



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

H04N 19/50 (2013.01); H04N 19/105 (2013.01); H04N 19/176 (2013.01); H04N 19/593 (2013.01); H04N 19/44 (2013.01); H04N 19/61 (2013.01); H04N 19/70 (2013.01); H04N 19/91 (2013.01); G06T 9/00 (2013.01)

(21)(22) Заявка: 2016126179, 22.12.2014

(24) Дата начала отсчета срока действия патента:
22.12.2014

Дата регистрации:
28.03.2019

Приоритет(ы):

(30) Конвенционный приоритет:
03.01.2014 US 61/923,628;
21.03.2014 US 14/222,580

(45) Опубликовано: 28.03.2019 Бюл. № 10

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 30.06.2016

(86) Заявка РСТ:
US 2014/071780 (22.12.2014)

(87) Публикация заявки РСТ:
WO 2015/102975 (09.07.2015)

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

(72) Автор(ы):

ЧЖУ Лихуа (US),
САЛЛИВАН Гари Дж. (US),
ВУ Юнцзюнь (US)

(73) Патентообладатель(и):

МАЙКРОСОФТ ТЕКНОЛОДЖИ
ЛАЙСЕНСИНГ, ЭлЭлСи (US)

(56) Список документов, цитированных в отчете
о поиске: D. FLYNN, "BoG report on range
extensions topics", Joint Collaborative Team
on Video Coding (JCT-VC) of ITU-T SG16
WP3 and ISO/IEC JTC1/SC29/WG11, 15th
Meeting: Geneva, CH, 23 Oct. - 1 Nov. 2013,
Document: JCTVC-O0352, 49 л., опубл.
01.11.2013, с. 8-15. CHAO PANG et al., "AhG5:
Intra block copying with padding", Joint
Collaborative Team on Video (см. прод.)

(54) НОВОВВЕДЕНИЯ В ПРЕДСКАЗАНИЕ БЛОЧНЫХ ВЕКТОРОВ И ОЦЕНКУ
ВОССТАНОВЛЕННЫХ ЗНАЧЕНИЙ ОТСЧЕТОВ В ОБЛАСТИ ПЕРЕКРЫТИЯ

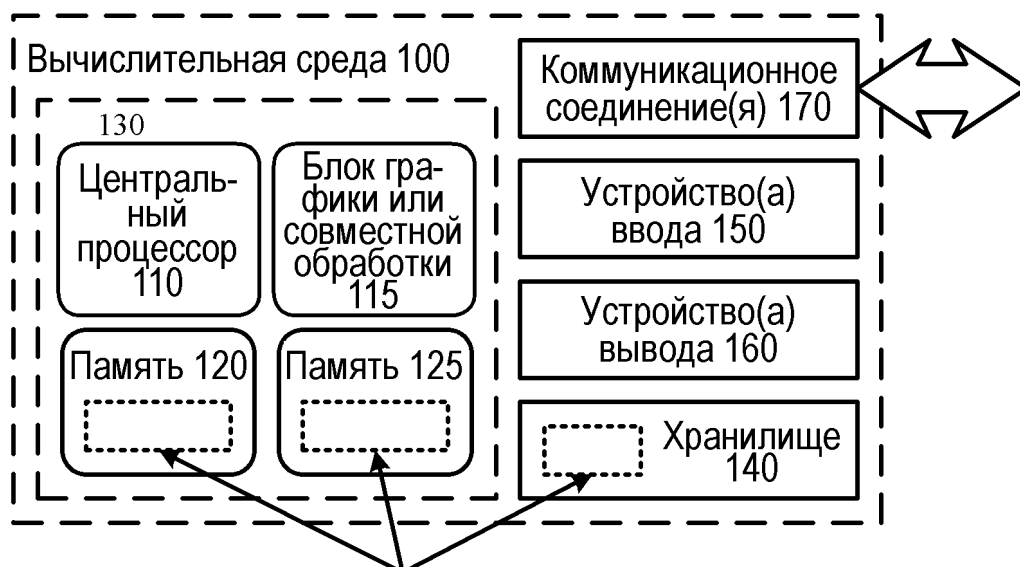
(57) Реферат:

Группа изобретений относится к кодированию и декодированию данных. Технический результат заключается в расширении арсенала средств. Для этого применяется способ кодирования данных, выполняемый в вычислительном устройстве, которое реализует видеокодер или кодер изображений, также используется вычислительное устройство, выполненное с возможностью кодирования данных и машиночитаемый носитель, на котором сохранены машиноисполняемые команды для предписания

вычислительному устройству, запрограммированному ими, выполнять операции. Некоторые из предложенных нововведений относятся к путям оценки восстановленных значений отсчетов в области перекрытия текущего блока во время оценки блочного вектора («BV»). Другие нововведения относятся к предсказанию значений BV во время кодирования или декодирования с использованием так называемого «пинг-понгового» подхода, согласно которому кодер

или декодер выбирает одно из пары пригодных значений BV при предсказании значения BV для

блока. 3 н. и 17 з.п. ф-лы, 27 ил.



Программное обеспечение 180, реализующее одно или несколько нововведений для внутреннего предсказания с блочным ко-пированием

ФИГ. 1

(56) (продолжение):

Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 15th Meeting: Geneva, CH, 23 Oct. - 1 Nov. 2013, Document: JCTVC-O0157, 6 л., опубли. 24.10.2013. J. SOLE et al., "HEVC Range Extensions Core Experiment 3 (RCE3): Intra block copy refinement", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 15th Meeting: Geneva, CH, 23 Oct. - 1 Nov. 2013, Document: JCTVC-O1123, 6 л., опубли. 20.11.2013. US 2013/0016777 A1, 17.01.2013. US 2013/0188867 A1, 25.07.2013. US 2008/0063080 A1, 13.03.2008. RU 2335859 C2, 10.10.2008.



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.

H04N 19/50 (2014.01)

H04N 19/105 (2014.01)

H04N 19/176 (2014.01)

(12) ABSTRACT OF INVENTION

(52) CPC

H04N 19/50 (2013.01); H04N 19/105 (2013.01); H04N 19/176 (2013.01); H04N 19/593 (2013.01); H04N 19/44 (2013.01); H04N 19/61 (2013.01); H04N 19/70 (2013.01); H04N 19/91 (2013.01); G06T 9/00 (2013.01)

(21)(22) Application: 2016126179, 22.12.2014

(24) Effective date for property rights:
22.12.2014

Registration date:
28.03.2019

Priority:

(30) Convention priority:
03.01.2014 US 61/923,628;
21.03.2014 US 14/222,580

(45) Date of publication: 28.03.2019 Bull. № 10

(85) Commencement of national phase: 30.06.2016

(86) PCT application:
US 2014/071780 (22.12.2014)

(87) PCT publication:
WO 2015/102975 (09.07.2015)

Mail address:

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

(72) Inventor(s):

CHZHU Likhua (US),
SALLIVAN Gari Dzh. (US),
VU Yuntszyun (US)

(73) Proprietor(s):

MAJKROSOFT TEKNOLODZHI
LAJSENSING, EIEISi (US)

(54) INNOVATIONS IN BLOCK VECTOR PREDICTION AND ESTIMATION OF RECONSTRUCTED SAMPLE VALUES WITHIN AN OVERLAP AREA

(57) Abstract:

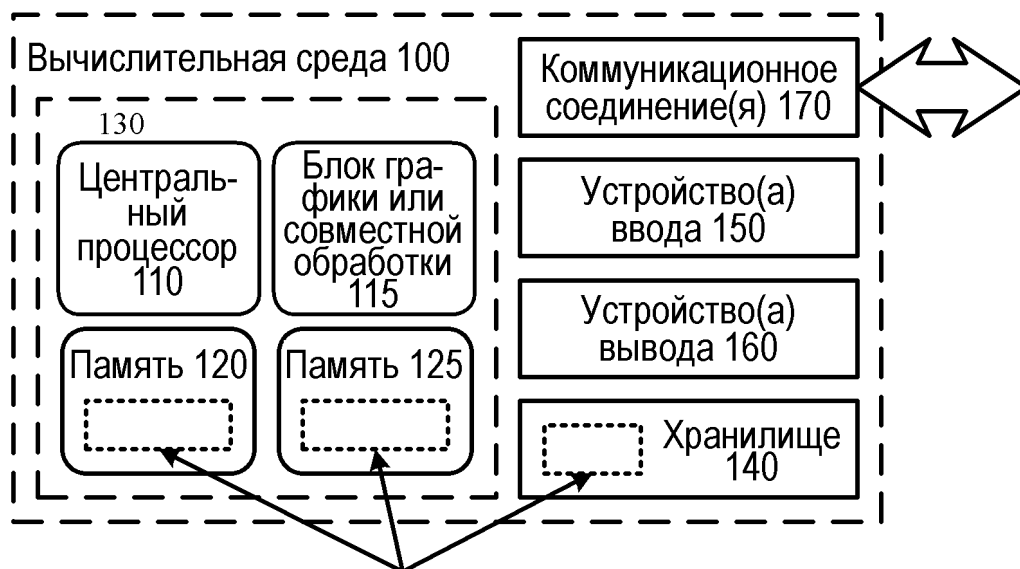
FIELD: electrical communication engineering.

SUBSTANCE: group of inventions relates to encoding and decoding data. For this purpose, a data encoding method implemented in a computing device that implements a video encoder or image encoder is used, computing device configured to encode data and a computer-readable media storing computer-executable instructions for causing a computing device programmed thereby to perform operations are also used. Some of the innovations relate to estimation of

reconstructed sample values within an overlap area of a current block during block vector ("BV") estimation. Other innovations relate to prediction of block vector ("BV") values during encoding or decoding using "ping-pong" approaches, in which an encoder or decoder selects between a pair of candidate BV values when predicting the BV value for a block.

EFFECT: technical result is wider range of means.

20 cl, 27 dwg



Программное обеспечение 180, реализующее одно или несколько нововведений для внутреннего предсказания с блочным ко-пированием

ФИГ. 1

УРОВЕНЬ ТЕХНИКИ

Инженеры используют сжатие (также называемое кодированием в источнике или кодированием в источнике) для уменьшения битрейта цифрового видео. Сжатие снижает затраты на хранение и передачу видеoinформации путем преобразования этой информации в форму с меньшим битрейтом. Распаковка (также называемая декодированием) восстанавливает версию исходной информации из сжатой формы. Система кодера/декодера представляет собой так называемый «кодек».

В последнее двадцатилетие были приняты различные стандарты кодеков для видео, в том числе стандарты ITU-T H.261, H.262 (MPEG-2 или ISO/IEC 13818-2), H.263 и H.264 (MPEG-4 AVC или ISO/IEC 14496-10), стандарты MPEG-1 (ISO/IEC 11172-2) и стандарты MPEG-4 Visual (ISO/IEC 14496-2) и стандарт SMPTE 421M (VC-1). Недавно был одобрен стандарт HEVC (ITU-T H.265 или ISO/IEC 23008-2). В настоящее время разрабатываются расширения к стандарту HEVC (например, для масштабируемого видеокодирования/ видеодекодирования для кодирования/декодирования видео с более высоким качеством в контексте глубины отсчета в битах или частоты дискретизации цветности, либо для мультимедийного кодирования/декодирования. Стандарт видеокодека, как правило, определяет опции для синтаксиса кодированного видеопотока, детализируя параметров в битовом потоке при использовании конкретных признаков при кодировании и декодировании. Во многих случаях стандарт видеокодека также обеспечивает детали, касающиеся операций декодирования, которые должен выполнять декодер для достижения подходящих результатов при декодировании. Помимо стандартов кодека различные патентованные форматы кодека определяют другие опции для синтаксиса кодированного битового потока видео и соответствующие операции декодирования.

Внутреннее блочное копирование («BC») представляет режим предсказания, рассматриваемый для расширений стандарта HEVC. Для режима внутреннего предсказания с BC значения отсчетов текущего блока картинки предсказывают, используя ранее восстановленные значения отсчетов в той же картинке. Блочный вектор («BV») указывает смещение от текущего блока до области картинки, которая включает в себя ранее восстановленные значения отсчетов, использованные для предсказания. Вектор BV передается в качестве сигнализации в битовом потоке. Внутреннее предсказание с BC является формой предсказания внутри картинки, то есть, при внутреннем предсказании с BC для блока картинки не используются какие-либо значения отсчетов, отличные от значений отсчетов в этой картинке.

Как в настоящее время определено в предварительных расширениях к стандарту HEVC и реализовано в некотором эталонном программном обеспечении для предварительных расширений к стандарту HEVC, режим внутреннего предсказания с блочным копированием имеет ряд проблем. Например, решения, принятые на стороне кодера, о том, как использовать внутренне предсказание с BC, принимаются неэффективным образом. В качестве другого примера, во многих сценариях значения BV передаются в качестве сигнализации неэффективным образом.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

В данном разделе подробно описываются нововведения в режиме внутреннего предсказания с блочным копированием (BC) в операциях на стороне кодера. Например, некоторые из предложенных нововведений относятся к путям оценки восстановленных значений отсчетов в области перекрытия текущего блока во время оценки блочного вектора («BV»). Другие нововведения относятся к предсказанию значений BV во время кодирования или декодирования с использованием так называемого «пинг-понгового» подхода, согласно которому кодер или декодер выбирает одно из пары пригодных

значений BV при предсказании значения BV для блока.

Согласно одному аспекту описанных здесь нововведений кодер или декодер выполняет обработку единицы кодирования (CU) текущей картинки, где CU содержит четыре единицы предсказания («PU»). Единица CU представляет собой набор, состоящий из одного или нескольких блоков, (в общем случае) предназначенных для кодирования и декодирования, причем каждая из PU представляет собой набор, состоящий из одного или нескольких блоков в CU, предназначенных для передачи в качестве сигнализации информации о предсказании и/или для обработки предсказания. Для каждой из четырех PU кодер или декодер (1) предсказывает значение BV, используя одно из пары пригодных значений BV, где значение флага указывает на выбор из пары пригодных значений BV, и (2) обрабатывает значение BV для PU, используя предсказанное значение BV. Упомянутая пара пригодных значений BV может отличаться для разных PU в CU. Например, для первой PU пара пригодных значений BV включает в себя первое начальное пригодное значение BV (обозначенное как BVinit_0) и второе начальное пригодное значение BV (обозначенное как BVinit_1). Для второй PU пара пригодных значений BV включает в себя значение BV для первой PU и BVinit_0. Для третьей PU пара пригодных значений BV включает в себя значения BV для первой и второй PU текущей CU. Наконец, для четвертой PU пара пригодных значений BV включает в себя значения BV для второй и третьей PU текущей CU. Однако, для последующей CU текущей картинки пара пригодных значений BV включает в себя значения BV для первой и второй PU текущей CU (даже если значения BV для первой и второй PU текущей CU были обработаны до значений BV для третьей и четвертой PU текущей CU).

Согласно другому аспекту описанных здесь нововведений кодер или декодер обрабатывает множество CU текущей картинки. По меньшей мере одна CU имеет единственную PU, и по меньшей мере одна из CU имеет множество PU. Для данной CU из множества CU (если данная CU имеет единственную PU) кодер или декодер (1) предсказывает значение BV, используя одно из пары пригодных значений BV (включая первое начальное пригодное значение BV, запомненное первым в буфере BV, и второе начальное пригодное значение BV, запомненное во втором буфере), где значение флага указывает на выбор из пары пригодных значений BV, и (2) выполняет обработку значения BV для единственной PU, используя предсказанное значение BV. Таким путем кодер или декодер избирательно обновляет первый буфер BV и второй буфер BV в зависимости от того, равно ли значение BV для данной PU значению BV, запомненному в первом буфере BV.

В противном случае (когда данная CU имеет множество PU) кодер или декодер для каждой из PU (1) предсказывает значение BV, используя одно из пары пригодных значений BV, где значение флага указывает на выбор из пары пригодных значений BV, и (2) выполняет обработку значения BV для PU, используя предсказанное значение BV. Упомянутая пара пригодных значений BV может отличаться для разных PU. Если PU является первой или второй PU данной CU, то кодер или декодер избирательно обновляет первый буфер BV и второй буфер BV в зависимости от того, равно ли значение BV для данной PU значению BV, запомненному в первом буфере BV. Для третьей PU или четвертой PU данной CU кодер пропускает обновление буферов BV.

