11B представляют собой концептуальные диаграммы, иллюстрирующие кандидатов временного предсказания вектора движения (TMPV). Фиг. 11A показывает пример кандидата TMPV. Кандидат TMVP, если он включен и доступен, добавляется в список кандидатов MV после кандидатов пространственного вектора движения. Процесс вывода вектора движения для кандидата TMPV является тем же самым для режимов слияния и AMPV, однако целевой опорный индекс для кандидата TMPV в режиме слияния всегда установлен в 0.

[0119] Первичным местоположением блока для вывода кандидата TMVP является нижний правый блок вне совмещенного PU, как показано на фиг. 11A в виде блока "T", чтобы компенсировать смещение к верхнему и левому блокам, используемым для генерации пространственных соседних кандидатов. Однако, если этот блок расположен вне текущей строки СТВ или информация движения недоступна, блок заменяется центральным блоком PU.

[0120] Вектор движения для кандидата TMVP выводится из совмещенной PU совмещенной изображения, указанной на уровне вырезки. Вектор движения для совмещенной PU называется совмещенным MV.

[0121] Фиг. 11B показывает пример масштабирования MV. Чтобы вывести вектор движения кандидата TMVP, совмещенный MV может потребовать масштабирования, чтобы скомпенсировать различия временных расстояний, как показано на фиг. 11B.

[0122] Некоторые другие аспекты режимов слияния и AMVP заслуживают упоминания следующим образом. Например, кодер 200 видео и декодер 300 видео могут выполнять масштабирование вектора движения. Предполагается, что значение векторов движения пропорционально расстоянию между изображениями во время представления. Вектор движения ассоциирует два изображения, опорное изображение и изображение, содержащее вектор движения (а именно, вмещающее изображение). Когда вектор движения используется для предсказания другого вектора движения, расстояние между вмещающим изображением и опорным изображением вычисляется на основе значений отсчета порядка изображения (POC).

[0123] Для предсказания вектора движения, его ассоциированное вмещающее изображение и опорное изображение могут быть разными. Поэтому вычисляется новое расстояние (на основе РОС). И вектор движения масштабируется на основе этих двух расстояний РОС. Для пространственного соседнего кандидата, вмещающие изображения для двух векторов движения одинаковы, в то время как опорные изображения различны. В HEVC, масштабирование вектора движения применяется как к TMVP, так и к AMVP для пространственных и временных соседних кандидатов.

[0124] В качестве другого примера, кодер 200 видео и декодер 300 видео могут выполнять генерацию искусственных кандидатов вектора движения. Если список кандидатов вектора движения не является полным, искусственные кандидаты вектора движения генерируются и вставляются в конец списка до тех пор, пока в нем не будут содержаться все кандидаты.

[0125] В режиме слияния, существует два типа искусственных кандидатов MV: комбинированный кандидат, выведенный только для B-вырезов, и нулевые кандидаты, используемые только для AMVP, если первый тип не обеспечивает достаточно искусственных кандидатов. Для каждой пары кандидатов, которые уже находятся в списке кандидатов и имеют необходимую информацию движения, двунаправленные комбинированные кандидаты вектора движения выводятся комбинацией вектора движения первого кандидата, ссылающегося на изображение в списке 0, и вектора движения второго кандидата, ссылающегося на изображение в списке 1.

[0126] В качестве другого примера, кодер 200 видео и декодер 300 видео могут выполнять процесс обрезки для вставки кандидата. Кандидаты из разных блоков могут оказаться одинаковыми, что снижает эффективность списка кандидатов слияния/AMVP. Для решения этой проблемы применяется процесс обрезки. Он сравнивает одного кандидата с другими в текущем списке кандидатов, чтобы избежать вставки идентичного кандидата в определенной степени. Чтобы уменьшить сложность, применяется лишь ограниченное количество процессов обрезки вместо сравнения каждого потенциального кандидата со всеми другими существующими кандидатами. Если применимо, применяются только следующие сравнения: верхний кандидат слияния сравнивается с левым кандидатом слияния, верхний правый кандидат слияния сравнивается с верхним кандидатом слияния, нижний левый кандидат слияния сравнивается с левым кандидатом слияния, верхний левый кандидат слияния сравнивается с левым кандидатом слияния и верхним кандидатом слияния.

[0127] Кодер 200 видео и декодер 300 видео также могут использовать другие способы предсказания движения. В развитии многоцелевого кодирования видео (VVC), основанный на истории способ предсказания вектора движения (HMVP) был предложен в L. Zhang etc., "CE4-related: History-based Motion Vector Prediction", Joint Video Experts Team Document: JVET-K0104 (далее "K0104"). Способ HMVP позволяет каждому блоку найти свой предсказатель MV из списка MV, декодированных из предыдущего, в дополнение к непосредственно смежным причинно-обусловленным соседним полям движения. Таблица с множеством кандидатов HMVP поддерживается во время процесса кодирования/декодирования. Таблица опустошается, когда обнаруживается новая вырезка. Всякий раз, когда имеется интер-кодированный блок, ассоциированная информация движения вставляется в таблицу способом "первым поступил - первым обслужен" (FIFO) как новый кандидат HMVP. Затем может применяться ограниченное правило FIFO. При вставке HMVP в таблицу, сначала может применяться проверка избыточности, чтобы найти, имеется ли в таблице идентичный HMVP. Если он найден, то этот конкретный HMVP может быть удален из таблицы, и все кандидаты HMVP

затем перемещаются.

[0128] Кандидаты HMVP могут быть использованы в процессе построения списка кандидатов слияния. Например, все кандидаты HMVP от последней записи до первой записи в таблице могут быть вставлены после кандидата TMVP. Обрезка может быть применена к кандидатам HMVP. Как только общее число доступных кандидатов слияния достигает сигнализированного максимально допустимого числа кандидатов слияния, процесс построения списка кандидатов слияния завершается.

[0129] Аналогично, кандидаты HMVP также могут быть использованы в процессе построения списка кандидатов AMVP. Векторы движения последних K кандидатов HMVP в таблице могут быть вставлены после кандидата TMVP. В некоторых примерах, для построения списка кандидатов AMVP используются только кандидаты HMVP с тем же опорным изображением, что и целевое опорное изображение AMVP. Обрезка может быть применена к кандидатам HMVP.

[0130] В HEVC, кодирование текущей STU может зависеть только от STU слева, сверху-слева, сверху и сверху-справа. Таким образом, параллельная обработка волнового фронта (WPP) может поддерживаться в HEVC. Однако способ HMVP в K0104 может вызвать зависимость между текущим блоком и всеми ранее кодированными STU в вырезке. Таким образом, WPP не может быть применена, если используется способ HMVP. Настоящее раскрытие описывает методы для использования HMVP с инициализацией STU, в которых зависимости остаются такими же, как и в HEVC. Настоящее раскрытие также описывает методы HMVP с инициализацией строки STU (сброс).

[0131] Согласно методам настоящего раскрытия, кодер 200 видео и декодер 300 видео могут выполнять HMVP с инициализацией STU. Таблица HMVP инициализируется в начале каждой STU. Инициализация может добавить MV из непосредственно соседних кодированных блоков текущей STU в таблицу HMVP. Непосредственно соседний кодированный блок может находиться слева, сверху, сверху-слева или сверху-справа от текущей STU, как в HEVC. Если включено временное предсказание вектора движения, то непосредственно соседний кодированный блок также может быть совмещенным блоком в опорном изображении.

[0132] На фиг. 12 представлена блочная диаграмма, иллюстрирующая пример единицы дерева кодирования (STU) и соседних блоков. В одном примере, только пространственные и временные кандидаты слияния для текущего блока STU используются для инициализации таблицы HMVP. Пример использования пространственных и временных кандидатов слияния HEVC показан на фиг. 12. Порядок вставки следующий: слева (0), сверху (1), сверху-справа (2) и сверху-слева (4). Местоположение временного кандидата слияния указывается как "Т". Следует отметить, что нижний правый временной кандидат слияния и нижний левый (3) кандидат недоступны, так как их местоположения находятся ниже текущей строки STU.

[0133] В другом примере, процесс вывода кандидата слияния для текущего блока STU используется для инициализации таблицы HMVP. В дополнение к пространственным и временным кандидатам слияния, для инициализации могут также использоваться другие кандидаты слияния (например, искусственные кандидаты вектора движения).

[0134] На фиг. 13 представлена блочная диаграмма, иллюстрирующая текущую CU в текущей STU. В некоторых примерах, таблица HMVP инициализируется как пустая в начале кодирования текущей STU. Однако, после того как первая CU кодирована, пространственные и временные кандидаты слияния первой CU, как показано на фиг. 13, добавляются в таблицу HMVP. И затем, MV первой CU также добавляется, если CU

кодируется методом интер-предсказания. Следует отметить, что если первая CU не равна текущей STU, то могут быть добавлены два временных кандидата слияния “T0” и “T1” по порядку. Фиг. 13 показывает пример кандидатов слияния первой CU в STU.

5 [0135] В другом примере, таблица HMVP инициализируется как пустая в начале кодирования текущей STU. Однако после кодирования первой CU, все кандидаты слияния первой CU добавляются в таблицу HMVP. И затем MV первой CU также добавляется, если она кодирована методом интер-предсказания.

10 [0136] Кодер 200 видео и декодер 300 видео также могут выполнять HMVP с инициализацией строки STU. В другом примере, описанная выше инициализация STU для HMVP применяется только к первой STU в строке STU. Подобно HMVP в K0104, процесс обрезки может быть применен к инициализированной таблице для удаления некоторых или всех дубликатов. Процесс обрезки также может не применяться к инициализированной таблице для снижения сложности.

15 [0137] Фиг. 14 представляет собой блок-схему, иллюстрирующую примерный кодер 200 видео, который может выполнять методы настоящего раскрытия. Фиг. 14 приведена в целях пояснения и не должна рассматриваться как ограничение методов, широко иллюстрируемых и описываемых в настоящем раскрытии. В целях пояснения, настоящее раскрытие описывает кодер 200 видео в контексте стандартов кодирования видео, таких как стандарт кодирования видео HEVC и разрабатываемый стандарт кодирования видео H.266. Однако методы настоящего раскрытия не ограничиваются этими стандартами кодирования видео и применимы в общем к кодированию и декодированию видео.

20 [0138] В примере на фиг. 14, кодер 200 видео включает в себя память 230 данных видео, модуль 202 выбора режима, модуль 204 генерации остатка, модуль 206 обработки преобразования, модуль 208 квантования, модуль 210 обратного квантования, модуль 212 обработки обратного преобразования, модуль 214 восстановления, модуль 216 фильтрации, буфер 218 декодированных картинок (DPB) и модуль 220 энтропийного кодирования. Любое или все из памяти 230 данных видео, модуля 202 выбора режима, модуля 204 генерации остатка, модуля 206 обработки преобразования, модуля 208 квантования, модуля 210 обратного квантования, модуля 212 обработки обратного преобразования, модуля 214 восстановления, модуля 216 фильтрации, DPB 218 и модуля 220 энтропийного кодирования могут быть реализованы в одном или более процессорах или в схемах обработки. Кроме того, кодер 200 видео может включать в себя дополнительные или альтернативные процессоры или схемы обработки для выполнения этих и других функций.

35 [0139] Память 230 данных видео может хранить данные видео, подлежащие кодированию компонентами кодера 200 видео. Кодер 200 видео может принимать данные видео, хранящиеся в памяти 230 данных видео, например, от источника 104 видео (фиг. 1). DPB 218 может действовать как память опорных картинок, которая хранит опорные данные видео для использования в предсказании последующих данных видео кодером 200 видео. Память 230 данных видео и DPB 218 могут быть образованы любым из множества устройств памяти, таких как динамическая память с произвольным доступом (DRAM), включая синхронную DRAM (SDRAM), магниторезистивную RAM (MRAM), резистивную RAM (RRAM) или другие типы устройств памяти. Память 230 данных видео и DPB 218 могут быть обеспечены одним и тем же устройством памяти или отдельными устройствами памяти. В различных примерах, память 230 данных видео может быть в чипе (однокристалльной) с другими компонентами кодера 200 видео, как иллюстрируется, или вне чипа относительно этих компонентов.

[0140] В настоящем раскрытии, ссылка на память 230 данных видео не должна интерпретироваться как ограниченная памятью внутренней для кодера 200 видео, если только она специально не описана как таковая, или памятью внешней для кодера 200 видео, если только она специально не описана как таковая. Скорее, ссылка на память 230 данных видео должна пониматься как опорная память, которая хранит данные видео, которые кодер 200 видео получает для кодирования (например, данные видео для текущего блока, который должен кодироваться). Память 106 на фиг. 1 может также обеспечивать временное хранение выходных данных из различных модулей кодера 200 видео.

[0141] Различные модули на фиг. 14 проиллюстрированы, чтобы способствовать пониманию операций, выполняемых кодером 200 видео. Модули могут быть реализованы как фиксированные функциональные схемы, программируемые схемы или их комбинация. Фиксированные функциональные схемы относятся к схемам, которые обеспечивают определенную функциональность и заранее настроены на операции, которые могут выполняться. Программируемые схемы относятся к схемам, которые могут быть запрограммированы для выполнения различных задач и обеспечивают гибкую функциональность в операциях, которые могут выполняться. Например, программируемые схемы могут исполнять программное обеспечение или микропрограммное обеспечение, которое побуждает программируемые схемы работать таким образом, как определяется инструкциями программного обеспечения или микропрограммного обеспечения. Фиксированные функциональные схемы могут исполнять программные инструкции (например, принимать параметры или выводить параметры), но типы операций, которые выполняют фиксированные функциональные схемы, обычно неизменны. В некоторых примерах, один или более модулей могут быть отдельными схемными блоками (фиксированными функциональными блоками или программируемыми блоками), а в некоторых примерах, один или более модулей могут быть интегральными схемами.

[0142] Кодер 200 видео может включать в себя арифметико-логические модули (ALU), элементарные функциональные модули (EFU), цифровые схемы, аналоговые схемы и/или программируемые ядра, образованные программируемыми схемами. В примерах, где операции кодера 200 видео выполняются с использованием программного обеспечения, исполняемого программируемыми схемами, память 106 (фиг. 1) может хранить объектный код программного обеспечения, который кодер 200 видео принимает и исполняет, или другая память в кодере 200 видео (не показана) может хранить такие инструкции.

[0143] Память 230 данных видео сконфигурирована для хранения принятых данных видео. Кодер 200 видео может извлекать изображение данных видео из памяти 230 данных видео и передавать данные видео в модуль 204 генерации остатка и модуль 202 выбора режима. Данные видео в памяти 230 данных видео могут быть необработанными видеоданными, которые должны быть закодированы.

[0144] Модуль 202 выбора режима включает в себя модуль 222 оценки движения, модуль 224 компенсации движения и модуль 226 интра-предсказания. Модуль 202 выбора режима может включать в себя дополнительные функциональные модули для выполнения предсказания видео в соответствии с другими режимами предсказания. В качестве примера, модуль 202 выбора режима может включать в себя модуль палитры, модуль внутриблочного копирования (который может быть частью модуля 222 оценки движения и/или модуля 224 компенсации движения), аффинный модуль, модуль линейной модели (LM) или тому подобное.

[0145] Модуль 202 выбора режима обычно координирует множество проходов кодирования для тестирования комбинаций параметров кодирования и результирующих значений скорости-искажения для таких комбинаций. Параметры кодирования могут включать в себя разбиение STU на CU, режимы предсказания для CU, типы преобразования для остаточных данных CU, параметры квантования для остаточных данных CU и т.д. Модуль 202 выбора режима может, в конечном счете, выбрать комбинацию параметров кодирования, имеющую значения скорости-искажения, которые лучше, чем у других тестируемых комбинаций.