Согласно еще одному аспекту описанных здесь нововведений кодер оценивает восстановленные значения отсчетов в области перекрытия текущего блока картинки. Текущий блок состоит из множества блоков меньшего размера. Область перекрытия покрывает части текущего блока, которые находятся в потенциально пригодных областях внутреннего предсказания по меньшей мере для одного из множества блоков

меньшего размера. Затем кодер выполняет оценку BV с целью определения значения BV для текущего блока. При оценке BV используется по меньшей мере несколько из оцененных восстановленных значений отсчетов в области перекрытия текущего блока.

5 Нововведения для предсказания BV и опций на стороне кодера для режима внутреннего предсказания с ВС можно реализовать в качестве части способа, части вычислительного устройства, адаптированного для выполнения того способа, или в качестве части материальной считываемой компьютером среды (носителя), где хранятся исполняемые компьютером команды, иницирующие выполнение вычислительным устройством упомянутого способа. Описанные здесь различные нововведения можно
10 использовать в сочетании или отдельно.

Вышеописанные и другие цели, признаки и преимущества изобретения станут более очевидными по прочтении нижеследующего подробного описания, в котором присутствуют ссылки на сопроводительные чертежи.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

15 Фиг. 1 - схема примерной вычислительной системы, в которой могут быть реализованы некоторые описанные здесь варианты осуществления изобретения;

Фигуры 2a и 2b - схемы примерных сетевых сред, в которых могут быть реализованы некоторые описанные здесь варианты осуществления изобретения;

20 Фиг. 3 - схема примерной системы кодера, вместе с которой могут быть реализованы некоторые описанные здесь варианты осуществления изобретения;

Фиг. 4 - схема примерной системы декодера, вместе с которой могут быть реализованы некоторые описанные здесь варианты осуществления изобретения;

25 Фигуры 5a и 5b - схемы, иллюстрирующие примерный видеокодер, вместе с которым могут быть реализованы некоторые описанные здесь варианты осуществления изобретения;

Фиг. 6 - схема, иллюстрирующая примерный видеodeкодер, вместе с которым могут быть реализованы некоторые описанные здесь варианты осуществления изобретения;

Фиг. 7 - схема, иллюстрирующая внутреннее предсказание с ВС для одного блока картинки;

30 Фиг. 8 - схема, иллюстрирующая примерные ограничения на область поиска значений BV;

Фиг. 9 - блок-схема, иллюстрирующая обобщенный способ кодирования с внутренним режимом предсказания с ВС с наложенным одним или несколькими ограничениями на выбор значений BV;

35 Фиг. 10 - схема, иллюстрирующая примерный порядок z-сканирования для блоков картинки;

Фиг. 11 - схема, иллюстрирующая значения BV в буферах BV, и фиг. 12 - схема, иллюстрирующая пригодные значения BV в другой структуре данных;

40 Фиг. 13 - схема, иллюстрирующая пример предсказания BV для единиц PU размером $2N \times 2N$ единицы CU размером $2N \times 2N$ согласно примерным пинг-понговым подходам;

Фиг. 14 - схема, иллюстрирующая примеры предсказания BV для единиц PU размером $L \times N$ единиц CU размером $2N \times 2N$ согласно примерным пинг-понговым подходам;

45 Фиг. 15 - схема, иллюстрирующая примеры предсказания BV для единиц PU размером $L \times 2N$ или PU размером $2N \times 2N$ единиц CU размером $2N \times 2N$ согласно примерным пинг-понговым подходам;

Фиг. 16 - блок-схема, иллюстрирующая примерную методику предсказания значений BV для CU, имеющей множество PU, согласно примерным пинг-понговым подходам;

Фиг. 17 - блок-схема, иллюстрирующая примерную методику предсказания значений

BV для единиц CU согласно примерным пинг-понговым подходам;

Фиг. 18 - блок-схема, иллюстрирующая обобщенную методику избирательного слияния блоков в более крупный блок во время оценки BV;

Фиг. 19 - схема, иллюстрирующая преимущество избирательного слияния блоков в более крупный блок во время оценки BV;

Фиг. 20 - блок-схема, иллюстрирующая обобщенную методику параллельного выполнения оценки BV и принятия решений о расщеплении для каждого блока;

Фиг. 21 - блок-схема, иллюстрирующая обобщенную методику параллельной оценки пригодного значения BV и принятия решений о разделении для каждого блока;

Фигуры 22a и 22b - схемы, иллюстрирующие область перекрытия текущего блока во время оценки BV;

Фиг. 23 - блок-схема, иллюстрирующая примерную методику оценки BV, согласно которой кодер оценивает восстановленные значения отсчетов в области перекрытия;

Фиг. 24 - схема, иллюстрирующая область перекрытия блоков с минимальной размерностью преобразования для видео в формате YUV 4:2:2 во время оценки BV.

ПОДРОБНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

В подробном описании изобретения представлены нововведения в предсказание блочного вектора («BV») во время кодирования и декодирования, а также нововведения, касающиеся принятия решений на стороне кодера для режима внутреннего предсказания с блочным копированием («BC») во время кодирования. В частности, в подробном описании изобретения представлены нововведения, касающиеся оценки значений отсчетов в области перекрытия текущего блока во время оценки BV. В этом описании также представлены нововведения, касающиеся предсказания BV во время кодирования или декодирования с использованием «пинг-понговых» подходов, согласно которым кодер или декодер осуществляет выбор из пары пригодных значений BV при предсказании значения BV для блока.

Хотя здесь показано, что описанные операции отчасти выполняются видеокодером или видеодекодером, во многих случаях эти операции могут выполняться средствами обработки медиа другого типа (например, кодером изображений, декодером изображений).

Некоторые из описанных здесь нововведений проиллюстрированы со ссылками на синтаксические элементы и операции, специфичные для стандарта HEVC. Например, показана ссылка на предварительную версию JCTVC-N1005 стандарта HEVC «High Efficiency Video Coding (HEVC) Range Extensions Text Specification: Draft 5,» JCTVC-O1005-v3, November 2013. Описанные здесь нововведения также можно реализовать для других стандартов или форматов.

В более общем случае возможно использование различных альтернатив для описанных здесь примеров. Например, некоторые из описанных здесь способов можно изменить, изменив порядок описанных действий способа путем разделения на части, повторения или отбрасывания некоторых действий способа и т.д. Различные аспекты раскрытой здесь технологии можно использовать в их сочетании или по отдельности. В разных вариантах осуществления используется одно или несколько из описанных здесь нововведений. Некоторые из описанных здесь нововведений направлены на решение одной или нескольких проблем, отмеченных в разделе «Уровень техники». Как правило, данная методика/средство не решает всех указанных проблем.

I. Примерные вычислительные системы

На фиг. 1 показан обобщенный пример подходящей вычислительной системы (100), в которой можно реализовать несколько из описанных здесь нововведений. Здесь не

предполагается, что вычислительная система (100) накладывает какое-либо ограничение на объем использования или функциональные возможности, так что эти нововведения могут быть реализованы в различных вычислительных системах общего назначения или специализированных вычислительных системах.

5 Обратимся к фиг. 1, где вычислительная система (100) включает в себя один или несколько обрабатывающих блоков (110, 115) и память (120, 125). Обрабатывающие блоки (110, 115) выполняют исполняемые компьютером программы. Обрабатывающий блок может представлять собой центральный процессор («CPU») общего назначения, процессор в прикладной специализированной интегральной схеме («ASIC») или
10 процессор любого другого типа. В многопроцессорной системе для повышения производительности исполняемые компьютером команды выполняются множеством обрабатывающих блоков. Например, на фиг. 1 показан центральный процессор (110), а также блок графической обработки или сопроцессорный блок (115). Материальная память (120, 125) может представлять собой энергозависимую память (например, регистры, кэш-память, RAM), энергонезависимую память (например, ROM, EEPROM, флэш-память и т.д.) или некоторую комбинацию из двух указанных устройств, доступных обрабатывающему блоку (блокам). В памяти (120, 125) хранится программное обеспечение (180), реализующее одно или несколько нововведений для предсказания BV и/или оценки значений отсчетов в области перекрытия текущего блока во время
15 оценки BV, или других опций на стороне кодера для режима внутреннего предсказания с BC, в виде исполняемых компьютером команд, подходящих для выполнения обрабатывающим блоком (блоками).

Вычислительная система может иметь дополнительные признаки. Например, вычислительная система (100) включает в себя хранилище (140), одно или несколько
25 устройств (150) ввода, одно или несколько устройств (160) вывода и одно или несколько коммуникационных соединений (170). Средства обеспечения межсоединений (не показаны), такие как шина, контроллер или сеть, обеспечивают соединения между компонентами вычислительной системы (100). Как правило, программное обеспечение операционной системы (не показано) обеспечивает операционную среду для других
30 программных средств, реализуемых в вычислительной системе (100), и координирует операции, выполняемые компонентами вычислительной системы (100).

Материальное хранилище данных (140) может быть съемным или несъемным и может включать в себя магнитные диски, магнитные ленты или кассеты, CD-ROM, DVD или любой другой носитель, который можно использовать для хранения информации,
35 которая может быть доступна в вычислительной системе (100). Хранилище (140) обеспечивает хранение команд для программного обеспечения (180), реализующего одно или несколько нововведений для предсказания BV и/или оценки значений отсчетов в области перекрытия текущего блока во время оценки BV, или других опций на стороне кодера для режима внутреннего предсказания с BC.

40 Устройство (устройства) (150) ввода могут представлять собой устройство сенсорного ввода, такое как клавиатура, мышь, перо или шаровой манипулятор, устройство ввода голосом, сканирующее устройство или иное устройство, обеспечивающее ввод в вычислительную систему (100). Для видео устройством (устройствами) (150) ввода может быть камера, видеокарта, карта селектора TV программ или аналогичное
45 устройство, которое получает видео в аналоговой или цифровой форме, либо CD-ROM или CD-RW, которые считывают отсчеты видео в вычислительную систему (100). Устройство (устройства) (160) вывода может представлять собой дисплей, принтер, динамик, устройство записи на CD или какое-либо другое устройство, которое

обеспечивает вывод из вычислительной системы (100).

Коммуникационное соединение (соединения) (170) предоставляет возможность связи через среду связи с другим вычислительным объектом. Коммуникационная среда переносит информацию, такую как исполняемые компьютером команды, ввод или вывод видео или иные данные в сигнале, модулированном данными. Сигналом, модулированным данными, является сигнал, имеющий одну или несколько характеристик, установленных или измененных таким образом, чтобы обеспечить кодирование информации в указанном сигнале. Например, но не как ограничение, коммуникационная среда может использовать электрическую, оптическую, радиочастотную (RF) или иную несущую.

Упомянутые нововведения можно описать в общем контексте считываемых компьютером сред. Считываемые компьютером среды представляют собой любые доступные материальные среды, к которым можно обращаться в данной вычислительной среде. Например, но не как ограничение, при использовании вычислительной системы (100) считываемые компьютером среды включают в себя память (120, 125), хранилище (140) и любые комбинации из них.

Упомянутые нововведения можно описать в общем контексте исполняемых компьютером команд, таких как команды, содержащиеся в программных модулях, которые исполняются вычислительной системой на целевом реальном или виртуальном процессоре. Обычно программные модули включают в себя стандартные программы, программы, библиотеки, объекты, классы, компоненты, структуры данных и т.д., которые решают конкретные задачи или реализуют конкретные типы абстрактных данных. Функциональные возможности программных модулей могут быть объединены или разделены между программными модулями в соответствии с требованиями в различных вариантах осуществления. Исполняемые компьютером команды для программных модулей могут выполняться в локальной или распределенной вычислительной системе.

Термины «система» и «устройство» используются здесь как взаимозаменяемые. Если из контекста в явном виде не следует иное, ни один из этих терминов не накладывает какое-либо ограничение на тип вычислительной системы или вычислительного устройства. В общем случае вычислительная система или вычислительное устройство может быть локальным или распределенным и может включать в себя любую комбинацию специализированных аппаратных средств и/или аппаратных средств общего назначения с программным обеспечением, реализующим описанные здесь функциональные возможности.

Раскрытые здесь способы также можно реализовать с использованием специализированных аппаратных вычислительных средств, сконфигурированных для выполнения любого из раскрытых здесь способов. Например, раскрытые способы можно реализовать с помощью интегральной схемы (например, ASIC) (такой как блок обработки цифровых сигналов (DSP), блок обработки графики (GPU)) или программируемое логическое устройство (PLD), такое как вентильная матрица, программируемая пользователем (FPGA)), специально спроектированной или сконфигурированной для реализации любого из раскрытых здесь способов.

Для представления в данном подробном описании используются термины типа «определяет» и «использует» для описания компьютерных операций в вычислительной системе. Эти термины являются абстракциями высокого уровня для операций, выполняемых компьютером, и их не следует путать с действиями, выполняемыми человеком. Реальные вычислительные операции, соответствующие этим терминам,

зависят от конкретной реализации. Используемый здесь термин «optimiz*» (в том числе его вариации, такие как «optimization» и «optimizing» (оптимизация)) относится к выбору среди опций при заданной области действия решения и не предполагает, что оптимизированный выбор является «наилучшим» или «оптимальным» выбором для

5 II. Примерные сетевые среды

На фигурах 2a и 2b показаны примерные сетевые среды (201, 202), которые включают в себя видеокодеры (220) и декодеры (270). Кодеры (220) и декодеры (270) соединены через сеть (250), использующую соответствующий коммуникационный

10 протокол. Сеть (250) может включать в себя Интернет или другую компьютерную сеть. В сетевой среде (201), показанной на фиг. 2a, каждое средство связи (210) в режиме реального времени («RTC») включает в себя кодер (220) и декодер (270) для двустороннего обмена. Данный кодер (220) может создавать выходные данные, совместимые с версией или расширением стандарта HEVC, (также известного как H.265), стандарта SMPTE 421M, стандарта ISO-IEC 14496-10 (также называемого H.264 или

15 AVC), другого стандарта или патентованного формата, с соответствующим декодером (270), получающим кодированные данные от кодера (220). Упомянутая двусторонняя связь может быть частью видеоконференции, видеотелефонного звонка или другого сценария двусторонней или многосторонней связи. Хотя сетевая среда (201) на фиг. 2a

20 включает в себя два средства (210) связи в режиме реального времени, сетевая среда (201) может вместо этого включать в себя три или более средства (210) связи в режиме реального времени, которые являются участниками многосторонней связи.

Средство (210) связи в режиме реального времени организует кодирование, выполняемое кодером (220). На фиг.3 показана примерная система (300) кодера, которая

25 может быть включена в средство (210) связи в режиме реального времени. В качестве альтернативы, в средстве (210) связи в режиме реального времени может быть использована другая система кодера. Средство (210) связи в режиме реального времени также организует декодирование, выполняемое декодером (270). На фиг.4 показана примерная система (400) декодера, которая может быть включена в средство (210) связи

30 в режиме реального времени. В качестве альтернативы, в средстве (210) связи в режиме реального времени может быть использована другая система декодера.

В сетевой среде (202), показанной на фиг. 2b, средство кодирования (212) включает в себя кодер (220), выполняющий кодирование видео для доставки на множество средств (214) воспроизведения, которые включают в себя декодеры (270). Однонаправленная

35 связь может быть обеспечена для системы видеонаблюдения, системы текущего контроля с использованием веб-камеры, презентации в режиме телеконференции с использованием удаленных настольных компьютеров или для другого сценария, в котором видео кодируют и посылают из одного места в одно или несколько других мест. Хотя сетевая среда (202) на фиг. 2b включает в себя два средства (214) воспроизведения, сетевая среда

40 (202) может включать в себя больше или меньше средств (214) воспроизведения. Обычно средство (214) воспроизведения осуществляет связь со средством (212) кодирования для определения потока видео для его приема средством (214) воспроизведения. Средство 214 воспроизведения принимает указанный поток, выполняет буферизацию полученных кодированных данных за соответствующий период и приступает к декодированию и

45 воспроизведению.

На фиг. 3 показана примерная система (300) кодера, которая может входить в состав средства (212) кодирования. В качестве альтернативы, в средстве (212) кодирования может использоваться другая система кодера. Средство (212) кодирования также может

включать в себя логический блок контроллера на стороне сервера для организации соединений с одним или несколькими средствами (214) воспроизведения. На фиг. 4 показана примерная система (400) декодера, которая может быть включена в средство (214) воспроизведения. В качестве альтернативы в средстве (214) воспроизведения
 5 может использоваться другая система декодера. Средство (214) воспроизведения также может включать в себя логический блок контроллера на стороне клиента для организации соединений со средством 212 кодирования.

III. Примерные системы кодера

На фиг. 3 представлена блок-схема примерной системы (300) кодера, вместе с которой
 10 можно реализовать некоторые описанные здесь варианты осуществления изобретения. Система (300) кодера может представлять собой средство кодирования общего назначения, способное работать в любом из множества режимов кодирования, таких как режим кодирования с малой задержкой для связи в реальном времени, режим транскодирования и режим кодирования с большой задержкой для создания медиа,
 15 предназначенных для воспроизведения из файла или потока, либо может представлять собой специализированное средство кодирования, адаптированное к одному указанному режиму кодирования. Система (300) кодера может быть реализована в виде модуля операционной системы, как часть библиотеки приложений или в виде автономного приложения. В конце концов, система (300) кодера принимает в целом
 20 последовательность исходных видеок кадров (311) от источника (310) видео и создает кодированные данные в качестве выхода в канал (390). Кодированные данные, выведенные в указанный канал, могут включать в себя контент, кодированный с использованием режима внутреннего предсказания с ВС.

Источником (310) видео может быть камера, карта селектора каналов, запоминающая
 25 среда или иной источник цифрового видео. Источник (310) видео создает последовательность видеок кадров с частотой кадров, например, 30 кадров в секунду. Используемый здесь термин «кадр» обычно относится к исходным, кодированным или восстановленным данным изображения. Для видео с прогрессивной разверткой кадром является видеок кадр с прогрессивной разверткой. Для чересстрочного видео в примерных
 30 вариантах осуществления чересстрочный видеок кадр является построчным перед кодированием. В качестве альтернативы кодируются два комплементарных чересстрочных видеополя в качестве чересстрочного видеок кадра или отдельные поля. Помимо указания на видеок кадр с прогрессивной разверткой термин «frame» (кадр), или «picture» (картинка), может указывать на одно непарное видеополе,
 35 комплементарную пару видеополей, плоскость видеообъекта, которая представляет видеообъект в данный момент времени, или интересующую область в более крупном изображении. Плоскость или область видеообъекта может являться частью более крупного изображения, которое включает в себя множество объектов или областей сцены.

Поступающий исходный кадр (311) запоминается в зоне (320) временного хранения
 40 исходных кадров, которая включает в себя множество зон (321, 322..., 32n) временного хранения (буферов кадров). Буфер (321, 322, и т.д.) запоминает исходный кадр в зоне (320) временного хранения исходных кадров. После запоминания одного или нескольких исходных кадров (311) в буферах (321, 322, и т.д.) кадров селектор (330) кадров выбирает
 45 отдельный исходный кадр из зоны (320) временного хранения исходных кадров. Порядок отбора кадров селектором (330) кадров для ввода в кодер (340) может отличаться от порядка, в котором эти кадры создаются источником (310) видео, например, кодирование некоторых кадров уместно задерживать, чтобы дать возможность в

первую очередь кодировать некоторые более поздние кадры, и обеспечить таким образом временное обратное предсказание. Перед кодером (340) система (300) кодера может включать в себя процессор предварительной обработки (не показан), выполняющий предварительную обработку (например, фильтрацию) выбранного кадра (331) перед кодированием. Эта предварительная обработка может также включать в себя преобразование цветового пространства в первичные (например, яркости) и вторичные (например, цветоразности красного и синего) компоненты и обработку передискретизации (например, для уменьшения пространственного разрешения компонент цветности) для кодирования. Как правило, перед кодированием видео преобразуют в цветовое пространство, например, YUV, в котором значения отсчетов компоненты яркости (Y) представляют значения яркости или интенсивности, а значения отсчетов компонент цветности (U, V) представляют цветоразностные значения. Значения отсчетов цветности могут быть подвергнуты субдискретизации для обеспечения более низкой частоты дискретизации цветности (например, для формата YUV 4:2:0 или YUV 4:2:2), либо значения отсчетов цветности могут иметь такое же разрешение, как значения отсчетов яркости (например, для формата YUV 4:4:4). В формате YUV 4:2:0 компоненты цветности подвергаются понижающей дискретизации с коэффициентом два по горизонтали и коэффициентом два по вертикали. В формате YUV 4:2:2 коэффициенты цветности подвергаются понижающей дискретизации с коэффициентом два по горизонтали. Либо кодирование видео может быть выполнено в другом формате (например, в формате RGB 4:4:4).

Кодер (340) выполняет кодирование выбранного кадра (331) для создания кодированного кадра (341), а также создает сигналы (342) операций управления памятью («MMCO») или информацию о наборе опорных картинок («RPS»). Если текущий кадр не является первым кадром, который был кодирован при выполнении процесса кодирования, то кодер (340) может использовать один или несколько ранее кодированных/декодированных кадров (369), которые были сохранены в зоне (360) временного хранения декодированных кадров. Указанные запомненные декодированные кадры (369) используют в качестве опорных кадров для межкадрового предсказания контента текущего исходного кадра (331). Информация (342) MMCO/RPS указывает декодеру, какие восстановленные кадры можно использовать в качестве опорных кадров, и, следовательно, должны быть запомнены в зоне хранения кадров.

В общем случае, кодер (340) включает в себя множество кодирующих модулей, которые решают задачи кодирования, такие как разделение на плитки, оценка внутреннего предсказания и предсказание, оценка и компенсация движения, частотные преобразования, квантование и энтропийное кодирование. Точные операции, выполняемые кодером (340) могут варьироваться в зависимости от формата сжатия. Формат выходных кодированных данных может представлять собой версию или расширение формата HEVC (H.265), формата Windows Media Video, формата VC-1, формата MPEG-x (например, MPEG-1, MPEG-2 или MPEG-4), формата H.26x (например, H.261, H.262, H.263, H.264) или другого формата.

Кодер (340) может разделить кадр на множество плиток одинакового размера или разных размеров. Например, кодер (340) разделяет кадр вдоль строк и столбцов плиток, что вместе с границами кадра определяет горизонтальные и вертикальные границы плиток в кадре, где каждая плитка представляет собой прямоугольную область. Плитки часто используют для улучшения опций для параллельной обработки. Кадр также можно сформировать в виде одного или нескольких слайсов, где один слайс может представлять весь кадр или область кадра. Слайс можно будет декодировать независимо

от других слайсов в кадре, что повышает устойчивость к ошибкам. Контент слайса или плитки дополнительно разделяется на блоки или другие наборы отсчетов в целях кодирования и декодирования.

5 Что касается синтаксиса согласно стандарту HEVC, кодер расщепляет контент кадра (или слайса, или плитки) на единицы кодировочного дерева. Единица кодировочного дерева («STU») включает в себя значения отсчетов яркости, сформированных в виде блока кодировочного дерева яркости («СТВ»), и соответствующие значения отсчетов цветности, сформированных в виде двух СТВ цветности. Размер STU (и ее СТВ) выбирается кодером и может, например, составлять 64×64, 32×32 или 16×16 значений отсчетов. Единица STU включает в себя одну или несколько единиц кодирования. 10 Единица кодирования («CU») содержит блок кодирования («СВ») яркости и два соответствующих СВ цветности. Например, STU с одним СТВ яркости размером 64×64 и двумя СТВ цветности (в формате 4:4:4 YUV) размером 64×64 может быть разделена на четыре CU, где каждая CU включает в себя один СВ яркости размером 32×32 и два СВ цветности размером 32×32, причем каждую CU можно разделить дополнительно на более мелкие CU. Либо, как в другом примере, STU с одним СТВ яркости размером 64×64 и двумя СТВ цветности размером 32×32 (в формате 4:2:0 YUV) можно разделить на четыре CU, где каждая CU включает в себя СВ яркости 32×32 и два СВ цветности 16×16, при этом каждую CU можно дополнительно разделить на более мелкие CU. 15 Минимальный допустимый размер CU (например, 8×8, 16×16) может быть передан в битовом потоке в качестве сигнализации.

Обычно CU имеет режим предсказания, например, внешнего или внутреннего. CU включает в себя один или несколько блоков предсказания в целях сигнализации о предсказании (например, подробности режима предсказания, значения смещения и т.д.) 25 и/или обработки предсказания. Единица предсказания (PU) содержит блок предсказания яркости (PB) и два PB цветности. Для CU, полученной посредством внутреннего предсказания PU имеет тот же размер, что и CU, если CU не имеет минимальный размер (например, 8×8). В этом случае CU можно разделить на четыре более мелких PU (например, каждая размером 4×4, если минимальный размер CU составляет 8×8) или 30 PU может иметь минимальный размер CU, как указано синтаксическим элементом для данной CU. Единица CU также имеет одну или несколько единиц преобразования в целях кодирования/декодирования остатков, где единица преобразования («ТУ») имеет блок преобразования («ТВ») и два ТВ цветности. Единица PU в CU, полученной с использованием внутреннего предсказания, может содержать одну ТУ (равную по 35 размеру единице PU) или несколько ТУ. Используемый здесь термин «блок» может указывать на СВ, PB, ТВ или другой набор значений отсчетов в зависимости от контекста. Кодер решает, каким образом разделить видео на единицы STU, CU, PU, ТУ и т.д.

Обратимся к фиг. 3, где кодер представляет полученный путем внутреннего кодирования блок исходного кадра (331) в контексте предсказания исходя из других, 40 ранее восстановленных значений отсчетов в кадре (331). В случае внутреннего предсказания с ВС средство оценки внутреннего предсказания оценивает смещение блока по отношению к другим, ранее восстановленным значениям отсчетов. Опорная область внутреннего предсказания (или для краткости «область внутреннего предсказания») представляет собой область отсчетов в кадре, которые используют для создания значений ВС предсказания для блока. Область внутрикадрового предсказания может быть указана значением блочного вектора («BV») (определено в оценке BV). 45 Что касается внутреннего пространственного предсказания для блока, то средство

оценки внутреннего предсказания оценивает результат экстраполяции соседних восстановленных значений отсчетов в данном блоке. Средство оценки внутреннего предсказания может вывести информацию о предсказании (например, значения блочного вектора (BV) для внутреннего предсказания с ВС и режим (направление) предсказания для внутреннего пространственного предсказания), которая подвергается энтропийному кодированию. Средство внутрикадрового предсказания использует упомянутую информацию о предсказаниях для определения значений внутреннего предсказания.

Кодер (340) представляет кодированный с использованием межкадрового предсказания блок исходного кадра (331) в контексте предсказания на основе опорных кадров. Средство оценки движения оценивает движение блока относительно одного или нескольких опорных кадров (369). При использовании множества опорных кадров это множество опорных кадров может быть взято из разных временных направлений или одного и того же временного направления. Опорная область предсказания с компенсацией движения является областью отсчетов в опорном кадре (кадрах), которые используют для создания значений предсказания с компенсацией движения для блока отсчетов текущего кадра. Средство оценки движения выводит информацию о движении, такую как информация о векторе движения («MV»), которая кодируется с использованием энтропийного кодирования. Компенсатор движения использует векторы MV для опорных кадров (369), чтобы определить значения предсказания с компенсацией движения для межкадрового предсказания.

Кодер может определить различия (если это имеет место) между значениями предсказания (внутреннего или внешнего) блока и соответствующими исходными значениями. Эти значения остатков предсказания дополнительно кодируют, используя частотное преобразование, квантование и энтропийное кодирование. Например, кодер (340) устанавливает значения для параметра квантования («QP») для кадра, плитки, слайса и/или другой части видео и квантует коэффициенты преобразования соответствующим образом. Энтропийный кодер кодера (340) выполняет сжатие квантованных значений коэффициентов преобразования, а также некоторой вспомогательной информации (например, информация о векторе движения, значения BV, значения QP, принятые решения о режиме, выбранные параметры). Типовые приемы энтропийного кодирования включают в себя экспоненциальное кодирование Голomba, арифметическое кодирование, дифференциальное кодирование, кодирование Хаффмана, кодирование длин серий, кодирование с преобразованием «переменная длина слова-переменная длина слова» («V2V»), кодирование с преобразованием «переменная длина слова-фиксированная длина слова» («V2F»), кодирование Лемпела-Зива («LZ»), словарное кодирование, энтропийное кодирование с разделением интервала вероятности («PIPE») и комбинации вышеуказанных способов кодирования. Энтропийный кодер может использовать разные приемы кодирования для информации разного рода и может использовать сочетания множества приемов (например, использовать кодирование Голomba-Райса с последующим арифметическим кодированием) и может осуществлять выбор из множества кодовых таблиц в рамках конкретного приема кодирования.

В контур компенсации движения в кодере (340) включен адаптивный деблокинг-фильтр для сглаживания разрывов на граничных строках и/или столбцах блоков в декодированном кадре. Вдобавок или в качестве альтернативы в качестве внутриконтурных операций фильтрации могут быть использованы другие виды фильтрации (такие как дереверберационная фильтрация, адаптивная контурная фильтрация («ALF») или фильтрация с адаптивным смещением («SAO»); не показаны).

Поскольку упомянутые зависимости и структуры упорядочения для кадров уже известны в кодере (340), кодированные кадры (341) и информация (342) о MMCO/RPS (или информационный эквивалент информации (342) о MMCO/RPS) обрабатываются эмулятором (350) процесса декодирования. Эмулятор (350) процесса декодирования реализует некоторые из функциональных возможностей декодера, например, задачи декодирования для восстановления опорных кадров. В соответствии с информацией (342) о MMCO/RPS эмулятор (350) процесса декодирования определяет, требуется ли восстановление и запоминание данного кодированного кадра (341) для использования в качестве опорного кадра при межкадровом предсказании последующих кадров, подлежащих кодированию. Если кодированный кадр (341) требуется запомнить, то эмулятор (350) процесса декодирования моделирует процесс декодирования, который будет выполняться декодером, получающим кодированный кадр (341), и создает соответствующий декодированный кадр (351). Действуя таким образом, при использовании кодером (340) декодированного кадра (кадров) (369), который был запомнен в зоне (360) хранения декодированных кадров, эмулятор (350) процесса декодирования также использует декодированный кадр (кадры) (369) из зоны (360) хранения как часть процесса декодирования.

Зона (360) временного хранения декодированных кадров включает в себя множество зон (361, 362, ..., 36n) хранения (буферов кадров). В соответствии с информацией (342) о MMCO/RPS эмулятор (350) процесса декодирования управляет контентом зоны (360) хранения, чтобы идентифицировать буферы (361, 362, и т.д.) кадров с кадрами, которые больше не нужны кодеру (340) для использования в качестве опорных кадров. После моделирования процесса декодирования эмулятор (350) процесса декодирования запоминает вновь декодированный кадр (351) в буфере (361, 362 и т.д.) кадра, который был идентифицирован указанным образом.

Кодированные кадры (341) и информация (342) о MMCO/RPS временно буферизируются в зоне (370) кодированных данных. Кодированные данные, собранные в зоне (370) кодированных данных, содержат в качестве части синтаксиса элементарно кодированного битового видеопотока, кодированные данные для одного или нескольких кадров. Кодированные данные, собранные в зоне (370) кодированных данных, также могут включать в себя метаданные для медиа, относящиеся к кодированным видеоданным (например, в виде одного или нескольких параметров в одном или нескольких сообщениях с дополнительной расширяющей информацией («SEI») или в сообщениях с информацией о годности видео («VUI»)).