[0146] Кодер 200 видео может разбивать изображение, извлеченное из памяти 230 данных видео, в последовательность STU и инкапсулировать одну или более STU в пределах вырезки. Модуль 202 выбора режима может разбивать STU изображения в соответствии с древовидной структурой, такой как структура QTBT или структура квадродерева HEVC, описанная выше. Как описано выше, кодер 200 видео может формировать одну или более CU из разбиения STU в соответствии с древовидной структурой. Такая CU может также упоминаться, в общем, как “блок видео” или “блок”.

[0147] В общем, модуль 202 выбора режима также управляет своими компонентами (например, модулем 222 оценки движения, модулем 224 компенсации движения, и модулем 226 интра-предсказания), чтобы генерировать блок предсказания для текущего блока (например, текущей CU или, в HEVC, перекрывающейся части PU и TU). Для интер-предсказания текущего блока, модуль 222 оценки движения может выполнять поиск движения для идентификации одного или более близко согласующихся опорных блоков в одной или более опорных изображениях (например, в одной или более из ранее кодированных картинок, сохраненных в DPB 218). В частности, модуль 222 оценки движения может вычислить значение, представляющее, насколько сходен потенциальный опорный блок с текущим блоком, например, в соответствии с суммой абсолютных разностей (SAD), суммой квадратов разностей (SSD), средней абсолютной разностью (MAD), среднеквадратичными разностями (MSD) или тому подобным. Модуль 222 оценки движения может обычно выполнять эти вычисления с использованием разностей от выборки к выборке между текущим блоком и рассматриваемым опорным блоком. Модуль 222 оценки движения может идентифицировать опорный блок, имеющий наименьшее значение, полученное в результате этих вычислений, указывая опорный блок, который наиболее близко согласуется с текущим блоком.

[0148] Модуль 222 оценки движения может формировать один или более векторов движения (MV), которые определяют положения опорных блоков в опорных изображениях относительно положения текущего блока в текущем изображении. Модуль 222 оценки движения может затем предоставить векторы движения в модуль 224 компенсации движения. Например, для однонаправленного интер-предсказания, модуль 222 оценки движения может обеспечить один вектор движения, тогда как для двунаправленного интер-предсказания, модуль 222 оценки движения может обеспечить два вектора движения. Модуль 224 компенсации движения может затем генерировать блок предсказания с использованием векторов движения. Например, модуль 224 компенсации движения может извлекать данные опорного блока с использованием вектора движения. В качестве другого примера, если вектор движения имеет точность дробной выборки, модуль 224 компенсации движения может интерполировать значения для блока предсказания в соответствии с одним или более интерполяционными фильтрами. Кроме того, для двунаправленного интер-предсказания, модуль 224 компенсации движения может извлекать данные для двух опорных блоков, идентифицированных соответствующими векторами движения, и комбинировать

извлеченные данные, например, путем усреднения от выборки к выборке или взвешенного усреднения.

[0149] В соответствии с методами настоящего раскрытия, буфер 218 декодированных картинок может включать в себя один или несколько буферов MVP истории для строк 5 STU. То есть, каждой строке STU может быть выделен свой собственный буфер MVP, или один буфер MVP может быть использован для множества строк STU. В любом случае, кодер 200 видео может сбросить буфер MVP для строки STU в начале декодирования данных видео строки STU. Модуль 224 компенсации движения или 10 другой модуль кодера 200 видео может быть сконфигурирован для сохранения только уникальных векторов движения в буфере MVP. Как обсуждалось выше, модуль 224 компенсации движения или другой модуль кодера 200 видео может быть сконфигурирован для использования правила FIFO для управления информацией движения, сохраняемой в буфере MVP, так что при добавлении вектора движения в буфер MVP, если буфер MVP заполнен, модуль 224 компенсации движения может 15 удалить самый ранний добавленный вектор движения из буфера MVP. В некоторых примерах, кодер 200 видео может поддерживать различные соответствующие буферы MVP для каждой из множества моделей движения, таких как, например, аффинная модель движения, информация движения режима внутриблочного копирования, информация движения компенсации локального освещения, подблочный MVP и 20 временное предсказание движения.

[0150] В качестве другого примера, для интра-предсказания или кодирования с интра-предсказанием, модуль 226 интра-предсказания может генерировать блок предсказания из выборок, соседних с текущим блоком. Например, для направленных режимов модуль 226 интра-предсказания может, обычно, математически комбинировать значения 25 соседних выборок и заполнять этими вычисленными значениями в определенном направлении текущий блок для формирования блока предсказания. В качестве другого примера, для режима DC, модуль 226 интра-предсказания может вычислять среднее значение соседних выборок для текущего блока и генерировать блок предсказания так, чтобы включать это полученное среднее значение для каждой выборки блока 30 предсказания.

[0151] Модуль 202 выбора режима предоставляет блок предсказания в модуль 204 генерации остатка. Модуль 204 генерации остатка получает необработанную, некодированную версию текущего блока из памяти 230 данных видео и блок предсказания из модуля 202 выбора режима. Модуль 204 генерации остатка вычисляет 35 от выборки к выборке разности между текущим блоком и блоком предсказания. Полученные разности от выборки к выборке определяют остаточный блок для текущего блока. В некоторых примерах, модуль 204 генерации остатка может также определять разности между значениями выборок в остаточном блоке для генерации остаточного блока с использованием дифференциальной импульсно-кодовой модуляции остатка 40 (RDPCM). В некоторых примерах, модуль 204 генерации остатка может быть сформирован с использованием одной или нескольких схем вычитания, которые выполняют двоичное вычитание.

[0152] В примерах, где модуль 202 выбора режима разбивает CU на PU, каждая PU может быть ассоциирована с единицей предсказания яркости и соответствующими 45 единицами предсказания цветности. Кодер 200 видео и декодер 300 видео могут поддерживать PU, имеющие различные размеры. Как указано выше, размер CU может относиться к размеру блока кодирования яркости CU, а размер PU может относиться к размеру единицы предсказания яркости PU. Предполагая, что размер конкретной CU

равен $2N \times 2N$, кодер 200 видео может поддерживать размеры PU $2N \times 2N$ или $N \times N$ для интра-предсказания и симметричные размеры PU $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ или аналогичные для интер-предсказания. Кодер 200 видео и декодер 300 видео также могут поддерживать асимметричное разбиение для размеров PU $2N \times nU$, $2N \times nD$, $nL \times 2N$ и $nR \times 2N$ для интер-предсказания.

[0153] В примерах, где модуль выбора режима дополнительно не разбивает CU на PU, каждая CU может быть ассоциирована с блоком кодирования яркости и соответствующими блоками кодирования цветности. Как указано выше, размер CU может относиться к размеру блока кодирования яркости CU. Кодер 200 видео и декодер 120 видео могут поддерживать размеры CU $2N \times 2N$, $2N \times N$ или $N \times 2N$.

[0154] Для других методов кодирования видео, таких как кодирование в режиме внутриблочного копирования, кодирование в аффинном режиме и кодирование в режиме линейной модели (LM), в качестве нескольких примеров, модуль 202 выбора режима, посредством соответствующих модулей, ассоциированных с методами кодирования, генерирует блок предсказания для текущего кодируемого блока. В некоторых примерах, таких как кодирование в режиме палитры, модуль 202 выбора режима может не генерировать блок предсказания, а вместо этого генерировать синтаксические элементы, указывающие способ восстановления блока на основе выбранной палитры. В таких режимах, модуль 202 выбора режима может предоставлять эти синтаксические элементы модулю 220 энтропийного кодирования для кодирования.

[0155] Как описано выше, модуль 204 генерации остатка принимает данные видео для текущего блока и соответствующего блока предсказания. Затем модуль 204 генерации остатка генерирует остаточный блок для текущего блока. Для генерации остаточного блока, модуль 204 генерации остатка вычисляет разности от выборки к выборке между блоком предсказания и текущим блоком.

[0156] Модуль 206 обработки преобразования применяет одно или более преобразований к остаточному блоку для генерации блока коэффициентов преобразования (упоминаемого здесь как “блок коэффициентов преобразования”). Модуль 206 обработки преобразования может применять различные преобразования к остаточному блоку для формирования блока коэффициентов преобразования. Например, модуль 206 обработки преобразования может применять дискретное косинусное преобразование (DCT), направленное преобразование, преобразование Карунена-Лоэва (KLT) или концептуально аналогичное преобразование к остаточному блоку. В некоторых примерах, модуль 206 обработки преобразования может выполнять множественные преобразования к остаточному блоку, например, первичное преобразование и вторичное преобразование, такое как преобразование вращения. В некоторых примерах, модуль 206 обработки преобразования не применяет преобразования к остаточному блоку.

[0157] Модуль 208 квантования может квантовать коэффициенты преобразования в блоке коэффициентов преобразования, чтобы сформировать блок квантованных коэффициентов преобразования. Модуль 208 квантования может квантовать коэффициенты преобразования блока коэффициентов преобразования в соответствии со значением параметра квантования (QP), ассоциированным с текущим блоком. Кодер 200 видео (например, посредством модуля 202 выбора режима) может регулировать степень квантования, применяемого к блокам коэффициентов преобразования, ассоциированным с текущим блоком, путем настройки значения QP, ассоциированного с CU. Квантование может вводить потери информации, и, таким образом, квантованные коэффициенты преобразования могут иметь более низкую точность, чем исходные

коэффициенты преобразования, сформированные модулем 206 обработки преобразования.

[0158] Модуль 210 обратного квантования и модуль 212 обработки обратного преобразования могут применять обратное квантование и обратные преобразования к блоку квантованных коэффициентов преобразования, соответственно, для восстановления остаточного блока из блока коэффициентов преобразования. Модуль 214 восстановления может формировать восстановленный блок, соответствующий текущему блоку (хотя, возможно, с некоторой степенью искажения), на основе восстановленного остаточного блока и блока предсказания, генерируемого модулем 202 выбора режима. Например, модуль 214 восстановления может добавлять выборки восстановленного остаточного блока к соответствующим выборкам из блока предсказания, генерируемого модулем 202 выбора режима, чтобы сформировать восстановленный блок.

[0159] Модуль 216 фильтрации может выполнять одну или более операций фильтрации на восстановленных блоках. Например, модуль 216 фильтрации может выполнять операции устранения блочности для уменьшения артефактов блочности вдоль краев CU. Операции модуля 216 фильтрации могут быть пропущены, в некоторых примерах.

[0160] Кодер 200 видео сохраняет восстановленные блоки в DPB 218. Например, в случаях, где операции модуля 216 фильтрации не требуются, модуль 214 восстановления может сохранять восстановленные блоки в DPB 218. В примерах, где требуются операции модуля 216 фильтрации, модуль 216 фильтрации может сохранять отфильтрованные восстановленные блоки в DPB 218. Модуль 222 оценки движения и модуль 224 компенсации движения могут извлекать опорное изображение из DPB 218, сформированную из восстановленных (и потенциально отфильтрованных) блоков, для интер-предсказания блоков последующих кодированных картинок. Кроме того, модуль интра-предсказания 226 может использовать восстановленные блоки в DPB 218 текущей изображения для интра-предсказания других блоков в текущем изображении.

[0161] В общем, модуль 220 энтропийного кодирования может энтропийно кодировать синтаксические элементы, принятые от других функциональных компонентов кодера 200 видео. Например, модуль 220 энтропийного кодирования может энтропийно кодировать блоки квантованных коэффициентов преобразования из модуля 208 квантования. В качестве другого примера, модуль 220 энтропийного кодирования может энтропийно кодировать синтаксические элементы предсказания (например, информации движения для интер-предсказания или информацию интра-режима для интра-предсказания) из модуля 202 выбора режима. Модуль 220 энтропийного кодирования может выполнять одну или более операций энтропийного кодирования на синтаксических элементах, которые являются другим примером данных видео, чтобы генерировать энтропийно-закодированные данные. Например, модуль 220 энтропийного кодирования может выполнять операцию контекстно-адаптивного кодирования переменной длины (CAVLC), операцию CABAC, операцию кодирования от переменной к переменной (B2B) длине, операцию основанного на синтаксисе контекстно-адаптивного двоичного арифметического кодирования (SBAC), операцию энтропийного кодирования с разбиением интервалов вероятности (PIPE), операцию экспоненциального кодирования Голomba или другой тип операции энтропийного кодирования данных. В некоторых примерах, модуль 220 энтропийного кодирования может работать в режиме обхода, где синтаксические элементы не являются энтропийно кодируемыми.

[0162] Кодер 200 видео может выводить битовый поток, который включает в себя энтропийно закодированные синтаксические элементы, необходимые для восстановления

блоков вырезки или изображения. В частности, модуль 220 энтропийного кодирования может выводить битовый поток.

[0163] Описанные выше операции описаны по отношению к блоку. Такое описание следует понимать как операции для блока кодирования яркости и/или блоков кодирования цветности. Как описано выше, в некоторых примерах, блоки кодирования яркости и цветности являются компонентами яркости и цветности CU. В некоторых примерах, блок кодирования яркости и блоки кодирования цветности являются компонентами яркости и цветности PU.

[0164] В некоторых примерах, операции, выполняемые в отношении блока кодирования яркости, не требуется повторять для блоков кодирования цветности. В качестве одного примера, операции для идентификации вектора движения (MV) и опорной изображения для блока кодирования яркости не требуется повторять для идентификации MV и опорной изображения для блоков цветности. Скорее, MV для блока кодирования яркости может быть масштабирован для определения MV для блоков цветности, и опорное изображение может быть тем же самым. В качестве другого примера, процесс интра-предсказания может быть одинаковым для блока кодирования яркости и блоков кодирования цветности.

[0165] Кодер 200 видео представляет пример устройства, сконфигурированного для кодирования данных видео, включающего в себя память, сконфигурированную, чтобы хранить данные видео, и один или более модулей обработки, реализованных в схемах и сконфигурированных, чтобы сохранять информацию движения для первой строки единиц дерева кодирования (CTU) в первом буфере памяти предсказателя вектора движения (MVP) истории и сохранять информацию движения для второй строки CTU во втором буфере памяти MVP истории, причем вторая строка CTU отличается от первой строки CTU.

[0166] Кодер 200 видео также представляет пример устройства, сконфигурированного, чтобы кодировать данные видео, включающего в себя память, сконфигурированную, чтобы хранить данные видео, и один или более модулей обработки, реализованных в схемах и сконфигурированных, чтобы сохранять кодированную информацию движения в буфере вектора движения предсказателя (MVP) истории, сохранять другой тип информации движения в буфере MVP истории, и кодировать информацию движения блока данных видео с использованием информации движения буфера MVP истории.

[0167] Кодер 200 видео также представляет пример устройства, сконфигурированного для кодирования данных видео, включающего в себя память, сконфигурированную, чтобы хранить данные видео, и один или более модулей обработки, реализованных в схемах и сконфигурированных, чтобы сохранять множество различных типов информации движения в соответствующих различных буферах предсказателя вектора движения (MVP) истории.

[0168] Фиг. 15 представляет собой блок-схему, иллюстрирующую примерный декодер 300 видео, который может выполнять методы настоящего раскрытия. Фиг. 15 обеспечена в целях пояснения и не ограничивает методы, широко иллюстрируемые и описанные в настоящем раскрытии. Для целей пояснения, настоящее раскрытие описывает декодер 300 видео в соответствии с методами JEM и HEVC. Однако методы настоящего раскрытия могут выполняться устройствами кодирования видео, которые сконфигурированы в соответствии с другими стандартами кодирования видео.

[0169] В примере на фиг. 15, декодер 300 видео включает в себя память 320 буфера кодированной изображения (CPB), модуль 302 энтропийного декодирования, модуль 304 обработки предсказания, модуль 306 обратного квантования, модуль 308 обработки

обратного преобразования, модуль 310 восстановления, модуль 312 фильтрации и буфер 314 декодированных картинок (DPB). Любое или все из памяти CPB 320, модуля 302 энтропийного декодирования, модуля 304 обработки предсказания, модуля 306 обратного квантования, модуля 308 обработки обратного преобразования, модуля 310 восстановления, модуля 312 фильтрации и DPB 314 могут быть реализованы в одном или более процессорах или в схемах обработки. Кроме того, декодер 300 видео может включать в себя дополнительные или альтернативные процессоры или схемы обработки для выполнения этих и других функций.

[0170] Модуль 304 обработки предсказания включает в себя модуль 316 компенсации движения и модуль 318 интра-предсказания. Модуль 304 обработки предсказания может включать в себя модули суммирования для выполнения предсказания в соответствии с другими режимами предсказания. В качестве примеров, модуль 304 обработки предсказания может включать в себя модуль палитры, модуль внутривидеоблочного копирования (который может образовывать часть модуля 316 компенсации движения), аффинный модуль, модуль линейной модели (LM) или тому подобное. В других примерах, декодер 300 видео может включать в себя больше, меньше или другие функциональные компоненты.

[0171] Память 320 CPB может хранить данные видео, такие как закодированный битовый поток видео, для декодирования компонентами декодера 300 видео. Данные видео, хранящиеся в памяти 320 CPB, могут быть получены, например, из считываемого компьютером носителя 110 (фиг. 1). Память 320 CPB может включать в себя CPB, который сохраняет закодированные данные видео (например, синтаксические элементы) из закодированного битового потока видео. Кроме того, память 320 CPB может хранить данные видео, отличные от синтаксических элементов закодированной изображения, такие как временные данные, представляющие выходы из различных модулей декодера 300 видео. DPB 314 обычно хранит декодированные изображения, которые декодер 300 видео может выводить и/или использовать в качестве опорных данных видео при декодировании последующих данных или картинок закодированного битового потока видео. Память 320 CPB и DPB 314 могут быть образованы любым из множества устройств памяти, таких как динамическая память с произвольным доступом (DRAM), включая синхронную DRAM (SDRAM), магниторезистивную RAM (MRAM), резистивную RAM (RRAM) или другие типы устройств памяти. Память 320 CPB и DPB 314 могут быть предоставлены одним и тем же устройством памяти или отдельными устройствами памяти. В различных примерах, память 320 CPB может быть в чипе с другими компонентами декодера 300 видео или вне чипа относительно этих компонентов.

[0172] Дополнительно или альтернативно, в некоторых примерах, декодер 300 видео может извлекать закодированные данные видео из памяти 120 (фиг. 1). То есть память 120 может хранить данные, как обсуждено выше для памяти 320 CPB. Аналогично, память 120 может хранить инструкции, которые должны исполняться декодером 300 видео, когда некоторая или вся из функциональности декодера 300 видео реализована в программном обеспечении, подлежащем исполнению схемой обработки декодера 300 видео.

[0173] Различные модули, показанные на фиг. 15, проиллюстрированы, чтобы способствовать пониманию операций, выполняемых декодером 300 видео. Модули могут быть реализованы как фиксированные функциональные схемы, программируемые схемы или их комбинации. Подобно фиг. 14, фиксированные функциональные схемы относятся к схемам, которые обеспечивают определенную функциональность и заранее настроены на операции, которые могут выполняться. Программируемые схемы

относятся к схемам, которые могут быть запрограммированы для выполнения различных задач и обеспечивают гибкую функциональность в операциях, которые могут выполняться. Например, программируемые схемы могут исполнять программное обеспечение или микропрограммное обеспечение, которое побуждает программируемые 5 схемы работать таким образом, как определяется инструкциями программного обеспечения или микропрограммного обеспечения. Фиксированные функциональные схемы могут исполнять программные инструкции (например, принимать параметры или выводить параметры), но типы операций, которые выполняют фиксированные функциональные схемы, обычно неизменны. В некоторых примерах, один или более 10 модулей могут быть отдельными схемными блоками (фиксированными функциональными блоками или программируемыми блоками), а в некоторых примерах, один или более модулей могут быть интегральными схемами.

[0174] Декодер 300 видео может включать в себя ALU, EFU, цифровые схемы, аналоговые схемы и/или программируемые ядра, образованные программируемыми 15 схемами. В примерах, где операции декодера 300 видео выполняются программным обеспечением, исполняемым на программируемых схемах, однокристалльная память (в чипе) или память вне чипа может хранить инструкции (например, объектный код) программного обеспечения, которые получает и исполняет декодер 300 видео.

[0175] Модуль 302 энтропийного декодирования может получать закодированные 20 данные видео из СРВ и энтропийно декодировать данные видео для воспроизведения синтаксических элементов. Модуль 304 обработки предсказания, модуль 306 обратного квантования, модуль 308 обработки обратного преобразования, модуль 310 восстановления и модуль 312 фильтрации могут генерировать декодированные данные видео на основе синтаксических элементов, извлеченных из битового потока.

[0176] В общем, декодер 300 видео восстанавливает изображение на поблочной 25 основе. Декодер 300 видео может выполнять операцию восстановления на каждом блоке индивидуально (где блок, который восстанавливается, т.е. декодируется, в текущее время, может называться “текущим блоком”).

[0177] Модуль 302 энтропийного декодирования может энтропийно декодировать 30 синтаксические элементы, определяющие квантованные коэффициенты преобразования блока квантованных коэффициентов преобразования, а также информацию преобразования, такую как указание(я) параметра квантования (QP) и/или режима преобразования. Модуль 306 обратного квантования может использовать QP, ассоциированный с блоком квантованных коэффициентов преобразования, для 35 определения степени квантования и, аналогично, степени обратного квантования для применения модулем 306 обратного квантования. Модуль 306 обратного квантования может, например, выполнять побитовую операцию сдвига влево для обратного квантования квантованных коэффициентов преобразования. При этом модуль 306 обратного квантования может формировать блок коэффициентов преобразования, 40 включающий коэффициенты преобразования.

[0178] После того как модуль 306 обратного квантования сформировал блок 45 коэффициентов преобразования, модуль 308 обработки обратного преобразования может применить одно или более обратных преобразований к блоку коэффициентов преобразования для генерации остаточного блока, ассоциированного с текущим блоком. Например, блок обработки 308 обратного преобразования может применять обратное DST, обратное целочисленное преобразование, обратное преобразование Карунена-Лоэва (KLT), обратное преобразование вращения, обратное направленное преобразование или другое обратное преобразование к блоку коэффициентов.

[0179] Кроме того, модуль 304 обработки предсказания генерирует блок предсказания в соответствии с синтаксическими элементами информации предсказания, которые были энтропийно декодированы модулем 302 энтропийного декодирования. Например, если синтаксические элементы информации предсказания указывают, что текущий блок является интер-предсказанным, модуль 316 компенсации движения может генерировать блок предсказания. В этом случае, синтаксические элементы информации предсказания могут указывать опорное изображение в DPB 314, из которой следует извлекать опорный блок, а также вектор движения, идентифицирующий местоположение опорного блока в опорном изображении относительно местоположения текущего блока в текущем изображении. Модуль 316 компенсации движения может, в принципе, выполнять процесс интер-предсказания способом, который по существу аналогичен тому, который описан в отношении модуля 224 компенсации движения (фиг. 14).

[0180] В соответствии с методами настоящего раскрытия, буфер 314 декодированных картинок может включать в себя один или более буферов MVP истории для строк STU. То есть каждой строке STU может быть выделен свой собственный буфер MVP, или один буфер MVP может быть использован для множества строк STU. В любом случае, декодер 300 видео может сбросить буфер MVP для строки STU в начале кодирования видео данной строки STU. Модуль 316 компенсации движения или другой модуль декодера 300 видео может быть сконфигурирован, чтобы хранить только уникальные вектора движения в буфере MVP. Как обсуждалось выше, модуль 316 компенсации движения или другой модуль декодера 300 видео может быть сконфигурирован, чтобы использовать правило FIFO для управления информацией движения, сохраненной в буфере MVP, так что при добавлении вектора движения в буфер MVP, если буфер MVP заполнен, модуль 316 компенсации движения может удалить самый ранний добавленный вектор движения из буфера MVP. В некоторых примерах, декодер 300 видео может поддерживать различные соответствующие буферы MVP для каждой из множества моделей движения, таких как, например, аффинная модель движения, информация движения режима внутриблочного копирования, информация движения компенсации локального освещения, подблочный MVP и временное предсказание движения.

[0181] В качестве другого примера, если синтаксические элементы информации предсказания указывают, что текущий блок является интра-предсказанным, модуль 318 интра-предсказания может генерировать блок предсказания в соответствии с режимом интра-предсказания, указанным синтаксическими элементами информации предсказания. Вновь, модуль 318 интра-предсказания может, в принципе, выполнять процесс интра-предсказания способом, который по существу аналогичен описанному в отношении модуля 226 интра-предсказания (фиг. 14). Модуль 318 интра-предсказания может извлекать данные из соседних выборок для текущего блока из DPB 314.

[0182] Модуль 310 восстановления может восстанавливать текущий блок с использованием блока предсказания и остаточного блока. Например, модуль 310 восстановления может добавлять выборки остаточного блока к соответствующим выборкам блока предсказания для восстановления текущего блока.

[0183] Модуль 312 фильтрации может выполнять одну или более операций фильтрации на восстановленных блоках. Например, модуль 312 фильтрации может выполнять операции устранения блочности для уменьшения артефактов блочности вдоль краев восстановленных блоков. Операции модуля 312 фильтрации не обязательно выполняются во всех примерах.

[0184] Декодер 300 видео может сохранить восстановленные блоки в DPB 314. Например, в случаях, когда операции модуля 312 фильтрации не требуются, модуль

310 восстановления может сохранить восстановленные блоки в DPB 314. В случаях, когда требуются операции модуля 312 фильтрации, модуль 312 фильтрации может сохранить отфильтрованные восстановленные блоки в DPB 314. Как обсуждалось выше, DPB 314 может предоставлять опорную информацию, такую как выборки текущей
5 изображения для интра-предсказания и ранее декодированные изображения для последующей компенсации движения, в модуль обработки 304 предсказания. Кроме того, декодер 300 видео может выводить декодированные изображения из DPB 314 для последующего представления на устройстве отображения, таком как устройство 118 отображения на фиг. 1.

10 [0185] Декодер 300 видео представляет пример устройства, сконфигурированного для декодирования данных видео, включающего в себя память, сконфигурированную, чтобы хранить данные видео, и один или более модулей обработки, реализованных в схемах и сконфигурированных, чтобы сохранять информацию движения для первой строки единиц дерева кодирования (СТУ) (которая может обрабатываться первым
15 подпроцессом процесса кодирования видео) в первом буфере памяти предсказателя вектора движения (MVP) истории, и сохранять информации движения для второй строки СТУ (которая может обрабатываться вторым подпроцессом процесса кодирования видео) во втором буфере памяти MVP истории, причем вторая строка СТУ отличается от первой строки СТУ. Второй подпроцесс может отличаться от первого подпроцесса.

20 [0186] Декодер 300 видео также представляет пример устройства, сконфигурированного для декодирования данных видео, включающего в себя память, сконфигурированную, чтобы хранить данные видео, и один или более модулей обработки, реализованных в схемах и сконфигурированных, чтобы сохранять
25 кодированную информацию движения в буфере предсказателя вектора движения (MVP) истории, сохранять другой тип информации движения в буфере MVP истории, и кодировать информацию движения для блока данных видео с использованием информации движения буфера MVP истории.

[0187] Декодер 300 видео также представляет пример устройства, сконфигурированного для декодирования данных видео, включающего в себя память,
30 сконфигурированную, чтобы хранить данные видео, и один или более модулей обработки, реализованных в схемах и сконфигурированных, чтобы сохранять множество различных видов информации движения в соответствующих буферах предсказателя вектора движения (MVP) истории.

[0188] На фиг. 16 представлена блок-схема последовательности операций, иллюстрирующая пример способа для кодирования текущего блока в соответствии с
35 методами настоящего раскрытия. Текущий блок может содержать текущую СУ. Хотя описание приведено относительно кодера 200 видео (фиг. 1 и 14), следует понимать, что другие устройства могут быть сконфигурированы для выполнения способа, аналогичного показанному на фиг. 16.

40 [0189] В этом примере, кодер 200 видео первоначально предсказывает текущий блок с помощью информации движения (350). Например, кодер 200 видео может сформировать блок предсказания для текущего блока с использованием информации движения. Кодер 200 видео может затем вычислить остаточный блок для текущего блока (352). Чтобы вычислить остаточный блок, кодер 200 видео может вычислить
45 разность между исходным, некодированным блоком и блоком предсказания для текущего блока. Кодер 200 видео может затем преобразовывать и квантовать коэффициенты остаточного блока (354). Затем кодер 200 видео может сканировать квантованные коэффициенты преобразования остаточного блока (356). Во время

сканирования или после сканирования, кодер 200 видео может энтропийно кодировать коэффициенты и информацию движения (358) с использованием методов настоящего раскрытия. Кодер 200 видео может кодировать коэффициенты с использованием CAVLC или CABAC.

5 [0190] Кодер 200 видео может формировать список кандидатов информации движения, включающий, например, кандидатов HMVP, в соответствии с любым или всеми из методов настоящего раскрытия, выбрать индекс кандидата, представляющий предсказатель для информации движения блока, и энтропийно кодировать индекс кандидата. В соответствии с методами настоящего раскрытия, кодер 200 видео может
10 сбросить буфер MVP перед использованием буфера MVP, чтобы сохранить информацию движения соответствующей строки STU. В некоторых примерах, каждая строка STU может иметь свой собственный буфер MVP, или один буфер MVP может использоваться для множества строк STU. Кроме того, кодер 200 видео может сохранить множество типов информации движения в буферах MVP, например, одном и том же буфере или
15 различных буферах соответствующих моделей движения. Кодер 200 видео может кодировать информацию движения текущего блока с использованием предсказателей векторов движения, выбранных из данных буфера MVP. Кодер 200 видео может затем выводить энтропийно закодированные данные блока (360), например, включающие в себя данные для коэффициентов и информацию движения, такую как индекс кандидата.
20 [0191] Таким образом, способ в соответствии с фиг. 16 представляет пример способа, включающего в себя сохранение информации движения для изображения первой строки единиц дерева кодирования (STU) в первом буфере памяти предсказателя вектора движения (MVP) истории; сброс второго буфера памяти MVP истории; и после сброса второго буфера MVP истории, сохранение информации движения для изображения
25 второй строки STU во втором буфере MVP истории, причем вторая строка STU отличается от первой строки STU.

[0192] Способ согласно фиг. 16 также представляет пример способа, включающего в себя сохранение информации движения в буфере предсказателя вектора движения (MVP) истории; сохранение другого типа информации движения в буфере MVP истории;
30 и кодирование информации движения блока данных видео с использованием информации движения буфера MVP истории.

[0193] Способ на фиг. 16 также представляет пример способа, включающего в себя сохранение множества различных типов информации движения в соответствующих различных буферах предсказателей векторов движения (MVP) истории.

35 [0194] На фиг. 17 представлена блок-схема последовательности операций, иллюстрирующая пример способа для декодирования текущего блока данных видео в соответствии с методами настоящего раскрытия. Текущий блок может содержать текущую CU. Хотя описание приведено в отношении декодера 300 видео (фиг. 1 и 15), следует понимать, что другие устройства могут быть сконфигурированы, чтобы
40 выполнять способ, аналогичный показанному на фиг. 17.