Агрегированные данные (371) из зоны (370) временно кодированных данных обрабатываются канальным кодером (380). Канальный кодер (380) может пакетировать и/или мультиплексировать агрегированные данные для передачи или хранения в виде потока медиа (например, в соответствии с потоком программ медиа или форматом транспортного потока, таким как ITU-T H.222.0 ISO/IEC 13818-1 или форматом транспортного протокола Интернет в режиме реального времени, например, IETF RFC 3550), и в этом случае канальный кодер (380) может добавить синтаксические элементы как часть синтаксиса потока передачи медиа. Либо канальный кодер (380) может сформировать агрегированные данные для хранения в виде файла (например, согласно формату контейнера медиа, такому как ISO/IEC 14496-12), и в этом случае канальный кодер (380) может добавить синтаксические элементы как части синтаксиса файла хранения медиа. Либо, в более общем случае, канальный кодер (380) может реализовать один или несколько протоколов мультиплексирования медиасистем или транспортных протоколов, и в этом случае канальный кодер (380) может добавить синтаксические

элементы как части синтаксиса указанного протокола (протоколов). Канальный кодер (380) обеспечивает вывод в канал (390), который представляет хранилище, коммуникационное соединение или другой канал для указанного вывода. Канальный кодер (380) или канал (390) также могут включать в себя другие элементы (не показаны), например, кодирование с прямым исправлением ошибок («FEC») и модуляцию аналогового сигнала.

IV. Примерные системы декодера

На фиг. 4 представлена блок-схема примерной системы (400) декодера, вместе с которым можно реализовать некоторые описанные здесь варианты осуществления изобретения. Система (400) декодера может представлять собой средство декодирования общего назначения, способное работать в любом из множества режимов декодирования, таких как режим декодирования с малой задержкой для связи в реальном времени, и режим декодирования с большой задержкой для воспроизведения медиа из файла или потока, либо может представлять собой специализированное средство декодирования, адаптированное к одному указанному режиму декодирования. Система (400) декодера может быть реализована в виде модуля операционной системы, в виде части библиотеки приложений или в виде автономного приложения. Система (400) декодера принимает в целом кодированные данные из канала (410) и создает восстановленные кадры в качестве выхода для адресата (490) выхода. Эти кодированные данные могут включать в себя контент, кодированный с использованием режима внутреннего предсказания с ВС.

Система (400) декодера включает в себя канал (410), который может представлять хранилище, коммуникационное соединение или другой канал для кодированных данных в качестве входных данных. Канал (410) создает кодированные данные, полученные с использованием канального кодирования. Канальный декодер (420) может обрабатывать кодированные данные. Например, канальный декодер (420) распаковывает и/или демультиплексирует данные, которые были агрегированы для передачи или хранения в виде потока медиа (например, согласно формату потока программ медиа или транспортного потока, такому как ITU-T H.222.0 ISO/IEC 13818-1 или формату транспортного протокола Интернет в режиме реального времени, такому как IETF RFC 3550), и в этом случае канальный декодер (420) может проанализировать синтаксические элементы, добавленные как часть синтаксиса потока передачи медиа. Либо канальный декодер (420) выделяет кодированные видеоданные, которые были агрегированы для хранения в виде файла (например, согласно формату контейнера медиа, такому как ISO/IEC 14496-12), и в этом случае канальный декодер (420) может выполнить синтаксический анализ элементов, добавленных как часть синтаксиса файла хранения медиа. Либо, в более общем случае, канальный декодер (420) может реализовать один или несколько протоколов демультиплексирования систем медиа или транспортных протоколов, и в этом случае канальный декодер (420) может выполнить синтаксический анализ элементов, добавленных как часть синтаксиса указанного протокола (протоколов). Канал (410) или канальный декодер (420) также могут включать в себя другие элементы (не показаны), например, для FEC декодирования и демодуляции аналогового сигнала.

Кодированные данные (421), выведенные из канального декодера (420), запоминаются в зоне (430) временных кодированных данных, пока не будет получено достаточное количество таких данных. Кодированные данные (421) включают в себя кодированные кадры (431) и информацию (432) о MMCO/RPS. Кодированные данные (421) в зоне (430) кодированных данных содержат, в качестве части синтаксиса элементарно

кодированного битового потока кодированные данные для одной или нескольких картинок. Кодированные данные (421) в зоне (430) кодированных данных также могут включать в себя метаданные медиа, относящиеся к кодированным видеоданным (например, в виде одного или нескольких параметров в одном или нескольких сообщениях SEI или сообщениях VUI).

В общем случае в зоне (430) кодированных данных временно хранятся кодированные данные (421), пока указанные кодированные данные (421) не используются декодером (450). В этот момент кодированные данные для кодированного кадра (421) и информация (432) о MMCO/RPS пересылаются из зоны (430) кодированных данных на декодер (450).

Так как декодирование продолжается, в зону (430) кодированных данных добавляются новые кодированные данные, а самые старые кодированные данные, остающиеся в зоне (430) кодированных данных, пересылаются на декодер (450).

Декодер (450) декодирует кодированный кадр (431) для создания соответствующего декодированного кадра (451). Когда это необходимо, при выполнении процесса

декодирования декодер (450) может использовать один или несколько ранее декодированных кадров (469) в качестве опорных кадров для межкадрового предсказания. Декодер (450) считывает указанные ранее декодированные кадры (469) из зоны (460) временного хранения декодированных кадров. Обычно декодер (450) включает в себя множество декодирующих модулей, которые решают такие задачи декодирования, как энтропийное декодирование, внутрикадровое предсказание, межкадровое предсказание с компенсацией движения, обратное квантование, обратные частотные преобразования и слияние плиток. Точный набор операций, выполняемых декодером (450), может изменяться в зависимости от формата сжатия.

Например, декодер (450) принимает кодированные данные для сжатого кадра или последовательности кадров и создает выходные данные, включающие в себя декодированный кадр (451). Имеющийся в декодере (450) буфер получает кодированные данные для сжатого кадра и в подходящее время делает полученные кодированные данные доступными для энтропийного декодера. Энтропийный декодер выполняет энтропийное декодирование кодированных квантованных данных, а также

кодированной вспомогательной информации, как правило, применяя операции, обратные энтропийному кодированию, выполненному в кодере. Компенсатор движения использует информацию о движении для одного или нескольких опорных кадров, формируя значения предсказания с компенсацией движения для любых кодированных (с использованием межкадрового кодирования) блоков восстанавливаемого кадра. Модуль

внутреннего предсказания может обеспечить пространственное предсказание значений отсчетов текущего блока на основе соседних, ранее восстановленных значений отсчетов, или, в случае внутреннего предсказания с ВС, предсказать значение отсчетов текущего блока, используя ранее восстановленные значения отсчетов из области внутрикадрового предсказания в данном кадре. Область внутрикадрового предсказания может быть

указана значением BV. Декодер (450) также восстанавливает значения остатков предсказания. Обратный квантователь выполняет обратное квантование для декодированных (с использованием энтропийного декодирования) данных. Например, декодер (450) устанавливает значения QR для картинки, плитки, слайса и/или другой части видео на основе синтаксических элементов в битовом потоке и выполняет обратное

квантование соответствующих коэффициентов преобразования. Средство обратного частотного преобразования преобразует квантованные данные в частотной области в данные пространственной области. Для блоков, полученных с использованием межкадрового предсказания, декодер (450) объединяет восстановленные значения

остатков предсказания со значениями предсказания с компенсацией движения. Декодер (450) может подобным же образом объединить значения остатков предсказания со значениями предсказания, полученными с использованием внутрикадрового предсказания. Контур компенсации движения в видеодекодере (450) включает в себя адаптивный деблокинг-фильтр для сглаживания разрывов в строках и/или столбцах на границах блоков в декодированном кадре (451). Вдобавок или в качестве альтернативы в операциях внутриконтурной фильтрации могут использоваться другие виды фильтрации (такие как деверберационная фильтрация, фильтрация ALF или SAO фильтрация; не показаны).

Зона (460) временного хранения декодированных кадров включает в себя множество зон (461, 462, ..., 46n) хранения (буферов кадров). Зона (460) хранения декодированных кадров представляет собой пример буфера декодированных картинок. Декодер (450) использует информацию (432) о MMCO/RPS для идентификации буфера (461, 462 и т.д.) кадров, в котором может храниться декодированный кадр (451). Декодер (450) запоминает декодированный кадр (451) в этом буфере кадров.

Секвенсор (480) вывода определяет, когда следующий кадр, создаваемый в порядке вывода, будет доступен в зоне (460) хранения декодированных кадров. Когда в зоне (460) хранения декодированных кадров появится следующий кадр (481), создаваемый в порядке вывода, он считывается секвенсором (480) вывода и выводится для адресата (490) вывода (например, на дисплей). В общем случае порядок, в котором выводятся секвенсором (480) вывода кадры из зоны (460) хранения декодированных кадров, может отличаться от порядка, в котором эти кадры декодировались декодером (450).

V. Примерные видеокодеры

На фигурах 5a и 5b представлена блок-схема обобщенного видеокодера (500), совместно с которым могут быть реализованы некоторые описанные здесь варианты осуществления. Кодер (500) принимает последовательность видеок картинок, включая текущую картинку, в виде входного видеосигнала (505) и создает кодированные данные в кодированном битовом потоке (595) видео в качестве выходных данных.

Кодер (500) основывается на блоках и использует блочный формат, который зависит от реализации. Блоки могут быть дополнительно разделены на другие составляющие, например, при предсказании, частотном преобразовании и/или энтропийном кодировании. Например, картинка может быть разделена на 64×64 блока, 32×32 блока или 16×16 блоков, которые, в свою очередь, можно разделить на более мелкие блоки значений отсчетов для кодирования и декодирования. В реализациях кодирования для стандарта HEVC кодер разделяет картинку на единицы CTU (CTB), CU (CB), PU (PB), и TU (TB).

Кодер (500) выполняет сжатие картинок, используя кодирование внутри одной картинки (внутреннее кодирование) и/или между картинками (внешнее кодирование). Многие компоненты кодера (500) используются как для внутреннего кодирования, так и для внешнего кодирования. Точный список операций, выполняемых этими компонентами, может существенно зависеть от типа сжимаемой информации.

Модуль (510) разделения на плитки выполняет необязательное разделение картинки на множество плиток одинакового размера или разных размеров. Например, модуль (510) разделения на плитки разделяет картинку по строкам и столбцам, которые вместе с границами картинки определяют горизонтальные и вертикальные границы плиток в картинке, где каждая плитка представляет собой прямоугольную область.

Средство (520) общего управления кодированием получает картинки для входного видеосигнала (505), а также сигнала обратной связи (не показан) от различных модулей

кодера (500). Средство (520) общего управления кодированием в общем случае обеспечивает управляющие сигналы (не показаны) для других модулей (таких как модуль (510) разделения на плитки, средство (530) преобразования/масштабирования/квантования, средство (535) масштабирования/обратного преобразования, средство (540) оценки внутреннего предсказания, средство (550) оценки движения и переключатель «внутр/внешн») для установки и изменения параметров кодирования во время кодирования. В частности, средство (520) общего управления кодированием может принять решение о том, использовать ли внутреннее предсказание с ВС во время кодирования, и принять решение о том, каким образом это сделать. Средство (520) общего управления кодированием также может оценить промежуточные результаты во время кодирования, например, выполнения анализа зависимости искажения от скорости передачи. Средство (520) общего управления кодированием создает общие управляющие данные (522), которые указывают решения, сделанные во время кодирования, так что соответствующий декодер может принять согласованные решения. Общие данные (522) управления подаются в средство форматирования заголовка/кодер (590) энтропийного кодирования.

Если текущая картинка предсказывается с использованием внешнего предсказания, компенсатор (550) движения оценивает движение блоков значений отсчетов текущей картинки входного видеосигнала (505) по отношению к одной или нескольким опорным картинкам. Буфер (570) декодированных картинок буферизирует одно или несколько восстановленных ранее кодированных картинок для использования в качестве опорных картинок. При использовании множества опорных картинок это множество опорных картинок может поступать с разных временных направлений или с одного и того же временного направления. Средство (550) оценки движения создает в качестве дополнительной информации данные (552) о движении, такие как данные вектора движения (MV), значения индекса режима слияния и данные по выбору опорных картинок. Данные (552) о движении подаются в средство (590) форматирования заголовка/энтропийного кодирования, а также на компенсатор (555) движения.

Компенсатор (555) движения применяет векторы движения к восстановленной опорной картинке (картинкам) из буфера (570) декодированных картинок. Компенсатор (555) движения создает предсказания с компенсацией движения для текущей картинки.

В отдельном тракте в декодере (500) средство (540) оценки внутреннего предсказания определяет, каким образом выполнить внутреннее предсказание для блоков значений отсчетов текущей картинки входного видеосигнала (505). Эта текущая картинка может быть целиком или частично закодирована с использованием внутреннего кодирования. Используя значения восстановления (538) текущей картинки для внутреннего пространственного предсказания, средство (540) оценки внутреннего предсказания определяет, каким образом выполнить пространственное предсказание значений отсчетов текущего блока текущей картинки, исходя из соседних ранее восстановленных значений отсчетов текущей картинки. Либо, если речь идет о внутреннем предсказании с ВС, средство (540) оценки внутреннего предсказания оценивает смещение значений отсчетов текущего блока для различных областей-кандидатов в текущей картинке. Значения BV могут быть созданы с использованием «пинг-понгового» подхода к предсказанию MV, как описывается ниже. Что касается внутреннего предсказания с ВС, то средство (540) оценки внутреннего предсказания может ограничить процесс выбора BV с использованием одного или нескольких ограничений, описанных ниже. Средство (540) оценки внутреннего предсказания может также ограничить использование внутреннего предсказания с ВС, когда разрешено ограниченное

внутреннее предсказание, как описано ниже, и может оценить восстановленные значения отсчетов в области перекрытия текущего блока в целях оценки BV.

Средство (540) оценки внутреннего предсказания создает в качестве вспомогательной информации данные (542) внутреннего предсказания, такие как информация, указывающая, используется ли пространственное предсказание или предсказание с ВС (например, значение флага для каждого внутреннего блока), направление режима предсказания (для внутреннего пространственного предсказания) и значения BV (для внутреннего предсказания с ВС). Данные (542) внутреннего предсказания подаются в средство (590) форматирования заголовка/энтропийного кодирования, а также в средство (545) внутреннего предсказания.

Согласно данным (542) внутреннего предсказания средство 545 внутреннего предсказания выполняет пространственное предсказание значений отсчетов текущего блока текущей картинки исходя из соседних ранее восстановленных значений отсчетов текущей картинки. Либо в случае внутреннего предсказания с ВС, средство (545) внутреннего предсказания предсказывает значения отсчетов текущего блока, используя ранее восстановленные значения отсчетов в области внутреннего предсказания, которая указана значением BV для текущего блока. Когда данные цветности для картинки имеют такое же разрешение, как данные яркости (например, когда формат представляет собой формат YUV 4:4:4 или формат RGB 4:4:4), значение, которое используется для блока цветности, может совпадать со значением BV, используемым для блока яркости. С другой стороны, когда данные цветности для картинки имеют уменьшенное разрешение по сравнению с данными яркости (например, когда форматом является формат YUV 4:2:0 или формат YUV 4:2:2), значение BV, которое используется для блока цветности, может быть уменьшено в масштабе и возможно округлено для согласования с разницей в разрешении цветности (например, для формата YUV 4:2:0 путем деления вертикальной и горизонтальной компонент значения BV на два и отбрасывания или округления их до целых значений), для формата YUV 4:2:2 путем деления горизонтальной компоненты значения BV на два и отбрасывания или округления его до целого числа).

Переключатель «внутр/внешн» выбирает значения предсказания с компенсацией движения или внутреннего предсказания для использования в качестве предсказания (558) для данного блока. Разница (если она имеет место) между блоком предсказания (558) и соответствующей частью исходной текущей картинки сигнала (505) входного видео обеспечивает значения остатка (518). Во время восстановления текущей картинки восстановленные значения остатков объединяют с результатами предсказания (558) для создания восстановленной версии (538) исходного контента из видеосигнала (505). Однако, при сжатии с потерями некоторая часть информации из видеосигнала (505) теряется.