[0195] Декодер 300 видео может принимать энтропийно кодированные данные для текущего блока, такие как энтропийно кодированная информация предсказания и энтропийно кодированные данные для коэффициентов остаточного блока, соответствующего текущему блоку (370). Как обсуждалось выше, энтропийно
45 кодированная информация предсказания может включать в себя, например, индекс кандидата в список кандидатов, который может включать кандидатов HMVP в соответствии с методами настоящего раскрытия. Декодер 300 видео может энтропийно декодировать энтропийно кодированные данные, чтобы определять информацию

предсказания для текущего блока и восстанавливать коэффициенты остаточного блока (372). Декодер 300 видео может предсказывать текущий блок (374), например, с использованием режима интер-предсказания, как указано информацией предсказания для текущего блока, что вычислить блок предсказания для текущего блока.

5 [0196] В частности, декодер 300 видео может формировать список кандидатов, включающий кандидатов HMVP, как описано выше, а затем определять кандидата из списка кандидатов, чтобы использовать в качестве предсказателя вектора движения для текущего блока с использованием декодированного индекса кандидата. В соответствии с методами настоящего раскрытия, декодер 300 видео может сбросить 10 буфер MVP перед использованием буфера MVP для сохранения информации движения соответствующей строки STU. В некоторых примерах, каждая строка STU может иметь свой собственный буфер MVP, или один буфер MVP может использоваться для множества строк STU. Кроме того, декодер 300 видео может сохранить множество типов информации движения в буферах MVP, например, в том же самом буфере или 15 различных буферах, соответствующих моделям движения. Декодер 300 видео может выбирать с использованием предсказателей векторов движения из данных буфера MVP.

[0197] Декодер 300 видео может затем восстановить вектор движения для текущего блока с использованием предсказателя вектора движения, а затем предсказать текущий блок с использованием вектора движения, чтобы сгенерировать блок предсказания. 20 Декодер 300 видео может затем обратно сканировать восстановленные коэффициенты (376), чтобы создать блок квантованных коэффициентов преобразования. Декодер 300 видео может затем обратно квантовать и обратно преобразовать коэффициенты для получения остаточного блока (378). Декодер 300 видео может, в конечном счете, декодировать текущий блок путем комбинирования блока предсказания и остаточного 25 блока (380).

[0198] Таким образом, способ согласно фиг. 17 представляет пример способа, включающего в себя сохранение информации движения для первой строки единиц дерева кодирования (STU) изображения в первом буфере памяти предсказателя вектора движения (MVP) истории; сброс второго буфера памяти MVP истории; и после сброса 30 второго буфера MVP истории, сохранение информации движения для второй строки STU изображения во втором буфере MVP истории, причем вторая строка STU отличается от первой строки STU.

[0199] Способ согласно фиг. 17 также представляет пример способа, включающего в себя сохранение информации движения в буфере вектора движения предсказателя 35 (MVP) истории; сохранение другого типа информации движения в буфере MVP истории; и кодирование информации движения блока данных видео с использованием информации движения буфера MVP истории.

[0200] Способ согласно фиг. 17 также представляет пример способа, включающего в себя сохранение множества различных типов информации движения в соответствующих 40 буферах предсказателя вектора движения (MVP) истории.

[0201] На фиг. 18 представлена блок-схема последовательности операций, иллюстрирующая примерный способ кодирования (кодирования или декодирования) 45 данных видео в соответствии с методами настоящего раскрытия. Например, способ согласно фиг. 18 может быть выполнен во время этапа 350 на фиг. 16 или этапа 374 на фиг. 17. Для примера и пояснения, способ согласно фиг. 18 поясняется в отношении декодера 300 видео, хотя кодер 200 видео может также выполнять этот или аналогичный способ.

[0202] Декодер 300 видео может кодировать блоки первой строки STU изображения

(390), например, с использованием интра- или интер-предсказания. Декодер 300 видео сохраняет информацию движения первой строки единиц дерева кодирования (STU) изображения в первом буфере (392), например, DPB 314. Декодер 300 видео может использовать информацию движения первого буфера, чтобы кодировать информацию движения, используемую во время кодирования интер-предсказания. В некоторых примерах, первый подпроцесс процесса кодирования видео, выполняемого декодером 300 видео, может кодировать первую строку STU.

[0203] Декодер 300 видео может также сбросить второй буфер (394), например, DPB 314. Второй буфер может быть тем же самым, что и первый буфер, или другим буфером. Декодер 300 видео может также кодировать блоки второй строки STU (396). Декодер 300 видео может сохранить информацию движения второй строки STU во втором буфере (398). В некоторых примерах, второй подпроцесс процесса кодирования видео, выполняемого декодером 300 видео, может кодировать вторую строку STU, причем второй подпроцесс отличается от первого подпроцесса.

[0204] Таким образом, способ согласно фиг. 18 представляет пример способа, включающего в себя сохранение информации движения для первой строки единиц дерева кодирования (STU) изображения в первом буфере памяти предсказателя вектора движения (MVP) истории; сброс второго буфера памяти MVP истории; и после сброса второго буфера MVP истории, сохранение информации движения для второй строки STU изображения во втором буфере MVP истории, причем вторая строка STU отличается от первой строки STU.

[0205] Следует иметь в виду, что, в зависимости от примера, некоторые действия или события любого из описанных здесь методов могут выполняться в другой последовательности, могут добавляться, объединяться или вообще исключаться (например, не все описанные действия или события необходимы для практической реализации методов). Кроме того, в некоторых примерах, действия или события могут выполняться одновременно, например, посредством многопоточной обработки, обработки прерываний или с помощью множества процессоров, а не последовательно.

[0206] В одном или более примерах, описанные функции могут быть реализованы в аппаратных средствах, программном обеспечении, микропрограммном обеспечении или любой их комбинации. При реализации в программном обеспечении, функции могут сохраняться или передаваться как одна или более инструкций или кода на считываемом носителе и исполняться аппаратным модулем обработки. Считываемые компьютером носители могут включать в себя считываемые компьютером носители, которые соответствуют осязаемому носителю, такому как носитель хранения данных, или коммуникационной среде, включающей в себя любой носитель, который обеспечивает передачу компьютерной программы из одного места в другое, например, в соответствии с протоколом связи. Таким образом, считываемые компьютером носители обычно могут соответствовать (1) осязаемым считываемым компьютером носителям хранения, которые являются энергонезависимыми, или (2) коммуникационному носителю (среде), такому как сигнал или несущая волна. Носителями хранения данных могут быть любые доступные носители, к которым может получить доступ один или более компьютеров или один или более процессоров для извлечения инструкций, кода и/или структур данных для реализации методов, описанных в настоящем раскрытии. Компьютерный программный продукт может включать в себя считываемый компьютером носитель.

[0207] В качестве примера, а не ограничения, такие считываемые компьютером носители хранения могут содержать RAM, ROM, EEPROM, CD-ROM или другие

устройства хранения на оптических дисках, устройства хранения на магнитных дисках или другие магнитные устройства хранения, флэш-память или любой другой носитель, который может быть использован для хранения желательного программного кода в форме инструкций или структур данных и к которому может получить доступ компьютер. Кроме того, любое соединение надлежащим образом определяется как считываемый компьютером носитель. Например, если инструкции передаются с веб-сайта, сервера или другого удаленного источника с использованием коаксиального кабеля, волоконно-оптического кабеля, витой пары, цифровой абонентской линии (DSL) или беспроводных технологий, таких как инфракрасное излучение, радио и микроволны, то коаксиальный кабель, волоконно-оптический кабель, витая пара, DSL или беспроводные технологии, такие как инфракрасное излучение, радио и микроволны, включаются в определение носителя. Однако следует понимать, что считываемые компьютером носители хранения и носители хранения данных не включают в себя соединения, несущие волны, сигналы или другие временные носители, а вместо этого направлены на не-временные, осязаемые носители информации. Магнитный диск (disk) и оптический диск (disc), как используется в настоящем документе, включают в себя компакт-диск (CD), лазерный диск, оптический диск, универсальный цифровой диск (DVD), гибкий диск и Blu-ray диск, где магнитные диски обычно воспроизводят данные магнитным способом, тогда как оптические диски воспроизводят данные оптическим способом с помощью лазеров. Комбинации вышеперечисленного также должны быть включены в объем считываемых компьютером носителей.

[0208] Инструкции могут исполняться одним или более процессорами, такими как один или более цифровых сигнальных процессоров (DSP), микропроцессоров общего назначения, специализированных интегральных схем (ASIC), программируемых вентильных матриц (FPGA) или других эквивалентных интегральных схем или дискретных логических схем. Соответственно, термины “процессор” и “схема обработки”, используемые здесь, могут относиться к любой из вышеперечисленных структур или любой другой структуре, подходящей для реализации методов, описанных здесь. Кроме того, в некоторых аспектах, функциональность, описанная здесь, может быть обеспечена в выделенных аппаратных средствах и/или программных модулях, сконфигурированных для кодирования и декодирования или встроенных в комбинированный кодек. Кроме того, методы могут быть полностью реализованы в одной или более схемах или логических элементах.

[0209] Методики согласно настоящему раскрытию могут быть реализованы в широком разнообразии устройств или приборов, включая беспроводный телефон, интегральную схему (IC) или набор интегральных схем (например, чипсет). Различные компоненты, модули или блоки описаны в настоящем раскрытии с акцентом на функциональных аспектах устройств, сконфигурированных для выполнения раскрытых методов, но не обязательно требующих реализации различными аппаратными модулями. Скорее, как описано выше, различные модули могут быть объединены в аппаратный модуль кодека или обеспечены набором взаимодействующих аппаратных модулей, включая один или более процессоров, как описано выше, во взаимосвязи с подходящим программным обеспечением и/или микропрограммным обеспечением.

[0210] Были описаны различные примеры. Эти и другие примеры входят в объем следующей формулы изобретения.

(57) Формула изобретения

1. Способ кодирования данных видео, содержащий этапы, на которых:

кодируют информацию движения для первой строки единиц дерева кодирования (СТУ) изображения, причем информация движения для первой строки СТУ включает в себя первые один или более векторов движения для первой строки СТУ;

формируют первые один или более блоков предсказания для первых одной или более единиц кодирования (СУ) первой строки СТУ с использованием первых опорных блоков, идентифицируемых информацией движения для первой строки СТУ;

кодируют первые один или более остаточных блоков для первых одной или более СУ, причем первые один или более остаточных блоков представляют соответственные разности между первыми одним или более блоками предсказания и первыми одной или более СУ;

сохраняют информацию движения для первой строки СТУ упомянутого изображения в первом буфере предиктора векторов движения, основывающегося на истории (MVP истории), в памяти, включая сохранение первых одного или более векторов движения для первой строки СТУ в первом буфере MVP истории для использования в качестве первых предикторов векторов движения, чтобы предсказывать первые один или более других векторов движения упомянутого изображения;

сбрасывают второй буфер MVP истории в памяти;

кодируют информацию движения для второй строки СТУ упомянутого изображения, причем информация движения для второй строки СТУ включает в себя вторые один или более векторов движения для второй строки СТУ;

формируют вторые один или более блоков предсказания для вторых одной или более СУ второй строки СТУ с использованием вторых опорных блоков, идентифицируемых информацией движения для второй строки СТУ;

кодируют вторые один или более остаточных блоков для вторых одной или более СУ, причем вторые один или более остаточных блоков представляют соответственные разности между вторыми одним или более блоками предсказания и вторыми одной или более СУ; и

после сброса второго буфера MVP истории, сохраняют информацию движения для второй строки СТУ упомянутого изображения во втором буфере MVP истории, причем вторая строка СТУ отличается от первой строки СТУ, включая сохранение вторых одного или более векторов движения для второй строки СТУ для использования в качестве вторых предикторов векторов движения, чтобы предсказывать вторые один или более других векторов движения упомянутого изображения.

2. Способ по п.1, в котором второй буфер MVP истории отличается от первого буфера MVP истории, причем упомянутое сохранение информации о движении для первой строки СТУ содержит этап, на котором сохраняют, посредством первого подпроцесса процесса кодирования видео, информацию движения для первой строки СТУ, и упомянутое сохранение информации движения для второй строки СТУ содержит этап, на котором сохраняют, посредством второго подпроцесса процесса кодирования видео, информацию движения для второй строки СТУ, причем первый подпроцесс отличается от второго подпроцесса.

3. Способ по п.1, в котором второй буфер MVP истории отличается от первого буфера MVP истории, при этом при упомянутом сбросе второго буфера MVP истории второй буфер MVP истории сбрасывают в ответ на обработку начала второй строки СТУ во время параллельной обработки волнового фронта.

4. Способ по п.1, в котором

при упомянутом сохранении информации движения для первой строки СТУ сохраняют только информацию движения, уникальную в пределах первого буфера MVP истории,

в первый буфер MVP истории, и

при упомянутом сохранении информации движения для второй строки STU сохраняют только информацию движения, уникальную в пределах второго буфера MVP истории, во второй буфер MVP истории.

5 5. Способ по п.1, дополнительно содержащий этапы, на которых:

когда первый буфер MVP истории становится заполненным, удаляют один или более самых ранних введенных векторов движения из первого буфера MVP истории в соответствии с правилом “первым пришел - первым обслужен” (FIFO); и

10 когда второй буфер MVP истории становится заполненным, удаляют один или более самых ранних введенных векторов движения из второго буфера MVP истории в соответствии с правилом FIFO.

6. Способ по п.1, в котором

при упомянутом сохранении информации движения для первой строки STU сохраняют информацию движения для каждого из множества разных типов моделей движения в 15 соответствующих разных буферах MVP истории из первого множества буферов MVP истории, включающего в себя первый буфер MVP истории; и

при упомянутом сохранении информации движения для второй строки STU сохраняют информацию движения для каждого из множества разных типов моделей движения в 20 соответствующих разных буферах MVP истории из второго множества буферов MVP истории, включающего в себя второй буфер MVP истории.

7. Способ по п.6, в котором упомянутое множество разных типов моделей движения содержит одно или более из аффинной модели движения, информации движения режима 25 внутриблочного копирования, информации движения компенсации локального освещения, подблочного предсказания векторов движения (MVP) и временного предсказания движения.

8. Способ по п.1, дополнительно содержащий этап, на котором генерируют синтетический MVP из двух или более MVP в одном или более буферах MVP истории.

9. Способ по п.8, в котором упомянутые два или более MVP имеют разные типы информации движения, содержащие две или более из кодированной информации 30 движения, информации движения аффинной модели движения, информации движения режима внутриблочного копирования, информации движения компенсации локального освещения, информации движения подблочного MVP, информации движения временного предиктора движения, информации синтетического вектора движения, основывающейся на информации движения MVP, информации движения пространственного MVP и 35 информации движения временного MVP.

10. Способ по п.1, в котором

упомянутое сохранение информации движения для первой строки STU дополнительно содержит этап, на котором предварительно заполняют первый буфер MVP истории 40 первой заранее определенной информацией движения, причем первая заранее определенная информация движения содержит первый нулевой вектор движения с первым индексом опорного кадра и первым направлением интер-предсказания, и

упомянутое сохранение информации движения для второй строки STU дополнительно содержит этап, на котором предварительно заполняют второй буфер MVP истории 45 второй заранее определенной информацией движения, причем вторая заранее определенная информация движения содержит второй нулевой вектор движения с вторым индексом опорного кадра и вторым направлением интер-предсказания.

11. Способ по п.1, в котором изображение, включающее в себя первую строку STU и вторую строку STU, является первым временным слоем, при этом

упомянутое сохранение информации движения для первой строки STU дополнительно содержит этап, на котором предварительно заполняют первый буфер MVP истории первой информацией движения из первых одного или более кодированных изображений в первом временном слое или одном и более временных слоях ниже, чем первый временной слой, и

упомянутое сохранение информации движения для второй строки STU дополнительно содержит этап, на котором предварительно заполняют второй буфер MVP истории второй информацией движения из вторых одного или более кодированных изображений в первом временном слое или одном и более временных слоях ниже, чем первый временной слой,

причем способ дополнительно содержит этап, на котором масштабируют по меньшей мере одну из первой информации движения и второй информации движения в соответствии с временными разностями.

12. Способ по п.1, дополнительно содержащий этапы, на которых:
инициализируют первый буфер MVP истории первым нулевым вектором движения, имеющим первый индекс опорного кадра и первое направление интер-предсказания;
и

инициализируют второй буфер MVP истории вторым нулевым вектором движения, имеющим второй индекс опорного кадра и второе направление интер-предсказания.

13. Способ по п.1, в котором вторая строка STU находится непосредственно ниже первой строки STU, причем способ дополнительно содержит этап, на котором инициализируют или модифицируют второй буфер MVP истории с использованием первого буфера MVP истории.

14. Способ по п.1, дополнительно содержащий этапы, на которых:
декодируют единицы кодирования (CU) единиц дерева кодирования (STU) первой строки STU с использованием первого буфера MVP истории; и

декодируют единицы кодирования (CU) единиц дерева кодирования (STU) второй строки STU с использованием второго буфера MVP истории.

15. Способ по п.1, дополнительно содержащий этапы, на которых:
кодируют единицы кодирования (CU) единиц дерева кодирования (STU) первой строки STU с использованием первого буфера MVP истории; и

кодируют единицы кодирования (CU) единиц дерева кодирования (STU) второй строки STU с использованием второго буфера MVP истории.

16. Способ по п.1, дополнительно содержащий этапы, на которых:
кодируют первую информацию движения единиц кодирования (CU) единиц дерева кодирования (STU) первой строки STU с использованием первого буфера MVP истории, причем кодирование первой информации движения содержит кодирование первой информации движения с использованием режима расширенного предсказания вектора движения (AMVP), режима слияния или аффинного режима; и

кодируют вторую информацию движения единиц кодирования (CU) единиц дерева кодирования (STU) второй строки STU с использованием второго буфера MVP истории, причем кодирование второй информации движения содержит кодирование второй информации движения с использованием режима AMVP, режима слияния или аффинного режима.

17. Способ по п.1, дополнительно содержащий этапы, на которых:
добавляют один или более векторов движения к первым одному или более векторам движения первой однонаправленной информации движения в первом буфере MVP истории для преобразования первой однонаправленной информации движения в первую

двунаправленную информацию движения, и

добавляют один или более векторов движения ко вторым одному или более векторам движения второй однонаправленной информации движения во втором буфере MVP истории для преобразования второй однонаправленной информации движения во

5 вторую двунаправленную информацию движения.

18. Способ по п.1, в котором

при упомянутом сохранении информации движения для первой строки STU сохраняют только те векторы движения, которые отличаются от других векторов движения в первом буфере MVP истории на пороговое значение, в первом буфере MVP истории, и

10 при упомянутом сохранении информации движения для второй строки STU сохраняют только те векторы движения, которые отличаются от других векторов движения во втором буфере MVP истории на пороговое значение, во втором буфере MVP истории.

19. Способ по п.1, в котором

при упомянутом сохранении информации движения для первой строки STU сохраняют 15 данные, указывающие, ассоциированы ли векторы движения информации движения для первой строки STU с кодированными в режиме слияния единицами кодирования (CU) или AMVP-кодированными CU, и

при упомянутом сохранении информации движения для второй строки STU сохраняют 20 данные, указывающие, ассоциированы ли векторы движения информации движения для второй строки STU с кодированными в режиме слияния CU или AMVP-кодированными CU.

20. Способ по п.1, в котором первая строка STU имеет длину, меньшую полной ширины изображения.

21. Способ по п.1, в котором при упомянутом сохранении информации движения 25 для первой строки STU информацию движения сохраняют для количества CU в первой строке STU, которое меньше общего количества CU в первой строке STU.

22. Устройство для кодирования данных видео, содержащее:

память, приспособленную хранить данные видео; и

30 один или более модулей обработки, реализованных в схемах и выполненных с возможностью:

кодировать информацию движения для первой строки единиц дерева кодирования (STU) изображения, причем информация движения для первой строки STU включает в себя первые один или более векторов движения для первой строки STU;

35 формировать первые один или более блоков предсказания для первых одной или более единиц кодирования (CU) первой строки STU с использованием первых опорных блоков, идентифицируемых информацией движения для первой строки STU;

кодировать первые один или более остаточных блоков для первых одной или более CU, причем первые один или более остаточных блоков представляют соответственные 40 разности между первыми одним или более блоками предсказания и первыми одной или более CU;

сохранять информацию движения для первой строки STU упомянутого изображения в первом буфере предиктора векторов движения, основывающегося на истории (MVP истории), в памяти, причем информация движения для первой строки STU включает в 45 себя первые один или более векторов движения для первой строки STU для использования в качестве первых предикторов векторов движения, чтобы предсказывать первые один или более других векторов движения упомянутого изображения;

сбрасывать второй буфер MVP истории в памяти;

кодировать информацию движения для второй строки STU упомянутого изображения,

причем информация движения для второй строки СТU включает в себя вторые один или более векторов движения для второй строки СТU;

формировать вторые один или более блоков предсказания для вторых одной или более СУ второй строки СТU с использованием вторых опорных блоков, идентифицируемых информацией движения для второй строки СТU;

кодировать вторые один или более остаточных блоков для вторых одной или более СУ, причем вторые один или более остаточных блоков представляют соответственные разности между вторыми одним или более блоками предсказания и вторыми одной или более СУ; и

после сброса второго буфера MVP истории, сохранять информацию движения для второй строки СТU упомянутого изображения во втором буфере MVP истории, причем вторая строка СТU отличается от первой строки СТU, при этом информация движения для второй строки СТU включает в себя вторые один или более векторов движения для второй строки СТU для использования в качестве вторых предикторов векторов движения, чтобы предсказывать вторые один или более других векторов движения упомянутого изображения.

23. Устройство по п.22, при этом второй буфер MVP истории отличается от первого буфера MVP истории, причем один или более модулей обработки исполняют первый подпроцесс процесса кодирования видео, чтобы сохранять информацию движения для первой строки СТU, и один или более модулей обработки исполняют второй подпроцесс процесса кодирования видео, чтобы сохранять информацию движения для второй строки СТU, при этом первый подпроцесс отличается от второго подпроцесса.

24. Устройство по п.22, при этом второй буфер MVP истории отличается от первого буфера MVP истории, причем чтобы сбросить второй буфер MVP истории один или более модулей обработки выполнены с возможностью сбрасывать второй буфер MVP истории в ответ на обработку начала второй строки СТU во время параллельной обработки волнового фронта.

25. Устройство по п.22, в котором один или более модулей обработки выполнены с возможностью сохранять только информацию движения, уникальную в пределах первого буфера MVP истории, в первом буфере MVP истории, и

один или более модулей обработки выполнены с возможностью сохранять только информацию движения, уникальную в пределах второго буфера MVP истории, во втором буфере MVP истории.

26. Устройство по п.22, в котором один или более модулей обработки дополнительно выполнены с возможностью:

когда первый буфер MVP истории становится заполненным, удалять один или более самых ранних введенных векторов движения из первого буфера MVP истории в соответствии с правилом “первым пришел - первым обслужен” (FIFO); и

когда второй буфер MVP истории становится заполненным, удалять один или более самых ранних введенных векторов движения из второго буфера MVP истории в соответствии с правилом FIFO.

27. Устройство по п.22, в котором

один или более модулей обработки выполнены с возможностью сохранять информацию движения для каждого из множества разных типов моделей движения в соответствующих разных буферах MVP истории из первого множества буферов MVP истории, включающего в себя первый буфер MVP истории; и

один или более модулей обработки выполнены с возможностью сохранять

информацию движения для каждого из множества разных типов моделей движения в соответствующих разных буферах MVP истории из второго множества буферов MVP истории, включающего в себя второй буфер MVP истории.

28. Устройство по п.27, при этом упомянутое множество разных типов моделей движения содержит одно или более из аффинной модели движения, информации движения режима внутриблочного копирования, информации движения компенсации локального освещения, подблочного предсказания вектора движения (MVP) и временного предсказания движения.

29. Устройство по п.22, в котором один или более модулей обработки дополнительно выполнены с возможностью генерировать синтетический MVP из двух или более MVP в одном или более буферах MVP истории.

30. Устройство п.29, при этом два или более MVP имеют разные типы информации движения, содержащие две или более из кодированной информации движения, информации движения аффинной модели движения, информации движения режима внутриблочного копирования, информации движения компенсации локального освещения, информации движения подблочного MVP, информации движения временного предиктора движения, информации синтетического вектора движения, основывающейся на информации движения MVP, информации движения пространственного MVP и информации движения временного MVP.

31. Устройство по п.22, дополнительно содержащее устройство отображения, выполненное с возможностью отображения данных видео.

32. Устройство по п.22, дополнительно содержащее камеру, выполненную с возможностью захвата данных видео.

33. Устройство по п.22, причем устройство содержит одно или более из камеры, компьютера, мобильного устройства, устройства вещательного приемника или телевизионной приставки.

34. Устройство п.22, причем устройство содержит по меньшей мере одно из: интегральной схемы; микропроцессора и устройства беспроводной связи.

35. Устройство по п.22, при этом первая строка STU имеет длину, меньшую полной ширины изображения.

36. Устройство по п.22, при этом чтобы сохранять информацию движения для первой строки STU один или более модулей обработки выполнены с возможностью сохранять в памяти информацию движения для количества CU в первой строке STU, которое меньше общего количества CU в первой строке STU.

37. Машиночитаемый носитель данных, в котором сохранены инструкции, которые при их исполнении предписывают процессору:

кодировать информацию движения для первой строки единиц дерева кодирования (STU) изображения, причем информация движения для первой строки STU включает в себя первые один или более векторов движения для первой строки STU;

формировать первые один или более блоков предсказания для первых одной или более единиц кодирования (CU) первой строки STU с использованием первых опорных блоков, идентифицируемых информацией движения для первой строки STU;

кодировать первые один или более остаточных блоков для первых одной или более CU, причем первые один или более остаточных блоков представляют соответственные разности между первыми одним или более блоками предсказания и первыми одной или более CU;

сохранять информацию движения для первой строки СТU упомянутого изображения в первом буфере предиктора векторов движения, основывающегося на истории (MVP истории), в памяти, причем информация движения для первой строки СТU включает в себя первые один или более векторов движения для первой строки СТU для

5 использования в качестве первых предикторов векторов движения, чтобы предсказывать первые один или более других векторов движения упомянутого изображения;

сбрасывать второй буфер MVP истории в памяти;

кодировать информацию движения для второй строки СТU упомянутого изображения, причем информация движения для второй строки СТU включает в себя вторые один

10 или более векторов движения для второй строки СТU;

формировать вторые один или более блоков предсказания для вторых одной или более СУ второй строки СТU с использованием вторых опорных блоков, идентифицируемых информацией движения для второй строки СТU;

15 кодировать вторые один или более остаточных блоков для вторых одной или более СУ, причем вторые один или более остаточных блоков представляют соответственные разности между вторыми одним или более блоками предсказания и вторыми одной или более СУ; и

после сброса второго буфера MVP истории, сохранять информацию движения для второй строки СТU упомянутого изображения во втором буфере MVP истории, причем

20 вторая строка СТU отличается от первой строки СТU, при этом информация движения для второй строки СТU включает в себя вторые один или более векторов движения для второй строки СТU для использования в качестве вторых предикторов векторов движения.

38. Машиночитаемый носитель данных по п.37, при этом второй буфер MVP истории

25 отличается от первого буфера MVP истории, причем инструкции, которые предписывают процессору сохранять информацию движения для первой строки СТU, содержат инструкции, которые предписывают процессору исполнять первый подпроцесс процесса кодирования видео, чтобы сохранять информацию движения для первой строки СТU, и при этом инструкции, которые предписывают процессору сохранять информацию

30 движения для второй строки СТU, содержат инструкции, которые предписывают процессору исполнять второй подпроцесс процесса кодирования видео, чтобы сохранять информацию движения для второй строки СТU, причем первый подпроцесс отличается от второго подпроцесса.

39. Машиночитаемый носитель данных по п.37, причем второй буфер MVP истории

35 отличается от первого буфера MVP истории, при этом инструкции, которые предписывают процессору сбрасывать второй буфер MVP истории, содержат инструкции, которые предписывают процессору сбрасывать второй буфер MVP истории в ответ на обработку начала второй строки СТU во время параллельной обработки волнового фронта.

40. Машиночитаемый носитель данных по п.37, при этом

инструкции, которые предписывают процессору сохранять информацию движения для первой строки СТU, содержат инструкции, которые предписывают процессору сохранять только информацию движения, уникальную в пределах первого буфера MVP истории, в первом буфере MVP истории, и

45 инструкции, которые предписывают процессору сохранять информацию движения для второй строки СТU, содержат инструкции, которые предписывают процессору сохранять только информацию движения, уникальную в пределах второго буфера MVP истории, во втором буфере MVP истории.

41. Машиночитаемый носитель данных по п.37, дополнительно содержащий инструкции, которые предписывают процессору:

когда первый буфер MVP истории становится заполненным, удалять один или более самых ранних введенных векторов движения из первого буфера MVP истории в соответствии с правилом “первым пришел - первым обслужен” (FIFO); и

когда второй буфер MVP истории становится заполненным, удалять один или более самых ранних введенных векторов движения из второго буфера MVP истории в соответствии с правилом FIFO.

42. Машиночитаемый носитель данных по п.37, в котором

инструкции, которые предписывают процессору сохранять информацию движения для первой строки СТU, содержат инструкции, которые предписывают процессору сохранять информацию движения для каждого из множества разных типов моделей движения в соответствующих разных буферах MVP истории из первого множества буферов MVP истории, включающего в себя первый буфер MVP истории; и

инструкции, которые предписывают процессору сохранять информацию движения для второй строки СТU, содержат инструкции, которые предписывают процессору сохранять информацию движения для каждого из множества разных типов моделей движения в соответствующих разных буферах MVP истории из второго множества буферов MVP истории, включающего в себя второй буфер MVP истории.

43. Машиночитаемый носитель данных по п.42, при этом упомянутое множество разных типов моделей движения содержит одно или более из аффинной модели движения, информации движения режима внутриблочного копирования, информации движения компенсации локального освещения, подблочного предсказания вектора движения (MVP) и временного предсказания движения.

44. Машиночитаемый носитель данных по п.37, дополнительно содержащий инструкции, которые предписывают процессору генерировать синтетический MVP из двух или более MVP в одном или более буферах MVP истории.

45. Машиночитаемый носитель данных п.44, при этом упомянутые два или более MVP имеют разные типы информации движения, содержащие две или более из кодированной информации движения, информации движения аффинной модели движения, информации движения режима внутриблочного копирования, информации движения компенсации локального освещения, информации движения подблочного MVP, информации движения временного предиктора движения, информации синтетического вектора движения, основывающейся на информации движения MVP, информации движения пространственного MVP и информации движения временного MVP.

46. Машиночитаемый носитель данных по п.37, при этом первая строка СТU имеет длину, меньшую полной ширины изображения.

47. Машиночитаемый носитель данных по п.37, в котором инструкции, которые предписывают процессору сохранять информацию движения для первой строки СТU, содержат инструкции, которые предписывают процессору сохранять информацию движения для количества СУ в первой строке СТU, которое меньше общего количества СУ в первой строке СТU.