В средстве (530) преобразования/масштабирования/квантования преобразователь частоты выполняет преобразование видеoinформации пространственной области в данные частотной области (например, спектральное, преобразование). Для видеокодирования на блочной основе частотный преобразователь применяет дискретное косинусное преобразование («DCT»), его целочисленную аппроксимацию или прямое блочное преобразование другого типа в блоки данных остатков предсказания (или данные о значениях отсчетов, если предсказание (558) дало нулевой результат), обеспечивая блоки коэффициентов частотного преобразования. Кодер (500) способен также указывать, что данный шаг преобразования пропускается. Средство масштабирования/квантования выполняет масштабирование и квантование коэффициентов преобразования. Например, квантователь применяет линейное

квантование с зоной нечувствительности к данным частотной области, используя размер шага, который изменяется от кадра к кадру, от плитки к плитке, от слайса к слайсу, от блока к блоку, в зависимости от частоты или на другой основе. Квантованные данные (532) о коэффициентах преобразования подаются в средство (590) форматирования заголовка/энтропийного кодирования.

В средстве (535) масштабирования/обратного преобразования средство масштабирования/обратного квантования выполняет обратное масштабирование и обратное квантование квантованных коэффициентов преобразования. Обратный частотный преобразователь выполняет обратное частотное преобразование, создавая блоки восстановленных остатков предсказания или значений отсчетов. Кодер (500) объединяет восстановленные остатки со значениями предсказания (558) (например, значения предсказания с компенсацией движения, значения внутреннего предсказания) для формирования восстановления (538).

Что касается внутреннего предсказания, то значения восстановления (538) могут быть направлены обратно в средство (540) оценки внутреннего предсказания и средство (545) внутреннего предсказания. Также значения восстановления (538) могут быть использованы для предсказания (с компенсацией движения) последующих картинок. Значения восстановления (538) кроме того могут подвергнуться фильтрации. Средство (560) управления фильтрацией определяет, каким образом выполнять фильтрацию с устранением блочности (деблокинг-фильтрацию) и фильтрацию с адаптивным смещением отсчетов (SAO) для значений восстановления (538) для данной картинки видеосигнала (505). Средство (560) управления фильтрацией создает данные (562) для управления фильтром, которые подаются в средство (590) форматирования заголовка/энтропийного кодирования и средство слияния/фильтр (фильтры) (565).

В средстве (565) слияния/фильтре (фильтрах) кодер (500) объединяет контент из разных плиток, формируя восстановленную версию картинки. Кодер (500) избирательно выполняет фильтрацию с устранением блочности и SAO фильтрацию в соответствии с данными (562) для управления фильтром, с тем, чтобы адаптивно сглаживать разрывы через границы в кадрах. Вдобавок или в качестве альтернативы могут использоваться другие виды фильтрации (такие как дереверберационная фильтрация или фильтрация ALF; не показаны). Фильтрация границ плиток выполняется избирательно или вообще не выполняется в зависимости от настроек кодера (500), а кодер (500) может предоставить синтаксис в кодированном потоке для указания о том, была ли применена указанная фильтрация. Буфер (570) декодированных картинок буферизирует восстановленную текущую картинку для ее использования при последующем предсказании с компенсацией движения.

Средство (590) форматирования заголовка/энтропийного кодирования выполняет форматирование и/или энтропийное кодирование общих управляющих данных (522), данных (532) квантованных коэффициентов преобразования, данных (542) внутреннего предсказания, данных (552) движения и данных (562) управления фильтром. Например, средство (590) форматирования заголовка/энтропийного кодирования используют контекстно-ориентированное двоичное арифметическое кодирование для энтропийного кодирования различных синтаксических элементов, таких как синтаксические элементы для значений BV. Средство (590) форматирования заголовка/энтропийного кодирования может использовать пинг-понговый подход к предсказанию BV, как описано ниже при кодировании значений BV.

Средство (590) форматирования заголовка/энтропийного кодирования обеспечивает кодированные данные в кодированном битовом потоке (595) видео. Формат

кодированного битового потока (595) видео может представлять собой вариант или расширение формата HEVC, формата Windows Media Video, формата VC-1, формата MPEG-х (например, MPEG-1, MPEG-2 или MPEG-4), формата H.26х (например, H.261, H.262, H.263, H.264) или другого формата.

- 5 В зависимости от реализации и требуемого типа сжатия возможно добавление, удаление, разбиение на множество модулей, объединение с другими модулями и/или замена подобными модулями кодера. В альтернативных вариантах осуществления кодеры с разными модулями и/или с другими конфигурациями модулей выполняют один или несколько описанных здесь приемов. В конкретных вариантах осуществления
- 10 кодеров, как правило, используют вариант или дополнительную версию кодера (500). Показанные здесь взаимосвязи между модулями в кодере (500) указывают общие информационные потоки в кодере; при этом другие взаимосвязи для простоты не показаны.

VI. Примерные видеodeкодеры

- 15 На фиг. 6 представлена блок-схема обобщенного декодера (600), совместно с которым могут быть реализованы некоторые описанные здесь варианты осуществления. Декодер (600) принимает кодированные данные в кодированном битовом потоке (605) видео и создает выходные данные, включающие в себя картинки для восстановленного видео (695). Формат кодированного битового потока (605) видео может представлять собой
- 20 вариант или расширение формата HEVC, формата Windows Media Video, формата VC-1, формата MPEG-х (например, MPEG-1, MPEG-2 или MPEG-4), формата H.26х (например, H.261, H.262, H.263, H.264) или другого формата.

- Декодер (600) основывается на блоках и использует блочный формат, который зависит от реализации. Блоки могут быть дополнительно разделены на другие
- 25 составляющие. Например, картинка может быть разделена на 64×64 блока, 32×32 блока или 16×16 блоков, которые, в свою очередь, можно разделить на более мелкие блоки значений отсчетов. В реализациях декодирования для стандарта HEVC картинка разделяется на единицы (блоки) CTU (CTB), CU (CB), PU (PB), и TU (TB).

- Декодер (600) выполняет распаковку картинок, используя внутреннее декодирование
- 30 и/или внешнее декодирование. Многие компоненты декодера (600) используются как для внутреннего декодирования, так и для внешнего декодирования картинок. Точный список операций, выполняемых этими компонентами, может зависеть от типа распаковываемой информации.

- Буфер получает кодированные данные в кодированном битовом потоке (605) видео
- 35 и делает полученные кодированные данные доступными для анализатора/энтропийного декодера (610). Анализатор/энтропийный декодер (610) выполняет энтропийное декодирование данных (полученных в результате энтропийного кодирования), применяя, как правило, операции, обратные энтропийному кодированию, выполненному в кодере (500) (например, контекстно-ориентированное двоичное арифметическое
- 40 декодирование). В результате выполнения синтаксического анализа и энтропийного декодирования анализатор/энтропийный декодер (610) создает данные (622) общего управления, данные (632) о квантованных коэффициентах преобразования, данные (642) внутреннего предсказания, данные (652) движения и данные (662) для управления фильтром. В частности, что касается данных (642) внутреннего предсказания, то
- 45 анализатор/энтропийный декодер (610) выполняет энтропийное декодирование синтаксических элементов для значений BV, например, используя контекстно ориентированное арифметическое декодирование. Анализатор/энтропийный декодер (610) может использовать пинг-понговый подход к предсказанию BV, как описано

ниже, при декодировании значений BV.

Средство (620) общего управления декодированием получает общие данные (622) управления и подает управляющие сигналы (не показаны) на другие модули (такие как средство масштабирования/обратный преобразователь (635), средство (645) внутреннего

5 предсказания, компенсатор (655) движения и переключатель «внутр/внешн») для установки и изменения параметров декодирования в ходе декодирования.

Если текущая картинка предсказывается с использованием внешнего предсказания, компенсатор (655) движения получает данные (652) движения, такие как данные вектора движения и данные выбора опорной картинки и значения индексов режима слияния.

10 Компенсатор (655) движения применяет векторы движения к восстановленной опорной картинке (картинкам) из буфера (670) декодированных картинок. Компенсатор (655) движения создает предсказания с компенсацией движения для кодированных (с применением межкадрового кодирования) блоков текущей картинки. В буфере (670) декодированных картинок запоминается одна или несколько ранее восстановленных

15 картинок для использования в качестве опорных картинок.

В отдельном тракте в декодере (600) средство (645) внутреннего предсказания получает данные (642) внутреннего предсказания, такие как информация, указывающая, используется ли при внутреннем предсказании пространственное предсказание или предсказание с ВС (например, значение флага для каждого внутреннего блока),

20 направление режима предсказания (для внутреннего пространственного предсказания) и значения BV (для внутреннего предсказания с ВС). В случае внутреннего пространственного предсказания средство (645) внутреннего предсказания, используя значения восстановления (638) текущей картинки в соответствии с данными режима предсказания, выполняет пространственное предсказание значений отсчетов текущего

25 блока текущей картинки исходя из соседних ранее восстановленных значений отсчетов текущей картинки. Либо, в случае внутреннего предсказания с ВС средство 645 внутреннего предсказания создает значения отсчетов текущего блока, используя ранее восстановленные значения отсчетов области внутреннего предсказания, которая указана значением BV для текущего блока.

30 Переключатель «внутр/внешн» выбирает значения предсказания с компенсацией движения или внутреннего предсказания для использования в качестве предсказания (658) для данного блока. Например, при использовании синтаксиса HEVC переключатель «внутр/внешн» может управляться на основе синтаксического элемента, кодированного для CU картинки, которая может содержать предсказанные (с применением внутреннего

35 предсказания) единицы CU и предсказанные (с использованием внешнего предсказания) единицы CU. Декодер (600) объединяет предсказание (658) с восстановленными значениями остатков для создания данных восстановления (638) контента из видеосигнала.

Для восстановления остатка средство масштабирования/обратный преобразователь

40 (635) принимает и обрабатывает данные (632) квантованных коэффициентов преобразования. В средстве масштабирования/обратном преобразователе (635) средство масштабирования/обратный квантователь выполняет обратное масштабирование и обратное квантование квантованных коэффициентов преобразования. Обратный частотный преобразователь выполняет обратное частотное преобразование, создавая

45 блоки восстановленных значений остатков предсказания или значений отсчетов. Например, обратный частотный преобразователь применяет обратное блочное преобразование к коэффициентам частотного преобразования, создавая данные значений отсчетов или данные остатков предсказания. Обратное частотное

преобразование может представлять собой обратное DCT, его целочисленную аппроксимацию или обратное частотное преобразование другого типа (например, обратное дискретное синусное преобразование или его целочисленная аппроксимация).

В случае внутреннего предсказания значения восстановления (638) могут быть направлены обратно в средство (645) внутреннего предсказания. Что касается внешнего предсказания, то значения предсказания (638) кроме того могут быть подвергнуты фильтрации. В блоке слияния/фильтре (фильтрах) (665) декодер (600) объединяет контент из разных плиток в восстановленную версию картинки. Декодер (600) избирательно выполняет фильтрацию с устранением блочности и SAO фильтрацию согласно данным (662) для управления фильтром и правилам для адаптации фильтра, с тем, чтобы адаптивно сгладить разрывы через границы в кадрах. Вдобавок, или в качестве альтернативы возможное использование фильтрации другого вида (например, дереверберационная фильтрация или ALF; не показано). Границы плиток могут быть избирательно отфильтрованы или вообще не фильтроваться в зависимости от настроек декодера (600) или синтаксической индикации в кодированных данных битового потока. Буфер (670) декодированных картинок буферизирует восстановленную текущую картинку для ее использования в последующем предсказании с компенсацией движения.

Декодер (600) также может включать в себя пост-обрабатывающий фильтр. Этот пост-обрабатывающий фильтр может выполнять дереверберационную фильтрацию, фильтрацию Винера, фильтрацию воспроизведения зернистости, SAO фильтрацию или фильтрацию другого вида.

В зависимости от реализации и требуемого типа распаковки в декодере возможно добавление модулей, удаление модулей, разбиение их на множество модулей, объединение с другими модулями и/или замена подобными модулями. В альтернативных вариантах осуществления декодеры с разными модулями и/или с другими конфигурациями модулей выполняют один или несколько описанных здесь приемов. В конкретных вариантах осуществления декодеров, как правило, используют вариант или дополнительную версию декодера (600). Показанные здесь взаимосвязи между модулями в декодере (600) указывают общие информационные потоки в декодере; при этом другие взаимосвязи для простоты не показаны.

VII. Режим внутреннего предсказания с блочным копированием

Этот раздел представляет различные признаки режима внутреннего предсказания с блочным копированием («BC»). Некоторые из этих признаков относятся к выбору значений блочного вектора («BV»), в то время как другие относятся к кодированию/ декодированию значений BV. Ряд других признаков могут облегчить выполнение внутреннего предсказания с BC, являющегося более эффективным с точки зрения показателя искажения в зависимости от скорости передачи и/или эффективности обработки кодирования и декодирования. В частности, внутреннее предсказание с BC может улучшить показатель искажения в зависимости от скорости передачи при кодировании конкретного «искусственно» созданного видеоконтента, такого как контент снимка экрана. Контент снимка экрана, как правило, включает в себя повторяющиеся структуры (например, графика, текстовые символы), которые обеспечивают возможности повышения эффективности внутреннего предсказания с BC. Контент снимка экрана обычно кодируют в формате (например, YUV 4:4:4 или RGB 4:4:4) с высоким разрешением дискретизации цветности, хотя также возможно его кодирование в формате с более низким разрешением дискретизации цветности (например, YUV 4:2:0, YUV 4:2:2).

A. Режим внутреннего предсказания с BC - Введение

В режиме внутреннего предсказания с ВС значения отсчетов текущего блока картинки предсказываются с использованием значений отсчетов в одной и той же картинке. Вектор BV указывает смещение от текущего блока до области картинки, которая включает в себя значения отсчетов, использованных для предсказания. Как правило, значения отсчетов, использованных для предсказания, представляют собой ранее восстановленные значения отсчетов. Вектор BV передается в битовом потоке в качестве сигнализации. Внутреннее предсказание с ВС является формой внутреннего предсказания для картинки, то есть, внутреннее предсказание с ВС для блока картинки не использует какие-либо значения отсчетов, отличные от значений отсчетов в той же картинке.

На фиг. 7 показано внутреннее предсказание с ВС для текущего блока (730) текущего кадра (710). Текущим блоком может быть блок кодирования («СВ») единицы кодирования («CU»), блок предсказания («PB») единицы предсказания («PU»), блок преобразования («TB») единицы преобразования («TU») или другой блок. Размер текущего блока может составлять 64×64 , 32×32 , 16×16 , 8×8 или некоторый другой размер. В более общем случае размер текущего блока составляет $m \times n$, где m и n являются каждое целым числом, и где m и n могут быть равны друг другу или могут иметь разные значения. В качестве альтернативы текущий блок может иметь другую форму (например, область кодированного видеообъекта непрямоугольной формы).

Вектор BV (740) указывает смещение (или сдвиг) от текущего блока (730) до области (750) картинки, которая включает в себя значения отсчетов, использованных для предсказания. Положим, что верхнее левое положение текущего блока - это положение (x_0, y_0) в текущем кадре, и положим, что верхнее левое положение области внутреннего предсказания - это положение (x_j, y_j) в текущем кадре. Вектор BV указывает смещение $(x_j - x_0, y_j - y_0)$. Например, если верхним левым положением текущего блока является положение (320, 256), а верхним левым положением области внутреннего предсказания является положение (295, 270), то значение BV составит (-25, 14). В этом примере отрицательное горизонтальное смещение указывает на положение влево от текущего блока, а отрицательное вертикальное смещение указывает на положение выше текущего блока.

В некоторых примерных реализациях область (750) внутреннего предсказания ограничена пределами того же слайса или плитки, что текущий блок (730). При указанном внутреннем предсказании с ВС не используются значения отсчетов в других слайсах или плитках. На местоположение области 750 внутреннего предсказания могут быть наложены одно или несколько других ограничений (например, для диапазона поиска, касающегося использования восстановленных значений отсчетов, кодированных (с применением внешнего кодирования) блоков).

Блоком с внутренним режимом предсказания с ВС может быть блок СВ, РВ или другой блок. Когда это блок СВ, вектор BV для этого блока может быть передан в качестве сигнализации на уровне CU (а для других блоков СВ в CU используют тот же самый BV или его масштабированную версию). Либо, когда блок является РВ блоком, вектор BV для этого блока может быть передан в качестве сигнализации на уровне PU (а для других блоков РВ в PU используют тот же самый BV или его масштабированную версию). В более общем случае вектор BV для блока внутреннего предсказания с ВС передается в качестве сигнализации на соответствующем синтаксическом уровне для данного блока.