48. Устройство для кодирования данных видео, содержащее:

память, приспособленную хранить данные видео;

средство для кодирования информации движения для первой строки единиц дерева кодирования (СТU) изображения, причем информация движения для первой строки СТU включает в себя первые один или более векторов движения для первой строки

СТУ;

средство для формирования первых одного или более блоков предсказания для первых одной или более единиц кодирования (CU) первой строки СТУ с использованием первых опорных блоков, идентифицируемых информацией движения для первой строки СТУ;

средство для кодирования первых одного или более остаточных блоков для первых одной или более CU, причем первые один или более остаточных блоков представляют соответственные разности между первыми одним или более блоками предсказания и первыми одной или более CU;

средство для сохранения информации движения для первой строки СТУ упомянутого изображения в первом буфере предиктора векторов движения, основывающегося на истории (MVP истории), в памяти, включая средство для сохранения первых одного или более векторов движения для первой строки СТУ в первом буфере MVP истории для использования в качестве первых предикторов векторов движения, чтобы предсказывать первые один или более других векторов движения упомянутого изображения;

средство для сброса второго буфера MVP истории в памяти;

средство для кодирования информации движения для второй строки СТУ упомянутого изображения, причем информация движения для второй строки СТУ включает в себя вторые один или более векторов движения для второй строки СТУ;

средство для формирования вторых одного или более блоков предсказания для вторых одной или более CU второй строки СТУ с использованием вторых опорных блоков, идентифицируемых информацией движения для второй строки СТУ;

средство для кодирования вторых одного или более остаточных блоков для вторых одной или более CU, причем вторые один или более остаточных блоков представляют соответственные разности между вторыми одним или более блоками предсказания и вторыми одной или более CU; и

средство для сохранения информации движения для второй строки СТУ упомянутого изображения во втором буфере MVP истории после сброса второго буфера MVP истории, причем вторая строка СТУ отличается от первой строки СТУ, включая средство для сохранения вторых одного или более векторов движения для второй строки СТУ во втором буфере MVP истории для использования в качестве вторых предикторов векторов движения, чтобы предсказывать вторые один или более других векторов движения упомянутого изображения.

49. Устройство по п.48, при этом второй буфер MVP истории отличается от первого буфера MVP истории, причем средство для сохранения информации движения для первой строки СТУ содержит первый подпроцесс процесса кодирования видео, и средство для сохранения информации движения для второй строки СТУ содержит второй подпроцесс процесса кодирования видео.

50. Устройство по п.48, при этом второй буфер MVP истории отличается от первого буфера MVP истории, причем средство для сброса второго буфера MVP истории содержит средство для сброса второго буфера MVP истории в ответ на обработку начала второй строки СТУ во время параллельной обработки волнового фронта.

51. Устройство по п.48, в котором

средство для сохранения информации движения для первой строки СТУ содержит средство для сохранения только информации движения, уникальной в пределах первого буфера MVP истории, в первом буфере MVP истории, и

средство для сохранения информации движения для второй строки СТУ содержит

средство для сохранения только информации движения, уникальной в пределах второго буфера MVP истории, во втором буфере MVP истории.

52. Устройство по п.48, дополнительно содержащее:

5 средство для удаления, когда первый буфер MVP истории становится заполненным, одного или более самых ранних введенных векторов движения из первого буфера MVP истории в соответствии с правилом “первым пришел - первым обслужен” (FIFO); и

средство для удаления, когда второй буфер MVP истории становится заполненным, одного или более самых ранних введенных векторов движения из второго буфера MVP истории в соответствии с правилом FIFO.

10 53. Устройство по п.48, в котором

средство для сохранения информации движения для первой строки STU содержит средство для сохранения информации движения для каждого из множества разных типов моделей движения в соответствующих разных буферах MVP истории из первого множества буферов MVP истории, включающего в себя первый буфер MVP истории;

15 и

средство для сохранения информации движения для второй строки STU содержит средство для сохранения информации движения для каждого из множества разных типов моделей движения в соответствующих разных буферах MVP истории второго множества буферов MVP истории, включающего в себя второй буфер MVP истории.

20 54. Устройство по п.53, при этом упомянутое множество разных типов моделей движения содержат одно или более из аффинной модели движения, информации движения режима внутриблочного копирования, информации движения компенсации локального освещения, подблочного предсказания вектора движения (MVP) и временного предсказания движения.

25 55. Устройство по п.54, дополнительно содержащее средство для генерации синтетического MVP из двух или более MVP в одном или более буферах MVP истории.

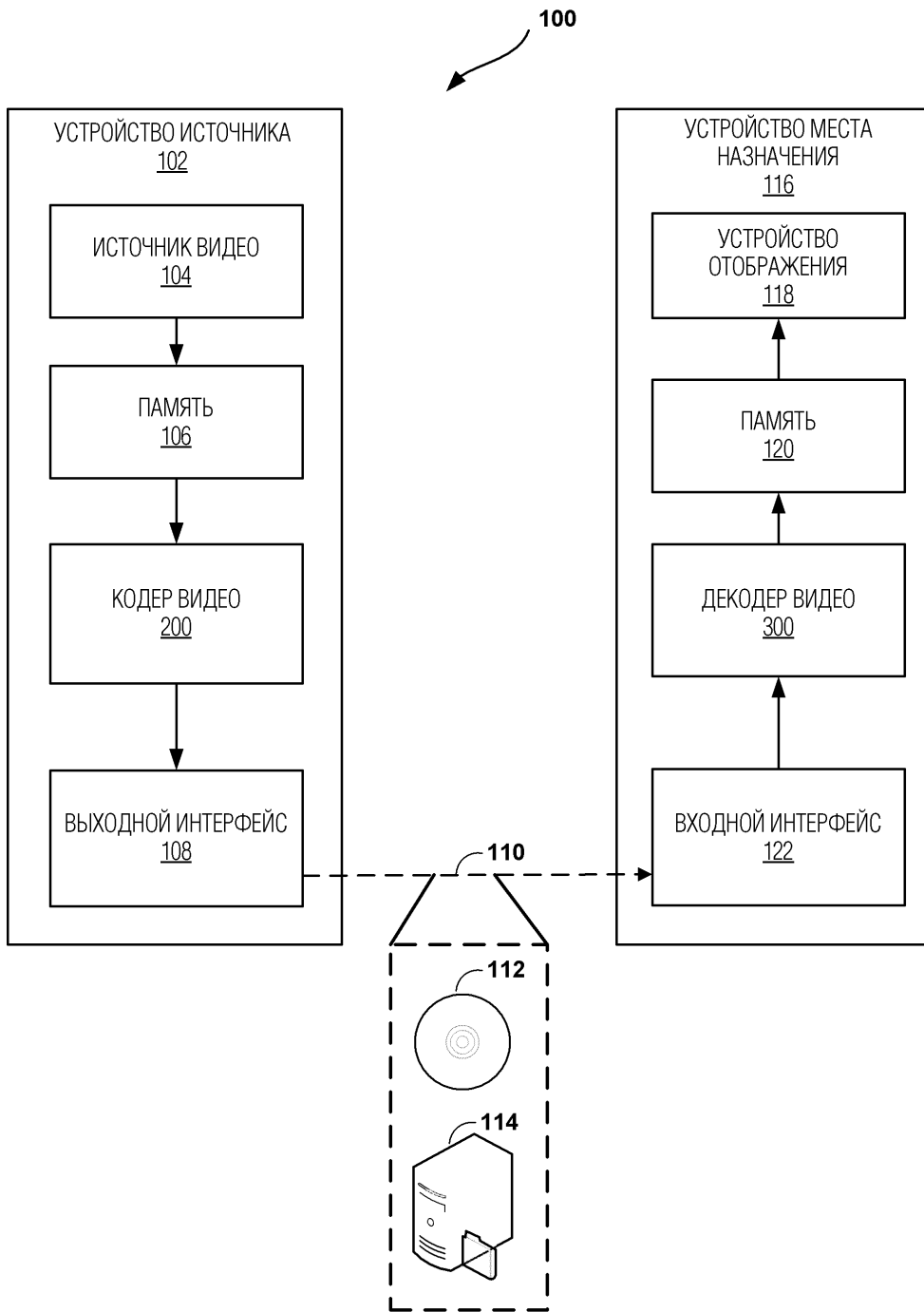
30 56. Устройство по п.55, при этом упомянутые два или более MVP имеют разные типы информации движения, содержащие две или более из кодированной информации движения, информации движения аффинной модели движения, информации движения режима внутриблочного копирования, информации движения компенсации локального освещения, информации движения подблочного MVP, информации движения временного предиктора движения, информации синтетического вектора движения, основывающейся на информации движения MVP, информации движения пространственного MVP и информации движения временного MVP.

35 57. Устройство по п.48, при этом первая строка STU имеет длину, меньшую полной ширины изображения.

40 58. Устройство по п.48, в котором средство для сохранения информации движения для первой строки STU содержит средство для сохранения информации движения для количества CU в первой строке STU, которое меньше общего количества CU в первой строке STU.

1

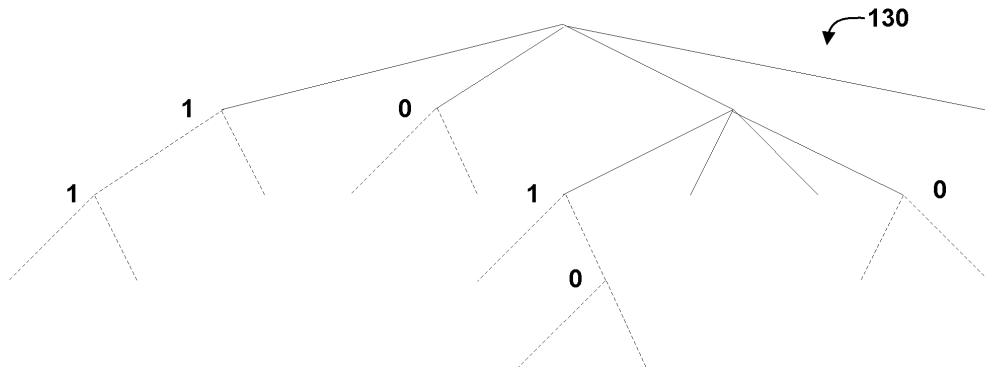
1/18



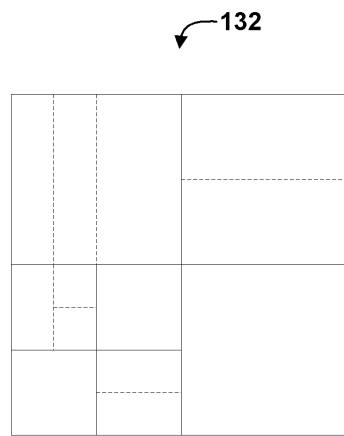
ФИГ. 1

2

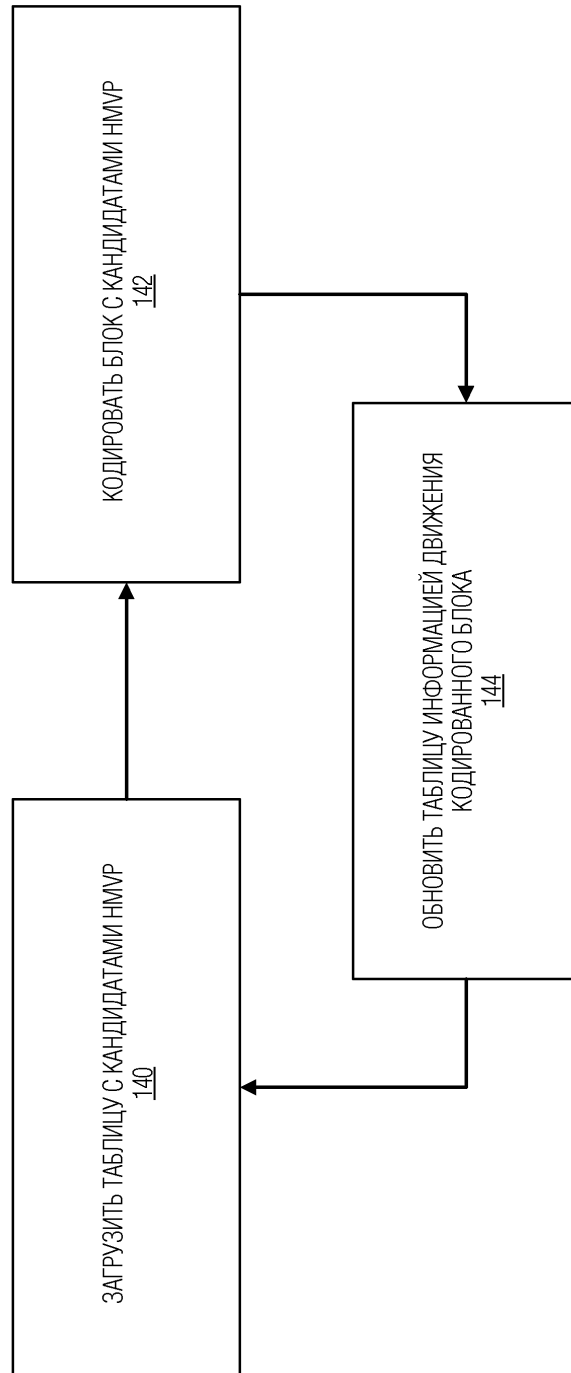
2/18



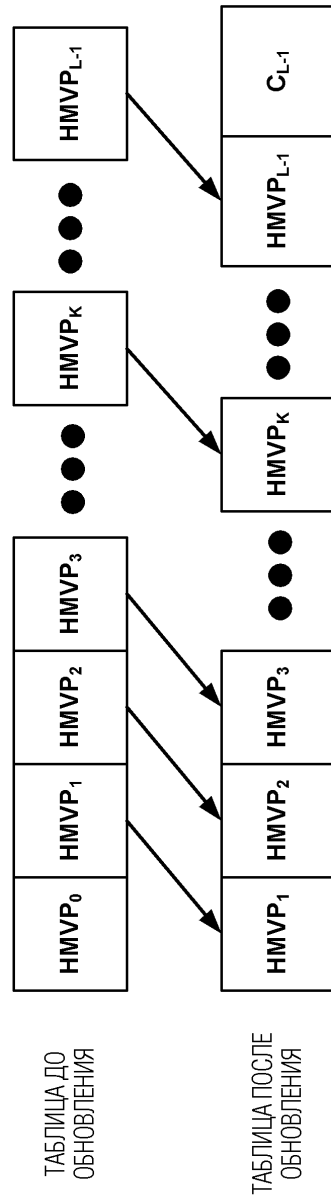
ФИГ. 2А



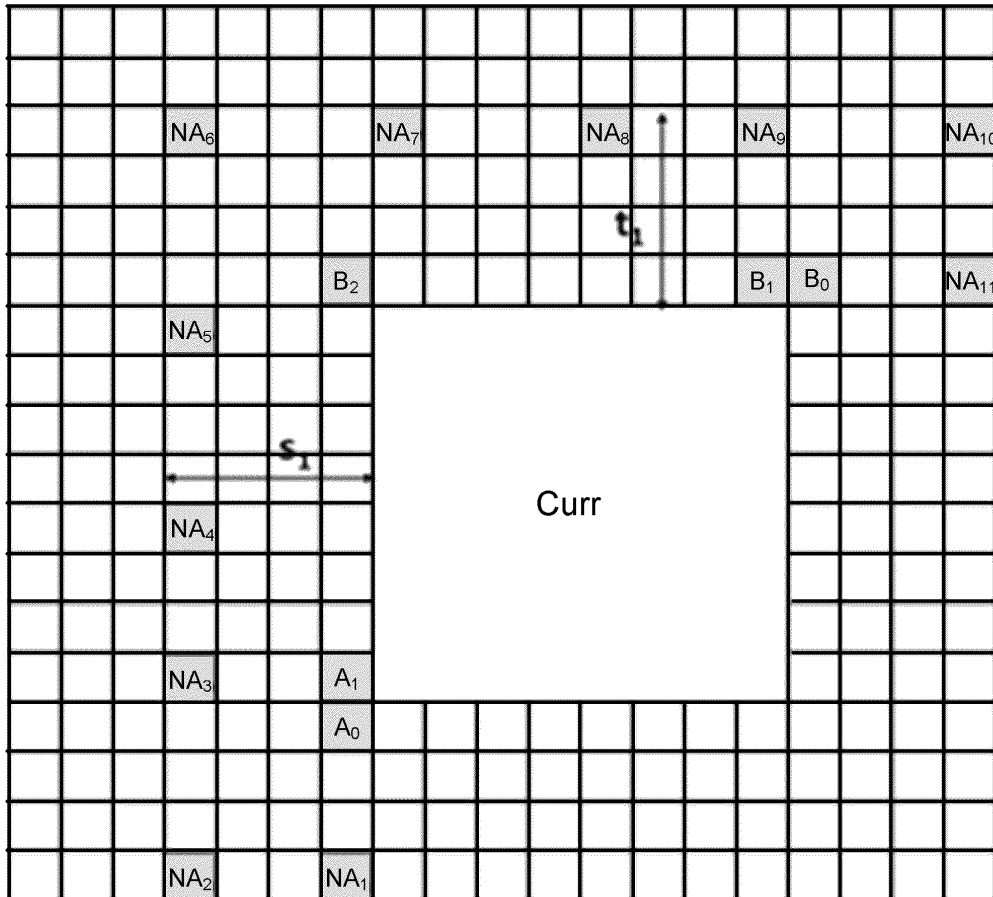
ФИГ. 2В



ФИГ. 3

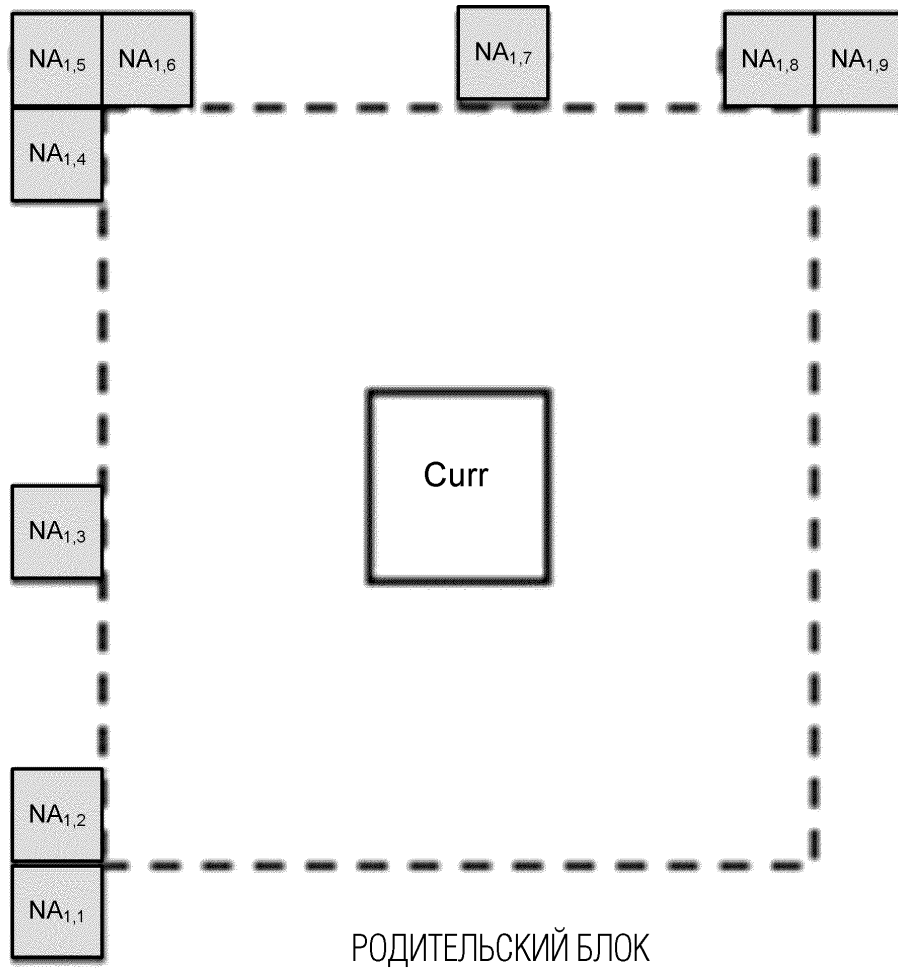


ФИГ. 4

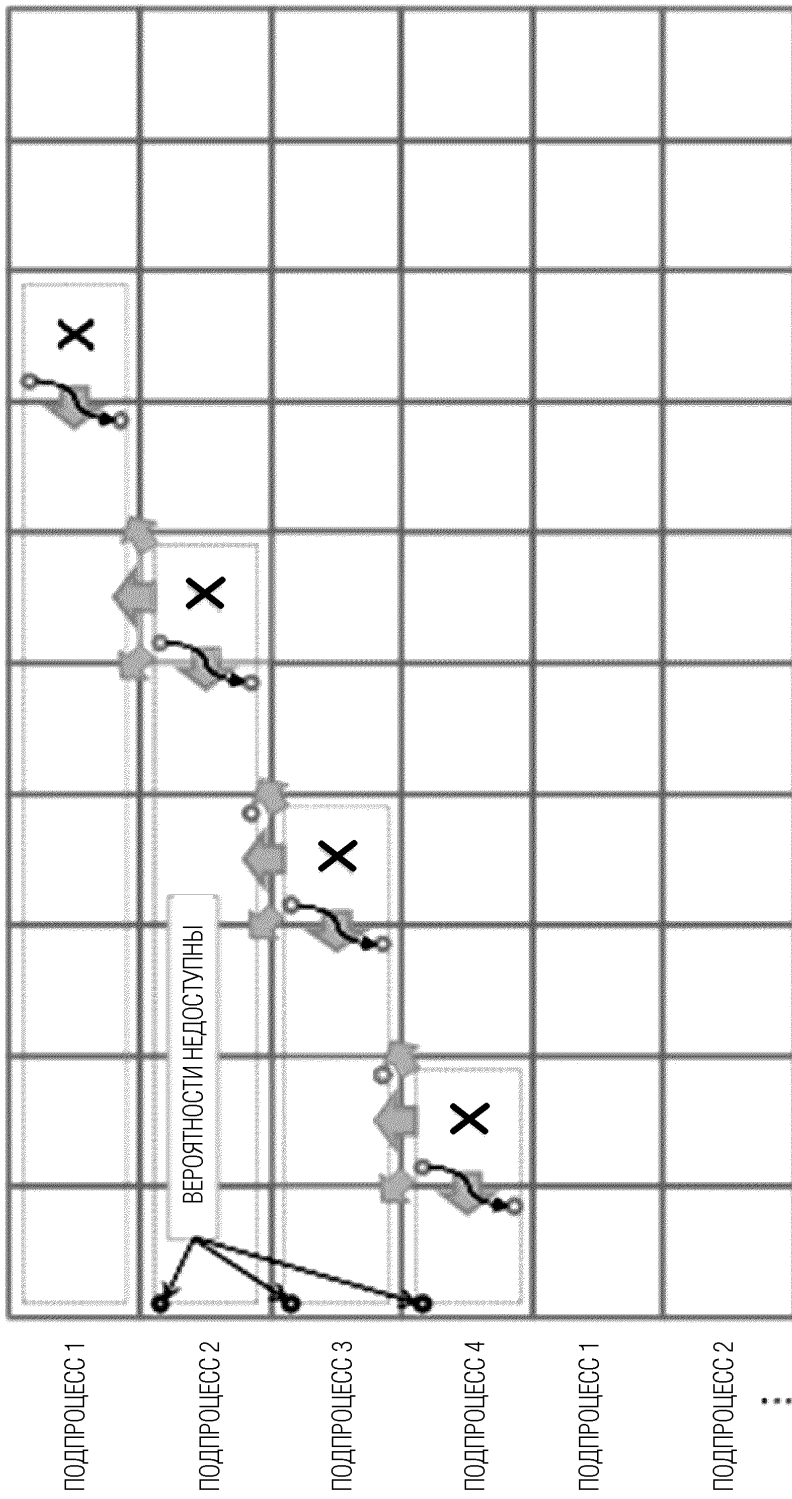


ФИГ. 5

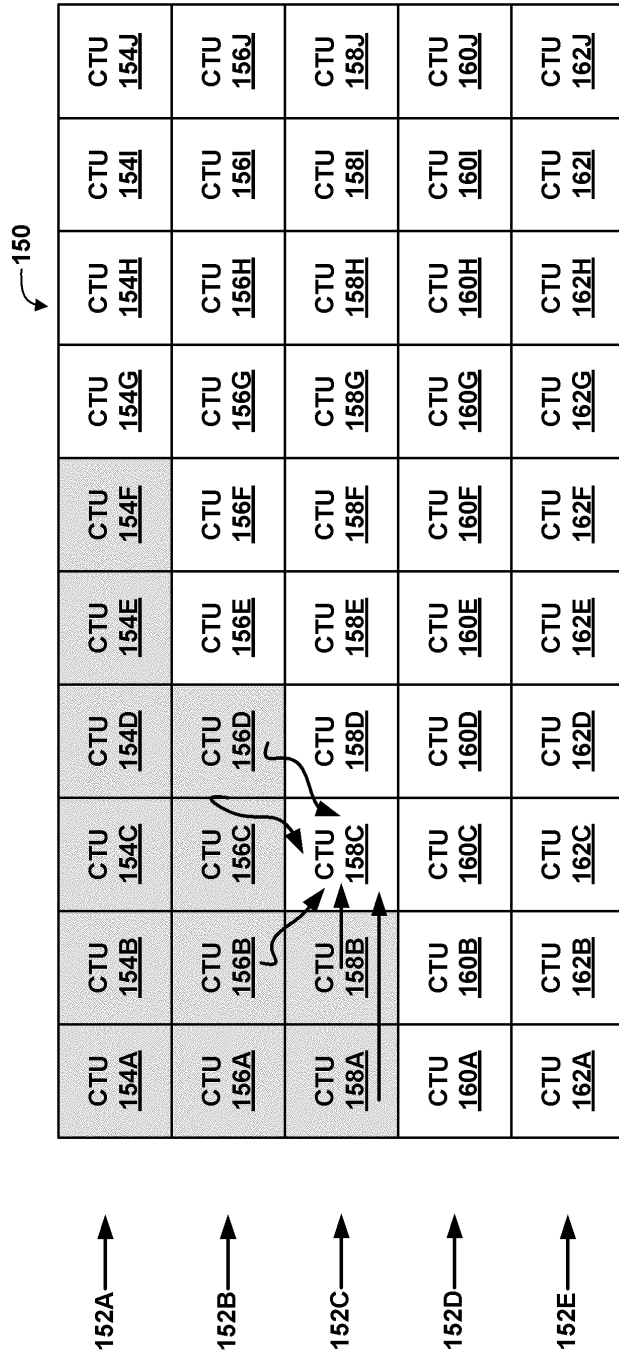
6/18



ФИГ. 6

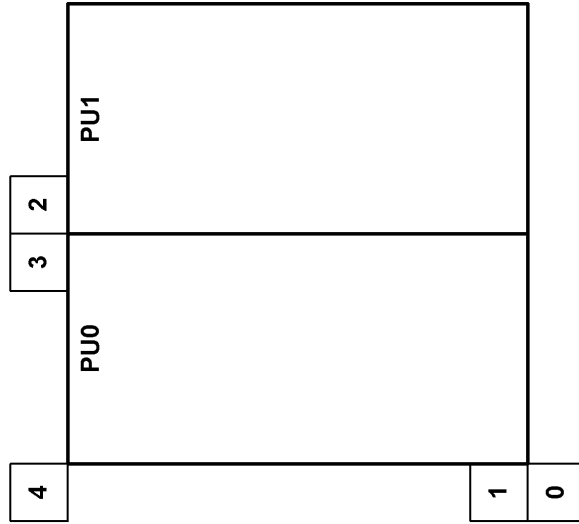


ФИГ. 7

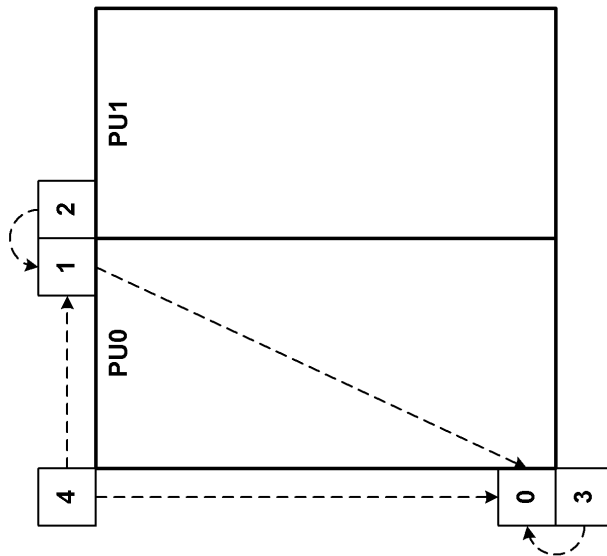


ФИГ. 9

10/18