Операции блочного копирования для предсказания согласно режиму внутреннего предсказания с ВС могут выполняться на уровне СВ (когда вектор BV передается в

качестве сигнализации для каждого СВ) или РВ (когда вектор BV передается в качестве сигнализации для каждого РВ). Например, положим, что блок СВ 16×16 имеет один блок РВ 16×16. Вектор BV (для РВ) применяется к блочной копии области 16×16. Когда область внутреннего предсказания ограничена, чтобы избежать перекрытия

5 предсказываемого блока 16×16, вектор BV имеет величину (абсолютное значение) по меньшей мере 16 по горизонтали или вертикали.

В качестве альтернативы, операции блочного копирования могут выполняться на уровне блоков ТВ в блоке РВ или СВ даже в том случае, когда BV передается в качестве сигнализации для РВ или СВ. Таким образом, вектор BV, применяемый для ТВ, может

10 обращаться к позициям других блоков ТВ в том же самом РВ или СВ. Например, положим, что блок СВ 16×16 имеет один блок РВ 16×16, но он разделен на шестнадцать блоков ТВ 4×4 с целью кодирования/декодирования остатка. Вектор BV (для блока РВ) применяется для блочного копирования области 4×4 для первого ТВ в порядке растрового сканирования, затем этот же BV используется для блочного копирования

15 области 4×4 для второго ТВ в порядке растрового сканирования и т.д. Область 4×4, используемая в операциях ВС для блока ТВ, может включать в себя положения в ранее восстановленных блоках ТВ в том же самом СВ после объединения значений остатков с предсказанными значениями для этих ранее восстановленных блоков ТВ. (Вектор BV еще не привязан к каким-либо местам в блоке ТВ, для которого выполняется

20 предсказание). Применение операций ВС на уровне ТВ облегчает использование векторов BV, имеющих относительно малую величину.

Операции внутреннего предсказания с ВС для блоков цветности единицы CU обычно соответствуют операциям внутреннего предсказания с ВС для блока яркости единицы CU. Обычно сегментация блоков РВ цветности и блоков ТВ цветности напрямую

25 соответствует сегментации блоков РВ яркости и блоков ТВ яркости в единице CU. При использовании формата видео YUV 4:4:4 размеры блоков РВ и ТВ цветности соответствуют размерам соответствующих блоков РВ и ТВ яркости. Когда форматом видео является формат YUV 4:2:0, ширина блоков РВ и блоков ТВ цветности составляет половину ширины, а высота составляет половину высоты соответствующих блоков РВ

30 и ТВ яркости. Однако, если блок ТВ яркости имеет минимальную размерность преобразования, то используется один блок ТВ цветности с этой минимальной размерностью преобразования. При использовании формата YUV 4:2:2 ширина блоков РВ и ТВ цветности составляет половину ширины соответствующих блоков РВ и ТВ яркости.

В некоторых реализациях для CU с внутренним предсказанием с ВС при внутреннем предсказании с ВС для блока цветности в PU используется то же самое значение BV, что и во внутреннем предсказании с ВС для блока яркости в PU, возможно после масштабирования и округления, когда данные цветности уменьшили разрешающую

40 способность по сравнению с данными яркости (например, при использовании формата YUV 4:2:0 или формата YUV 4:2:2). В качестве альтернативы, возможна передача в качестве сигнализации других значений BV для блока яркости и блоков цветности единицы PU.

В некоторых реализациях, если режимом предсказания блока яркости единицы PU является внутреннее предсказание с ВС, то режимом предсказания для блоков цветности

45 единицы PU также будет внутреннее предсказание с ВС. Например, сообщение о режиме предсказания передается в качестве сигнализации для PU. В качестве альтернативы, режимом предсказания может быть внутреннее предсказание с ВС для блока яркости или блоков цветности единицы PU, но не для их обоих.

В. Ограничение поиска BV для режима внутреннего предсказания с ВС

В некоторых примерных реализациях кодер ограничивает диапазон BV в соответствии с одним или несколькими ограничениями. Благодаря ограничению диапазона BV область восстановленных значений отсчетов, к которой обращаются путем быстрого доступа к памяти для внутреннего предсказания с ВС во время кодирования и декодирования, может быть уменьшена, что открывает дорогу к снижению затрат на реализацию.

На фиг. 8 показаны примерные ограничения на диапазон поиска значений BV. Вдобавок к текущему блоку (830) текущего кадра (810) на фиг. 8 показан диапазон поиска, определенный двумя блоками СТВ (820, 822). Текущий блок СТВ (820) является частью текущей СТУ и включает в себя текущий блок (830). При использовании СТВ (822) слева текущий СТВ (820) определяет диапазон поиска, в котором могут быть найдены допустимые векторы BV для текущего блока (830). Векторы BV (842, 844) относятся к областям, находящимся вне диапазона поиска, так что эти значения BV (842, 844) недопустимы.

В некоторых примерных реализациях диапазоном поиска для значений BV для текущего блока является текущий блок СТВ и блок СТВ слева от него. Например, блок СТВ может иметь размер 64×64 , 32×32 или 16×16 значений отсчетов, что дает диапазон поиска 128×64 , 64×32 или 32×16 значений отсчетов. Для внутреннего предсказания с ВС для текущего блока используют только одно значение отсчета в текущем блоке СТВ и блоке СТВ слева от него. Это упрощает реализацию кодера благодаря ограничению процесса поиска. Это также упрощает реализацию декодера благодаря ограничению количества значений отсчетов, которое буферизирует декодер в быстродействующей памяти для внутреннего предсказания. Другое ограничение состоит в том, что внутреннее предсказание не может обращаться к значениям отсчетов из другого слайса или плитки. Для текущего блока $m \times n$ с верхним левым положением (x_0, y_0) и СТВ, имеющих каждый размерность $CTB_{sizeX} \times CTB_{sizeY}$, кодер может проверить указанные ограничения для двумерного BV, имеющего горизонтальную компоненту BV[0] и вертикальную компоненту BV[1] следующим образом.

$$BV[0] \geq -((x_0 \% CTB_{sizeX}) + CTB_{sizeX})$$

$$BV[1] \geq -(y_0 \% CTB_{sizeY})$$

Значения отсчетов в положениях (x_0, y_0) , $(x_0 + BV[0], y_0 + BV[1])$ и $(x_0 + BV[0] + m - 1, y_0 + BV[1] + n - 1)$ должны быть в одном и том же слайсе.

Значения отсчетов в положениях (x_0, y_0) , $(x_0 + BV[0], y_0 + BV[1])$ и $(x_0 + BV[0] + m - 1, y_0 + BV[1] + n - 1)$ должны быть из одной и той же плитки.

На фиг. 9 показан прием (900) для кодирования с режимом внутреннего предсказания с ВС с одним или несколькими ограничениями на выбор значений вектора BV. Прием (900) может быть выполнен кодером, таким как был описан со ссылками на фиг. 3 или фигуры 5a-5b.

Для запуска процесса кодер определяет (910) вектор BV для текущего блока картинки. Текущим блоком может быть СВ, РВ или другой блок. Вектор BV указывает смещение до области внутри картинки. При определении BV кодер проверяет выполнение одного или нескольких ограничений.

Согласно одному возможному ограничению кодер проверяет диапазон значений отсчетов, используемых для внутреннего предсказания с ВС. Кодер может проверить, что предполагаемая область внутреннего предсказания находится в пределах диапазона, определенного текущим блоком СТВ и одним или несколькими другими блоками СТВ

(например, СТВ слева от текущего СТВ). Например, когда вектор BV имеет первую компоненту $BV[0]$ и вторую компоненту $BV[1]$, текущий блок имеет верхнее левое положение в положении (x_0, y_0) , а каждый из блоков СТВ имеет ширину CTB_{width} и высоту CTB_{height} , ограничение удовлетворяется: если $BV[0] \geq -((x_0 \% CTB_{width}) + CTB_{width})$ и $BV[1] \geq -(y_0 \% CTB_{height})$. Кодер может аналогичным образом проверить ограничение сверху на значения $BV[0]$ и $BV[1]$ в диапазоне поиска: $BV[0] < (CTB_{width} - m - (x_0 \% CTB_{width}))$ и $BV[1] < (CTB_{height} - n - (y_0 \% CTB_{height}))$. В качестве альтернативы, диапазон поиска может включать в себя больше или меньше блоков СТВ, или диапазон поиска может быть определен каким-либо другим путем.

Согласно еще одному возможному ограничению кодер ограничивает поиск текущим слайсом и плиткой (то есть, текущий блок и область поиска являются частью не более чем одного слайса картинки, и не более одной плитки картинки). Кодер может проверить, являются ли верхнее левое положение текущего блока, верхнее левое положение пригодной области внутреннего предсказания и нижнее правое положение пригодной области внутреннего предсказания частью одного слайса и одной плитки. Например, ограничение выполняется, если: (x_0, y_0) , $(x_0 + BV[0], y_0 + BV[1])$ и $(x_0 + BV[0] + m - 1, y_0 + BV[1] + n - 1)$ являются частью одного слайса и одной плитки.

В качестве альтернативы, кодер может проверить другие и/или дополнительные ограничения.

Кодер выполняет (920) внутреннее предсказание с ВС для текущего блока, используя вектор BV. Например, кодер выполняет внутреннее предсказание с ВС для всего текущего блока. Либо кодер выполняет внутреннее предсказание с ВС для множества блоков, связанных с текущим блоком (например, для множества блоков ТВ на поблочной основе, где блоки ТВ связаны с текущим блоком РВ, который имеет BV).

Кодер кодирует (930) вектор BV. Например, кодер кодирует (930) вектор BV, как описано ниже. Кодер может повторить выполнение приема (900) для другого блока в режиме внутреннего предсказания с ВС.

При использовании внутреннего предсказания с ВС кодер и декодер используют восстановленные значения отсчетов. Невосстановленные значения отсчетов могут быть представлены как части картинки, которые еще не были кодированы и восстановлены. Во избежание использования невосстановленных значений отсчетов для внутреннего предсказания с ВС кодер может установить ограничение на допустимые значения вектора BV, с тем чтобы для внутреннего предсказания с ВС в соответствии с BV использовались только действительно ранее восстановленные значения отсчетов.

В некоторых примерных реализациях кодер проверяет значение BV, учитывая порядки z-сканирования текущего блока и блока, содержащего нижнее правое положение предполагаемой области внутреннего предсказания. В частности, кодер проверяет, меньше ли порядок z-сканирования блока, содержащего позицию $(x_0 + BV[0] + m - 1, y_0 + BV[1] + n - 1)$, чем порядок z-сканирования блока, содержащего (x_0, y_0) . Если да, то блок, который содержит нижнее правое положение области внутреннего предсказания, был ранее восстановлен (и, следовательно, может остаться в области внутреннего предсказания). Вектор BV также удовлетворяет по меньшей мере одному из условий $BV[0] + m \leq 0$ и/или $BV[1] + n \leq 0$, что гарантирует отсутствие перекрытия области внутреннего предсказания и текущего блока.

Порядок z-сканирования следует заданной последовательности блоков, на которые разделена картинка. На фиг. 10 показан примерный порядок (1000) z-сканирования

для текущего блока (1020) и блоков, которые могли бы включать в себя нижнее правое положение области внутреннего предсказания для пригодного вектора BV . Текущим блоком (1030) может быть блок CB , блок PB или другой блок. Порядки z -сканирования обычно выделяются для блоков последовательно слева направо по строке, с повторением в последующих строках сверху вниз. Когда блок разделен на отдельные части, порядки z -сканирования присваивают в разделенном блоке рекурсивно. Для реализаций кодирования/декодирования согласно стандарту HEVC, порядок z -сканирования определяет сканирование от CTB к CTB с использованием шаблона растрового сканирования CTB (слева направо в одной строке CTB , с повторением на последующих строках CTB сверху вниз). Если блок CTB разделен на отдельные части, порядок z -сканирования соответствует шаблону растрового сканирования для блоков CB квадродерева в разделенном CTB . И наконец, если CB разделен на отдельные части (например, на множество блоков CB или множество блоков PB), то порядок z -сканирования соответствует шаблону растрового сканирования для блоков внутри разделенного блока CB .

В качестве альтернативы, когда внутреннее предсказание с BC может выполняться на поблочной (TB) основе, кодер и декодер могут проверить наличие возможного перекрытия между областью внутреннего предсказания и текущим блоком (TB), а затем используют результаты проверки для принятия решения о том, следует ли разделить текущий TB на несколько TB меньшего размера для применения операций внутреннего предсказания с BC . Положим, что размер блока TB составляет $m \times n$, где m и n могут быть равны друг другу или иметь разные значения. Если $HV[0] > -m$ и $HV[1] > -n$, то область внутреннего предсказания перекрывает текущий блок TB размером $m \times n$, что является проблемой, если текущий блок TB разделен на блоки TB меньшего размера для применения операций внутреннего предсказания с BC . Таким образом, если $BV[1] > -m$ и $BV[1] > -n$, то кодер и декодер разделяют текущий блок TB на блоки меньшего размера. Точно такое же условие проверяется (например, проверяется рекурсивно) для блоков TB меньшего размера, которые могут быть дополнительно разделены, если $BV[0] > -m$ и $BV[1] > -n$, даже для меньших значений m и n после разделения.

Например, положим, что BV для PB составляет $(-9, -5)$, а текущий TB представляет собой блок 32×32 . Кодер и декодер определяют, что $-9 > -32$ и $-5 > -32$, что указывает на то, что область внутреннего предсказания (чей верхний левый угол расположен в точке $(-9, -5)$) перекрывает текущий блок TB размером 32×32 . Кодер и декодер разделяют блок TB размером 32×32 на четыре TB размером 16×16 . Для каждого из блоков TB размером 16×16 кодер и декодер определяют, что $-9 > -16$ и $-5 > -16$, что указывает на перекрытие областью внутреннего предсказания (чей верхний левый угол расположен в точке $(-9, -5)$) текущего блока TB размером 16×16 . Кодер и декодер разделяют каждый блок TB размером 16×16 последовательно на четыре TB размером 8×8 . Для блока TB размером 8×8 BV для $(-9, -5)$ не представляет проблемы, так как блок TB 8×8 не подвергается дальнейшему разделению.

В этом сценарии, когда разделение TB происходит из-за значения BV и размера TB , кодер может пропустить передачу значения флага, который, в противном случае, сигнализировал бы о том, разделять ли текущий TB на TB меньшего размера. В битовом потоке кодированных данных недостает значения флага, указывающего декодеру на необходимость разделения текущего TB на TB меньшего размера. Вместо этого декодер может сделать вывод, что TB следует разделить из-за значения BV и размера TB . Это может сэкономить биты, которые, в противном случае, были бы потрачены на передачу сигнальной информации о разделении блоков TB .

С. Кодирование и декодирование значений BV

Для всех значений BV для блоков, кодированных с использованием внутреннего предсказания с ВС, может понадобиться значительное количество бит. Значения BV могут быть подвергнуты энтропийному кодированию для уменьшения битрейта. Для дальнейшего уменьшения битрейта для значений BV кодер может использовать предсказание значений BV. Значения BV часто обладают избыточностью, то есть, значение BV для данного блока часто близко или даже совпадает со значениями BV предшествующих блоков в данной картинке. Для предсказания BV значение BV для данного блока предсказывают с использованием предиктора BV. Разность (или приращение BV) между значением BV для данного блока и предиктором BV подвергается затем энтропийному кодированию. Как правило, приращение BV вычисляют для горизонтальной и вертикальной компонент значения BV и предиктора BV. При удовлетворительном выполнении предсказания BV приращение BV имеет распределение вероятности, поддерживающее эффективное энтропийное кодирование. Кодер и декодер могут использовать базовое предсказание BV и/или режим слияния/конкуренцию BV при кодировании/декодировании значений BV.