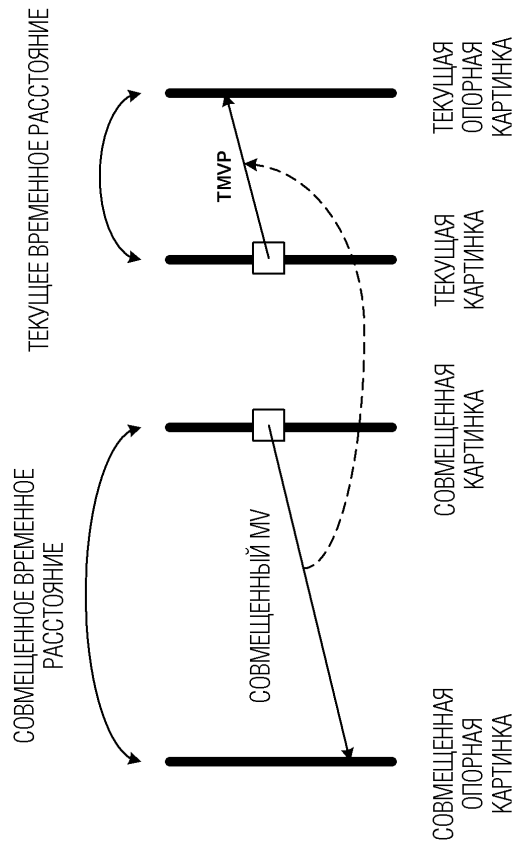


ФИГ. 10В

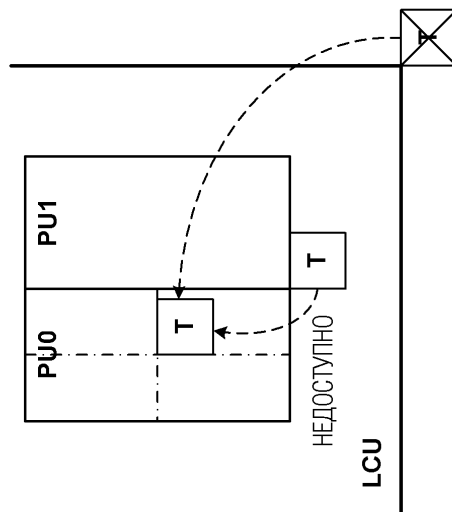


ФИГ. 10А

11/18

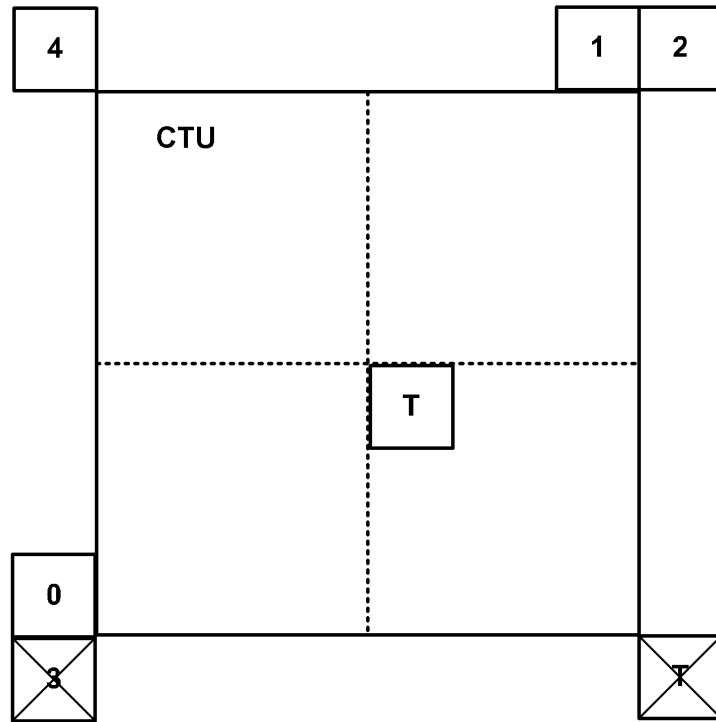


ФИГ. 11В



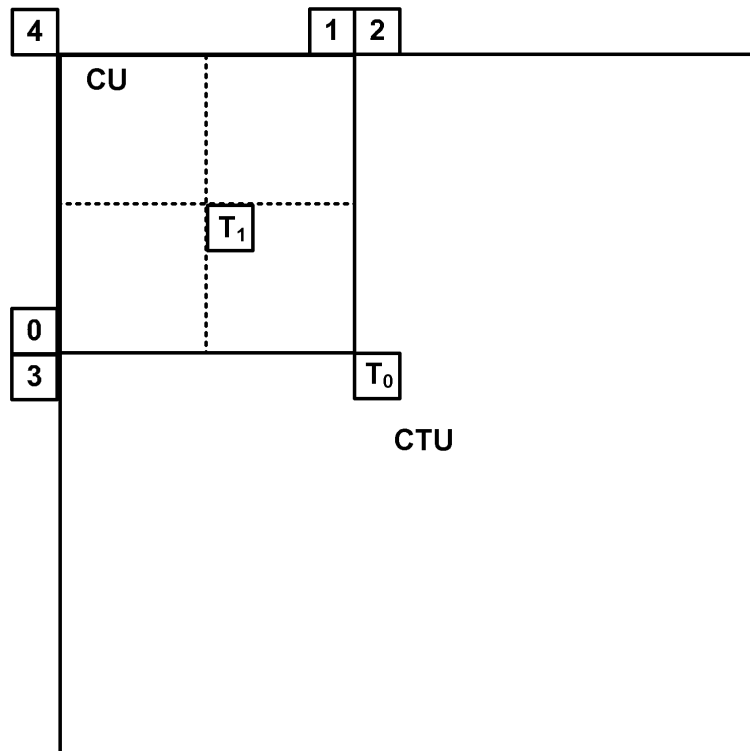
ФИГ. 11А

12/18



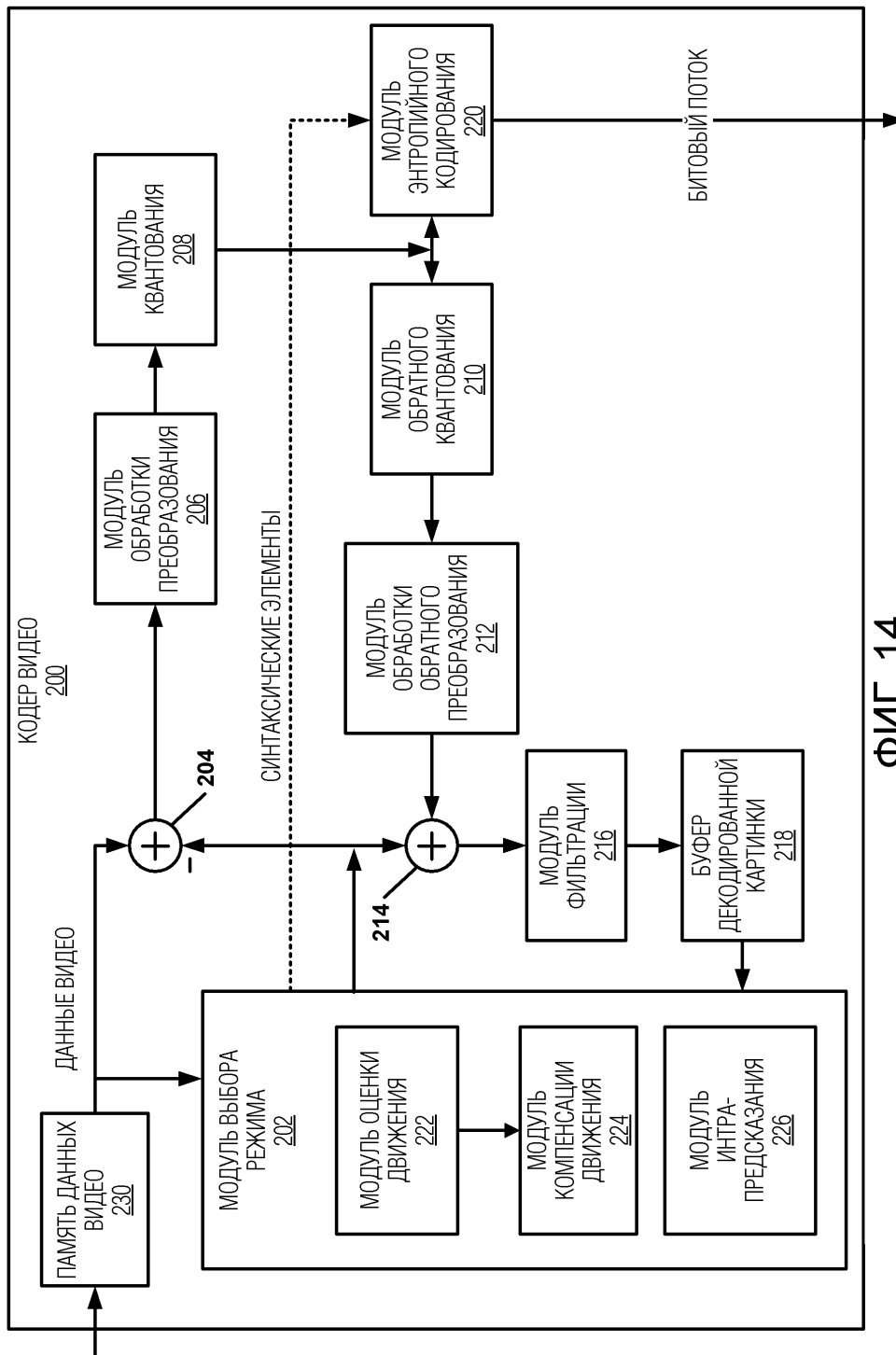
ФИГ. 12

13/18

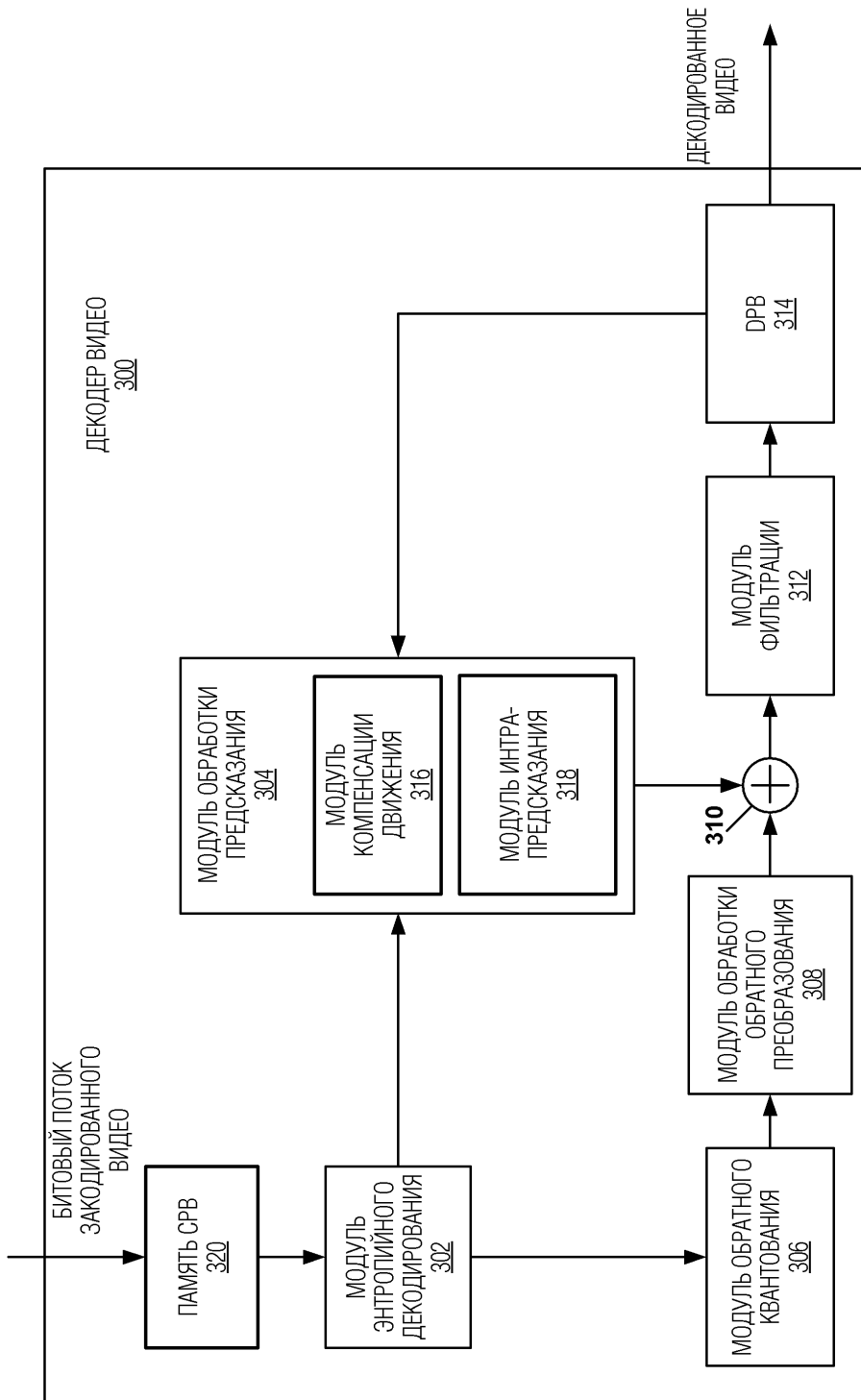


ФИГ. 13

14/18

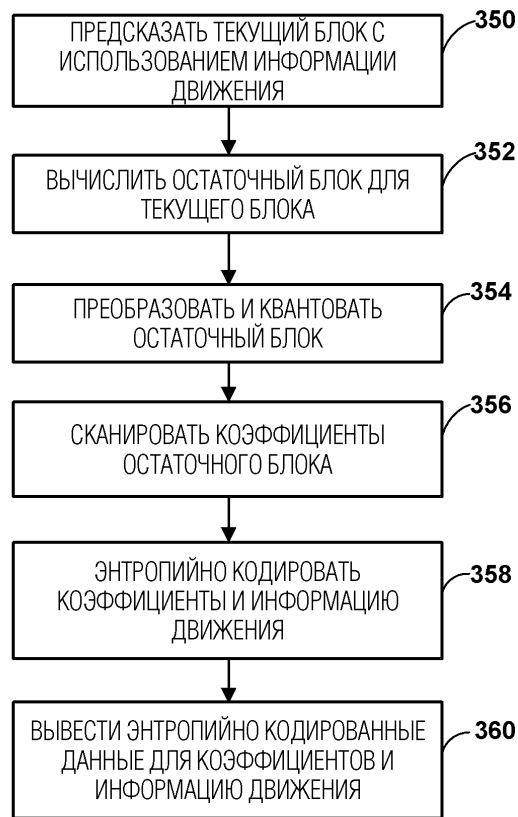


ФИГ. 14



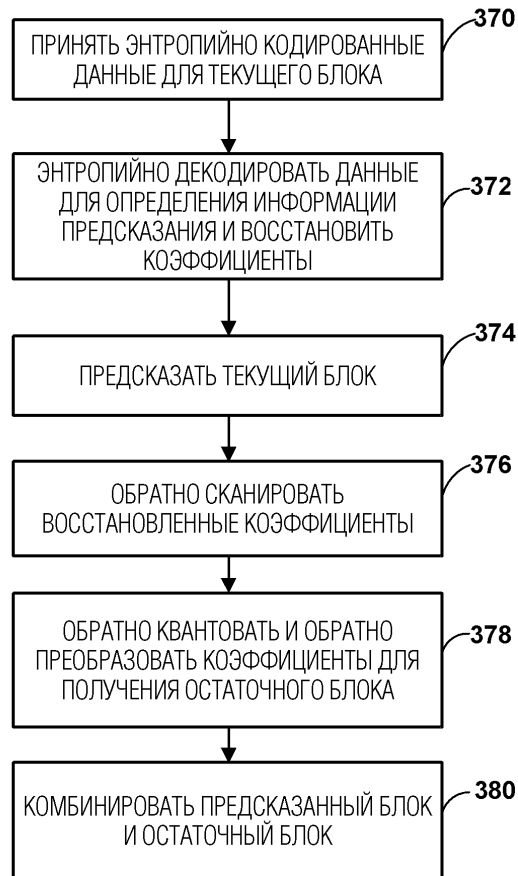
ФИГ. 15

16/18



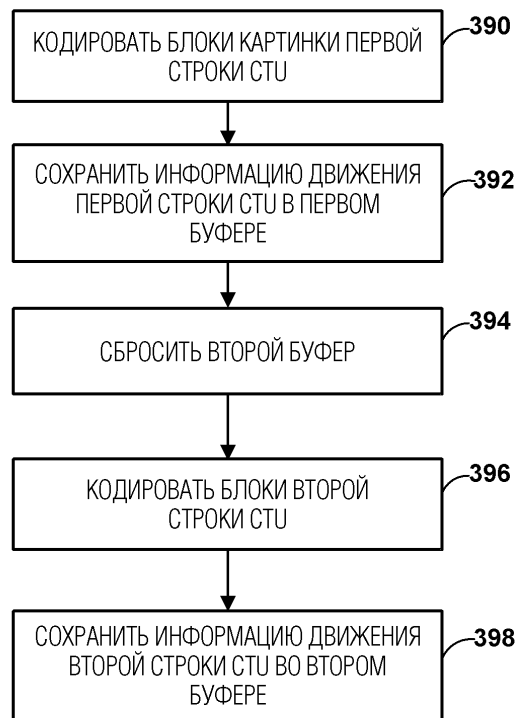
ФИГ. 16

17/18



ФИГ. 17

18/18



ФИГ. 18