Для базового предсказания BV во время кодирования и декодирования значение для текущего блока может быть предсказано на основе значений BV одного или нескольких предшествующих блоков. Например, значение BV соседнего блока (например, блока слева от текущего блока) можно использовать для определения предиктора BV для значения BV текущего блока. Если значение BV текущего блока составляет $(-80, -24)$, а предиктора BV $(-80, -32)$, то приращение BV, равное $(0, 8)$ подвергается энтропийному кодированию. Либо предиктор BV для значения BV текущего блока может представлять собой (покомпонентную) медиану или среднее значений BV для множества соседних блоков (например, для блоков, находящихся левее, выше, а также выше и левее текущего блока).

Во время декодирования декодер получает и выполняет энтропийное декодирование энтропийно-кодированного приращения BV (если оно имеется) для значения BV. Декодер также определяет предиктор BV для данного значения BV. (Предиктор BV, определенный декодером, совпадает с предиктором BV, определенным кодером). Затем восстановленная разность BV (если она имеется) комбинируется с предиктором BV.

Кодер и декодер могут использовать предиктор BV по умолчанию, когда действительное значение BV недоступно из предшествующего блока (например, при определении значения BV для первого блока, полученного с использованием внутреннего предсказания с ВС, в данной единице STU). Например, предиктором BV по умолчанию может быть $(0, 0)$. Либо предиктором BV по умолчанию может быть $(-W, 0)$ или $(-2*W, 0)$, где W - ширина текущего блока. Либо предиктором BV по умолчанию может быть $(0, -H)$ или $(0, -2*H)$, где H - высота текущего блока. По сравнению с нулевым предиктором BV по умолчанию предиктор BV по умолчанию с ненулевой компонентой ближе к значению BV текущего блока, что обеспечивает более эффективное энтропийное кодирование приращения BV. Либо компонента предиктора BV по умолчанию может иметь фиксированное ненулевое значение (например, 8 или 16).

В любом случае, когда значение BV, пригодное для текущего блока, ограничено конкретным диапазоном поиска (например, текущий СТВ и СТВ слева от текущего СТВ, как было описано в предыдущем разделе), и значения BV не могут привести к перекрытию текущего блока и области внутреннего предсказания, в некоторых случаях BV для текущего блока может иметь только одно пригодное значение. Например, если

(1) текущий СТВ имеет единственный СВ в виде текущего блока, (2) диапазон поиска для значений BV представляет собой текущей СТВ и СТВ слева от него, и (3) перекрытие между текущим блоком и его областью внутреннего предсказания не допускается, то тогда значение BV для указанного единственного СВ должно составлять $(-W, 0)$. Другие значения BV относятся к местам вне диапазона поиска или к местам в текущем блоке. В этом случае кодер может пропустить передачу значения BV (или приращение BV) для текущего блока в качестве сигнализации. Декодер может проверить выполнение условий для идентификации данной ситуации и получить значение BV $(-W, 0)$ без выполнения синтаксического анализа и декодирования какого-либо значения BV (или приращения BV) для текущего блока.

Либо кодер и декодер определяют одно или несколько пригодных значений BV для текущего блока из числа значений BV, использованных для восстановленных блоков, которые соседствуют в пространстве с текущим блоком (например, блок слева от текущего блока, блок над текущим блоком и т.д.). Пригодное значение (значения) BV может также включать в себя одно или несколько значений BV, использованных для восстановленных блоков, являющихся временными соседями текущего блока, где блок, временно являющийся соседом, расположен в соответствующей позиции в качестве текущего блока в другой картинке (например, в той же самой позиции или перекрывающейся позиции). Список пригодных значений BV может быть определен по установленным правилам во время кодирования и декодирования для исключения избыточных значений BV. Во время кодирования кодер может в качестве сигнализации передать один или несколько синтаксических элементов, указывающих, какое из пригодных значений BV использовать в качестве предиктора BV для текущего блока. В некоторых режимах этот предиктор BV может быть использован в качестве значения BV для текущего блока, что обеспечивает эффективное «слияние» значения BV текущего блока со значением BV соседнего блока, обеспечивая пригодное значение BV. Либо кодер может определить и выполнить кодирование приращения BV на основе значения BV и предиктора BV. Во время декодирования декодер может получить один или несколько синтаксических элементов, указывающих, какое пригодное значение (значения) BV использовать в качестве предиктора для текущего блока. В некоторых режимах предиктор BV может быть использован в качестве значения BV для текущего блока, что гарантирует эффективное «слияние» значения BV текущего блока со значением BV соседнего блока, обеспечивая пригодное значение BV. Либо декодер может получить и выполнить декодирование разности BV, которую декодер объединяет с предиктором BV для восстановления значения BV. Может быть обеспечен режим «пропуска» BV или «направления» BV, в котором предиктор BV (выбранный по установленному правилу) используют в качестве значения BV текущего блока без передачи в качестве сигнализации значений остатков для текущего блока. Примеры предсказания BV с использованием пинг-понгового подхода, при котором кодер или декодер осуществляют выбор из пары пригодных значений BV при предсказании значения BV для блока, описаны ниже.

Д. Примерные признаки внутреннего предсказания с ВС при использовании комбинированных реализаций

Как было отмечено выше, предшествующие признаки внутреннего предсказания с ВС могут быть использованы по отдельности и индивидуально. Либо эти признаки внутреннего предсказания с ВС могут быть использованы в комбинации.

Например, в одной комбинированной реализации, которая обычно соответствует синтаксису HEVC, значение BV передается в качестве сигнализации для PU (которая

может представлять собой CU или являться ее частью). Единица PU может включать в себя одну или несколько единиц TU. Процессы внутреннего предсказания с ВС выполняются на уровне блоков ТВ на поблочной (ТВ) основе с использованием значения BV, переданного в качестве сигнализации для PU. (Во всех ТВ используется одно и то же значение BV, а при внутреннем предсказании с ВС для текущего ТВ можно использовать восстановленные значения отсчетов других, более ранних ТВ той же самой CU) Значение BV можно предсказать с использованием значений BV одной или нескольких соседних PU (например, с использованием пинг-понгового подхода к предсказанию BV). Выбор значений BV ограничен: (а) с тем чтобы запретить кодеру выбор значений BV, которые могут инициировать появление значений отсчетов, подлежащих использованию в качестве опорных, которые будут находиться в областях, где еще не выполнялось кодирование/восстановление (то есть, значения отсчетов области внутреннего предсказания для текущего блока ТВ должны находиться в областях, покрываемых другими блоками ТВ, которые предшествуют текущему ТВ в порядке декодирования/битового потока; то есть, для данного ТВ значение BV ограничено обращением к области, находящейся вне данного ТВ); (б) для сокращения необходимого объема памяти в декодере (например, путем ограничения обращений в соответствии со значениями BV, которые должны находиться в текущем блоке СТВ и одном или двух блоках СТВ слева от текущего СТВ); и (с) для запрещения обращений в соответствии со значениями BV вне текущего слайса, вне текущей плитки или вне данной картинки.

VIII. Нововведения в предсказание блочного вектора

Согласно одному описанному здесь аспекту нововведений кодер и декодер используют предсказание блочного вектора («BV») согласно «пинг-понговому» подходу. В частности, приведенные в качестве примеров пинг-понговые подходы к предсказанию BV в некоторых сценариях повышают эффективность кодирования при кодировании экранного видеоконтента.

В примерных пинг-понговых подходах кодер и декодер поддерживают пару пригодных значений BV в качестве информации о состоянии. На фиг. 11 показана пара пригодных значений BV в буферах (1100) BV, которые можно использовать в примерных пинг-понговых подходах. В буферах (1100) BV, а именно, в буфере 0 запоминается первое начальное пригодное значение BV, обозначенное как BVinit_0. В буфере 1 хранится второе начальное пригодное значение BV, обозначенное как BVinit_1. В общем случае значение BV, являющееся переменным для буфера 0 для BV, можно обозначить как PBV0, а значение BV, являющееся переменным для буфера 1 для BV, можно обозначить как PBV1.

На фиг. 12 показана другая структура (1200) данных, где может храниться пара пригодных значений BV согласно примерным пинг-понговым подходам. В структуре (1200) данных хранится первое начальное пригодное значение BV, обозначено как BVinit_0 и связано со значением индекса idx0. Второе начальное пригодное значение BV (BVinit_1) связано со значением idx1 индекса в структуре (1200) данных. В общем случае переменное значение BV, связанное с idx0, можно обозначить как PBV0, а переменное значение BV, связанное с idx1, можно обозначить как PBV1.

В качестве альтернативы, кодер и декодер используют другую структуру данных для отслеживания пары пригодных значений BV для предсказания BV согласно пинг-понговому подходу.

Согласно примерным пинг-понговым подходам кодер из пары пригодных значений BV выбирает предиктор BV, чтобы использовать его для значения BV единицы PU.

Например, при кодировании PU с использованием предсказания BC с выбранным пригодным значением BV кодер выбирает пригодное значение BV, для которого область внутреннего предсказания лучше всего соответствует PU на основе некоторого показателя (например, сумма абсолютных приращений, среднеквадратическая ошибка).

5 Либо, когда PU имеет значение BV (идентифицированное посредством оценки BV), которое будет использоваться для внутреннего предсказания с BC, кодер выбирает пригодное значение BV, которое лучше всего соответствует значению BV для данной PU. Такой выбор обеспечивает минимальные приращения BV, что повышает эффективность энтропийного кодирования.

10 Для значения BV кодер в качестве сигнализации передает значение флага, указывающее выбранное значение из пары пригодных значений BV, с тем чтобы указать декодеру, какое из пары пригодных значений BV следует использовать в качестве предиктора BV. Это значение флага может быть закодировано (с использованием энтропийного кодирования) или передано в качестве сигнализации в виде значения
15 фиксированной длины. Значение флага может быть передано (по согласованию в качестве сигнализации в битовом потоке в виде отдельного синтаксического элемента, либо может быть передано в битовом потоке совместно с другим синтаксическим элементом.

Когда каждое пригодное значение BV из пары пригодных значений BV скорее всего
20 равно предиктору BV, возможна эффективная передача в качестве сигнализации значения флага в виде значения с фиксированной длиной (например, 1 бит) или с использованием двоичного арифметического кодирования в байпасном режиме (в предположении, что оба пригодных значения равновероятны). Первое значение (например, 0) указывает на выбор первого пригодного значения BV, а второе значение
25 (например, 1) указывает на выбор второго пригодного значения BV. Значение флага также может передаваться в качестве сигнализации в виде значения фиксированной длины, когда одно из пары вероятных значений BV вероятнее всего больше другого, но эффективность кодирования при этом может уменьшиться.

Когда одно из пригодных значений BV в упомянутой паре выбирается в качестве
30 предиктора с большей вероятностью, для передачи значений флага в качестве сигнализации может оказаться более эффективным использование контекстно-ориентированного арифметического кодирования. Например, вероятностный контент для значений флага может быть инициализирован с условием равной вероятности для упомянутых двух возможных значений флага при запуске кодирования или
35 декодирования для слайса с последующим обновлением во время кодирования/декодирования, чтобы учесть появление разных значений флага. При наличии отдельного контекста для предиктора BV значения флага могут повысить эффективность кодирования, но при этом возрастает сложность кодирования и декодирования из-за контекстного моделирования и т.д. Как правило, пригодное значение BV, выбор
40 которого ожидается с более высокой частотой, запоминается в первом буфере BV (например, буфер 0 для BV) и указывается с первым значением флага, а пригодное значение BV, выбор которого ожидается реже, запоминается в другом буфере BV (например, буфер 1 для BV) и указывается с другим значением флага. Таким образом, значения флага, указывающие первый буфер BV, могут усредненно кодироваться с
45 использованием меньшего количества дробных бит при арифметическом кодировании. Однако, для некоторого контента более общие пригодные значения BV могут запоминаться в другом буфере BV (например, буфер BV 1) с помощью контекстного моделирования, учитывающего изменение распределения предиктора BV.

Декодер выполняет синтаксический анализ/декодирование значения флага для значения BV и использует это значение флага для определения того, какое из пары пригодных значений BV является предиктором BV для данного значения BV. Декодер в качестве предиктора BV использует выбранное пригодное значение BV.

5 Кодер и декодер выполняют отслеживание и обновление значений BV в паре пригодных значений BV. В предыдущих пинг-понговых подходах при создании (или восстановлении) нового значения BV старое пригодное значение BV (PBV1) в паре заменяется новым значением BV (PBV0) в паре, а новое значение BV (PBV0) в паре заменяется только что созданным/восстановленным значением BV (BVnew) (то есть, 10 PBV1=PBV0, а PBV0=BVnew). Пара пригодных значений BV обновляется при создании или восстановлении каждого значения BV.

В описанных здесь примерных пинг-понговых подходах кодер и декодер могут автоматически обновлять пару пригодных значений BV для включения значения BV после создания или восстановления значения BV. Однако это может привести к тому, 15 что в паре пригодных значений BV будет два идентичных пригодных значения для BV. В этой ситуации кодер (определяющий идентичность обоих пригодных значений BV) может пропустить передачу значения флага в качестве сигнализации, которое указывает на выбор из пары пригодных значений BV, а декодер (распознающий, что оба пригодных значения BV идентичны) может пропустить операцию синтаксического анализа значения 20 флага, вместо простого использования любого из идентичных пригодных значений BV в качестве предиктора BV.

Либо в вышеописанных примерных пинг-понговых подходах кодер и декодер могут избирательно обновлять пару пригодных значений BV (и буфер BV), чтобы избежать идентичных пригодных значений BV. Кодер или декодер обновляет пару пригодных 25 значений BV только тогда, когда новое значение BV не равно пригодному значению BV, оставшемуся в указанной паре. Если новое значение BV равно пригодному значению BV, которое останется в указанной паре, кодер или декодер пропускает операцию обновления. Например, кодер или декодер избирательно обновляет пару пригодных значений BV (PBV0 и PBV1) для включения нового значения BVnew следующим образом

```
30  если (BVnew != PBV0) {
    PBV1=PBV0;
    PBV0=BVnew;
  } в противном случае {
    не совершать никаких действий;
35  }
```

Путем проверки выполнения условия BVnew != PBV0 кодер и декодер не допускают ситуацию, при которой PBV0 равно PBV1, и в этом случае значение флага, указывающее предиктор BV, расходуетя впустую. С другой стороны, проверка этого условия усложняет предсказание BV, что может оказаться неприемлемым при заданных 40 ресурсных ограничениях или удовлетворяется за счет повышения эффективности кодирования.

Положим, например, что PBV0 составляет (-12, 0), а PBV1 составляет (-16, 0). Если BVnew составляет (-17, 0), то (-16, 0) заменяется на (-12, 0) для PBV1, и (-12, 0) заменяется на (-17, 0) для PBV0. Предыдущее значение BV (-16, 0) из PBV1 отбрасывается. С другой 45 стороны, если BVnew составляет (-12, 0), то PBV0 и PBV1 не изменяются, поскольку BVnew соответствует PBV0. Если BVnew соответствует PBV1 (в этом примере оба составляют (-16, 0)), то выполняется изменение порядка PBV0 и PBV1 на противоположный: (-16, 0) заменяется на (-12, 0) для PBV1, а (12, 0) заменяется на (-16,

0) (для BVnew) для PBV0.

В описанных здесь примерных пинг-понговых подходах для единиц CU, имеющих четыре единицы PU, кодер и декодер используют разные схемы при обновлении пары пригодных значений BV, используемых для предсказания BV. Концептуально в другой
 5 схеме используется обновление пары пригодных значений BV на уровне CU в отличии от схемы обновления на уровне PU. По окончании предсказания BV для CU пара пригодных значений BV включает в себя значения BV из первых двух PU единицы CU (значение BV для PU1 в буфере 0 для BV, и значение BV для PU0 в буфере 1 для BV. В противоположность этому, согласно ранее рассмотренным пинг-понговым подходам
 10 по окончании предсказания BV для CU пара пригодных значений BV включает в себя значения BV из двух последних PU единицы CU, которые могут представлять собой PU3 и PU2.

В качестве альтернативы кодер и декодер могут использовать другую схему при обновлении упомянутой пары пригодных значений BV, используемых для предсказания
 15 BV. Если значения флага закодированы с использованием значений фиксированной длины, то кодер и декодер, например, могут «переключаться» с одного буфера на другой, попеременно обновляя два буфера BV с использованием значения BV, которое было только что создано или восстановлено. Таким путем кодер и декодер могут использовать единственную операцию для замены более старого пригодного значения
 20 BV (вместо перехода от более нового пригодного значения BV к месту более старого пригодного значения BV с добавлением нового значения BV к предыдущему более новому пригодному значению B), но кодер и декодер должны следить, какой буфер BV следует обновить далее. Также такое переключение может отрицательно сказаться на эффективности кодирования значений флага из-за помех, возникающих при выполнении
 25 контекстно ориентированного двоичного арифметического кодирования.

Когда отсутствуют действительные значения BV для использования в качестве пригодных значений BV (например, перед кодированием или декодированием первой CU в STU), инициализация пары пригодных значений BV может быть выполнена с использованием значений по умолчанию. Например, при инициализации обоим
 30 пригодным значениям BV присваиваются $(-2W, 0)$, где W - ширина CU, либо при инициализации одному из пригодных значений BV присваивают $(-2W, 0)$, а другому пригодному значению BV можно присвоить $(0, -2H)$, где H - высота CU (может быть такой же, как W). Либо при инициализации пригодного значения BV может быть присвоено другое значение, например, $(-W, 0)$ или $(0, -H)$. Либо в целях упрощения
 35 вместо использования значения по умолчанию, которое зависит от действительного размера CU, при инициализации пригодному значению BV может быть присвоено фиксированное значение, такое как $(-8, 0)$, $(-16, 0)$, $(0, -8)$ или $(0, -16)$, либо можно использовать значение, переданное в битовом потоке в качестве сигнализации. Это подходит, например, тогда, когда единицы CU размером 8 или 16 являются наиболее
 40 общими для предсказания BV. В любом случае после запуска предсказания BV действительные значения BV в паре пригодных значений BV заменяют значения BV по умолчанию.

Во всех последующих примерах, если не указано иное, в буфере 0 запоминается предпочтительное пригодное значение BV, которое скорее всего будет выбрано, а не
 45 другое пригодное значение BV, запомненное в буфере 1. Для повышения эффективности кодирования значений флага при контекстно ориентированном двоичном арифметическом кодировании можно использовать более высокую вероятность наиболее употребительного предиктора BV.

А. Пример предсказания BV для PU размером $2N \times 2N$ единицы CU размером $2N \times 2N$

На фиг. 13 показан пример предсказания BV для PU размером $2N \times 2N$ единицы CU размером $2N \times 2N$ согласно пинг-понговым подходам, приведенным в качестве примеров. Единица CU размером $2N \times 2N$ включает в себя одну единицу PU размером $2N \times 2N$ (PU0),

чей BV предсказывают с использованием пары пригодных значений BV.

На фиг. 13 в буфере 0 для BV изначально хранится первое начальное пригодное значение BV, BVinit_0 (PBV0), а в буфере 1 для BV изначально хранится второе начальное пригодное значение, BVinit_1 (PBV1). Значение BV для единственной PU размером $2N \times 2N$ (PU0) предсказывают с использованием либо PBV0, либо PBV1, как

указано значением флага для PU размером $2N \times 2N$. Значение BV для PU0 обрабатывают с использованием выбранного предиктора BV (например, во время кодирования или декодирования).

Затем упомянутая пара пригодных значений BV обновляется следующим образом.

PBV1 заменяют на PBV0, и PBV0 заменяют на значение BV для PU0. То есть, PBV1=

PBV0, и PBV0=BV_{PU0}. На фиг. 13 буфер 1 для BV обновляется для запоминания BVinit_1,

а буфер 0 для BV обновляется для запоминания значения BV для PU0 (BV_{PU0}). В качестве

альтернативы, как объяснялось выше, кодер или декодер избирательно обновляет пару

пригодных значений BV (и буферов BV) в зависимости от того, равно ли новое значение

BV, BV_{new}, (здесь это BV_{PU0}) значению PBV0.

В. Пример предсказания BV для PU размером $N \times N$ единицы CU размером $2N \times 2N$

На фиг. 14 показан пример предсказания BV для четырех PU размером $N \times N$ единицы

CU размером $2N \times 2N$ согласно пинг-понговым подходам, приведенным в качестве

примеров. Единица CU размером $2N \times 2N$ включает в себя четыре единицы PU размером

$N \times N$ (PU0, PU1, PU2 и PU3), чьи значения BV предсказывают с использованием пары

пригодных значений BV, которые обновляются во время процесса предсказания BV для CU.

На фиг. 14 в буфере 0 для BV изначально хранится первое начальное пригодное

значение BV, BVinit_0 (PBV0), а в буфере 1 для BV изначально хранится второе

начальное пригодное значение BVinit_1 (PBV1). Значение BV для первой PU размером

$N \times N$ (PU0) предсказывают с использованием либо PBV0, либо PBV1, как указано

значением флага для PU0. Значение BV для PU0 (BV_{PU0}) обрабатывают с использованием

выбранного предиктора BV (например, во время кодирования или декодирования).

Затем упомянутая пара пригодных значений BV обновляется путем замены PBV1

на PBV0, и путем замены PBV0 на значение BV для PU0. То есть, PBV1=PBV0, и PBV0=

BV_{PU0}. На фиг. 14 буфер 1 для BV обновляется для запоминания BVinit_0, а буфер 0

для BV обновляется для запоминания BV_{PU0}. В качестве альтернативы, как объяснялось

выше, кодер или декодер избирательно обновляет пару пригодных значений BV (и

буферов BV) в зависимости от того, равно ли новое значение BV, BV_{new} (здесь это

BV_{PU0}) значению PBV0.

Далее предсказывают значение BV для PU размером $N \times N$ (PU1) с использованием

либо PBV0 (на фиг. 14 это BV_{PU0} ее соседа слева), либо PBV1 (на фиг. 14 это BVinit_0),

как указано значением флага для PU1. Значение BV для PU1 (BV_{PU1}) обрабатывают с

использованием выбранного предиктора BV (например, во время кодирования или

декодирования).

Затем выполняется обновление пары пригодных значений BV путем замены PBV1

на PBV0 и замены PBV0 на значение BV для PU1. То есть, PBV1=PBV0, и PBV0=BV_{PU1}.

На фиг. 14 буфер BV 1 обновляется для запоминания BV_{PU0} , а буфер BV 0 обновляется для запоминания BV_{PU1} . В качестве альтернативы, как объяснялось выше, кодер или декодер избирательно обновляет пару пригодных значений BV (и буферов BV) в зависимости от того, равно ли новое значение BV, BV_{new} (здесь это BV_{PU1}) значению $PBV0$.

Далее предсказывают значение BV для третьей PU размером $N \times N$ (PU2) с использованием либо $PBV0$ (на фиг. 14 это BV_{PU1} ее соседа выше справа), либо $PBV1$ (на фиг. 14 это BV_{PU0} ее соседа сверху), как показано значением флага для PU2. Для этого значения флага в буфере 1 для BV запоминается предпочтительное пригодное значение BV, но не в буфере 0 для BV. Такая настройка учитывает BV_{PU0} (сосед сверху), являющуюся более вероятным кандидатом, чем BV_{PU1} (сосед сверху справа). Значение BV для PU2 (BV_{PU2}) обрабатывается с использованием выбранного предиктора BV (например, во время кодирования или декодирования). После этого пригодные значения BV в буфере 0 для BV (на фиг. 14 это BV_{PU1}) и буфере 1 для BV (на фиг. 14 это BV_{PU0}) не изменяются.

Наконец, значение BV для четвертой PU размером $N \times N$ предсказывают с использованием либо значения BV для PU2 (BV_{PT0}), которое буферизировано в другом месте, либо $PBV1$ (на фиг. 14 это BV_{PU1} соседа сверху), как указано значением флага для PU3. Для этого значения флага в буфере 0 для BV снова запоминается предпочтительное пригодное значение BV, хотя другое пригодное значение BV в буфере BV 1 не запоминается. Значение BV для PU3 (BV_{PU3}) обрабатывается с использованием выбранного предиктора BV (например, во время кодирования или декодирования). После этого пригодные значения BV в буфере 0 для BV и буфере 1 для BV не изменяются. На фиг. 14 по окончании предсказания BV для четырех PU размером $N \times N$ единицы CU, пара пригодных значений BV составляет BV_{PU1} (в буфере 0 для BV) и BV_{PU0} (в буфере 1 для BV).

В противоположность этому, согласно предыдущим пинг-понговым подходам для единиц PU размером $N \times N$ единицы CU размером $2N \times 2N$ при создании или восстановлении значения BV пригодное значение BV в буфере 0 BV перемещают в буфер 1 для BV, а только что созданное/восстановленное значение BV запоминают в буфере 0 для BV. По окончании предсказания BV для четырех PU размером $N \times N$ единицы CU размером $2N \times 2N$, пара пригодных значений BV будет представлять собой BV_{PU3} (в буфере 0 для BV) и BV_{PU2} (в буфере 1 для BV). На практике вышеперечисленные пинг-понговые подходы имеют ряд недостатков. Если следующая CU имеет единственную PU размером $2N \times 2N$, то BV_{PU0} текущей CU обычно будет представлять собой наилучший предиктор BV, но при этом BV_{PU0} текущей CU не сохраняется в качестве одного из пригодных значений BV. Либо, если следующая CU имеет две PU размером $N \times 2N$, то BV_{PU0} и BV_{PU1} текущей CU обычно представляют собой наилучшие предикторы BV для PU0 и PU1 следующей CU соответственно, но при этом BV_{PU0} и BV_{PU1} текущей CU не сохраняются в качестве пригодных значений BV.

Однако согласно описанным здесь пинг-понговым подходам после предсказания BV для единиц PU размером $N \times N$ единицы CU размером $2N \times 2N$ (как показано на фиг. 14), пара пригодных значений BV будет представлять собой BV_{PU1} (в буфере 0 для BV) и BV_{PU0} (в буфере 1 для BV). Это сохраняет пригодные значения BV, которые являются

скорее всего наиболее пригодными для предсказания BV для следующей CU.

С. Пример предсказания BV для PU размером $N \times 2N$ или PU размером $2N \times N$ единицы CU размером $2N \times 2N$

На фиг. 15 показан пример предсказания BV для двух PU размером $N \times 2N$ или двух PU размером $2N \times N$ единицы CU размером $2N \times 2N$ согласно пинг-понговым подходам, приведенным в качестве примеров. Единица CU размером $2N \times 2N$ включает в себя две единицы PU размером $N \times 2N$ (левую PU0 и правую PU1) или две PU размером $2N \times N$ (верхнюю PU0 и нижнюю PU1), чьи значения BV предсказывают с использованием пары пригодных значений BV, которые обновляются во время процесса предсказания BV для CU.

На фиг. 15 в буфере 0 для BV изначально хранится первое начальное пригодное значение BV BVinit_0 (PBV0), а в буфере 1 для BV изначально хранится второе начальное пригодное значение BVinit_1 (PBV1). Значение BV для первой PU (PU0) предсказывают с использованием либо PBV0, либо PBV1, как указано значением флага для PU0. Значение BV для PU0 (BV_{PU0}) обрабатывают с использованием выбранного предиктора BV (например, во время кодирования или декодирования).

Затем упомянутая пара пригодных значений BV обновляется путем замены PBV1 на PBV0, и путем замены PBV0 на значение BV для PU0. То есть, PBV1=PBV0, и PBV0=BV_{PU0}. На фиг. 15 буфер 1 для BV обновляется для запоминания BVinit_0, а буфер 0 для BV обновляется для запоминания BV_{PU0}. В качестве альтернативы, как объяснялось выше, кодер или декодер избирательно обновляет пару пригодных значений BV (и буферов BV) в зависимости от того, равно ли новое значение BV, BV_{new}, (здесь это BV_{PU0}) значению PBV0.

Далее предсказывают значение BV для второй PU (PU1) с использованием либо PBV0 (на фиг. 15 это BV_{PU0} ее соседа), либо PBV1 (на фиг. 15 это BVinit_0), как указано значением флага для PU1. Значение BV для PU1 (BV_{PU1}) обрабатывают с использованием выбранного предиктора BV (например, во время кодирования или декодирования).

Затем выполняется обновление пары пригодных значений BV путем замены PBV1 на PBV0 и замены PBV0 на значение BV для PU1. То есть, PBV1=PBV0, и PBV0=BV_{PU1}. На фиг. 15 буфер BV 1 обновляется для запоминания BV_{PU0}, а буфер BV 0 обновляется для запоминания BV_{PU1}. В качестве альтернативы, как объяснялось выше, кодер или декодер избирательно обновляет пару пригодных значений BV (и буферов BV) в зависимости от того, равно ли новое значение BV BV_{new} (здесь это BV_{PU1}) значению PBV0.

D. Примерные приемы предсказания BV согласно пинг-понговым подходам

На фиг. 16 показан пример приема для предсказания значений BV для единицы CU с множеством единиц PU согласно примерным пинг-понговым подходам. Единица CU представляет собой набор из одного или нескольких блоков в общем случае для целей кодирования и декодирования, а каждая из единиц PU представляет собой набор из одного или нескольких блоков в CU в целях передачи информации о предсказании в качестве сигнализации и/или в целях обработки предсказания. Этот прием (1600) может выполнять кодер изображений или кодер видео, такой как кодер, описанный со ссылками на фиг. 3 или на фигуры 5a-5b, либо другой кодер. Этот прием (1600) может выполнять декодер изображений или декодер видео, такой как декодер, описанный со ссылками на фиг. 4 или фиг. 6, либо другой декодер.

Для запуска пара пригодных значений BV состоит из первого начального пригодного

значения BV BVinit_0 (PBV0) и второго начального пригодного значения BVinit_1 (PBV1), которые хранятся в буферах 0 и 1 для BV соответственно.

Для каждой из множества единиц PU текущей CU текущей картинки кодер или декодер предсказывает (1610) значение BV для PU, используя одно из пары пригодных значений BV. Значение флага указывает на выбор в паре пригодных значений BV. Пара пригодных значений BV может отличаться для разных PU единицы CU. Например, когда текущая CU включает в себя четыре PU, для первой PU пара пригодных значений PBV0 и PBV1 может представлять собой BVinit_0 и BVinit_1 соответственно. Для второй PU пара пригодных значений BV PBV0 и PBV1 может представлять собой BV для первой PU (BV_{PU0}) и BVinit_0 соответственно. Для третьей PU пара пригодных значений BV PBV0 и PBV1 может представлять собой значение BV для второй PU (BV_{PU1}) и BV_{PU0} соответственно. Наконец, для четвертой PU пара пригодных значений BV может представлять собой BV_{PU1} и значение BV для третьей PU (BV_{PU2}).

Затем кодер или декодер обрабатывает (1620) значение BV для данной PU, используя предсказанное значение BV для данной PU. Например, во время кодирования кодер определяет значение приращения BV, используя значение BV для данной PU и предсказанное значение BV. Либо во время декодирования декодер восстанавливает значение BV для данной PU на основе по меньшей мере частично предсказанного значения BV.

Буферы BV могут обновляться автоматически или избирательно (в зависимости от того, равно ли новое значение BV, BV_{new}, значению PBV0). Например (как показано на фиг. 14), когда текущая CU включает в себя четыре PU, перед предсказанием BV для первой PU буфер 0 для BV запоминает BVinit_0, а буфер 1 запоминает BVinit_1. Перед предсказанием BV для второй PU (PU1) буфер 0 для BV запоминает BV_{PU0}, а буфер 1 для BV запоминает BVinit_1. Перед предсказанием BV для третьей PU (PU2) буфер 0 для BV запоминает BV_{PU1} а буфер 1 для BV запоминает BV_{PU0}. После обработки для PU2 в буфере 0 для BV сохраняется BV_{PU1}, а в буфере 1 для BV сохраняется BV_{PU0}.

После предсказания BV для текущей CU, после обновления буферов BV пара пригодных значений BV будет представлять собой BV_{PU1} текущей CU. Эти значения станут начальными пригодными значениями BV для предсказания BV для последующей CU текущей картинки. Таким образом, когда текущая CU включает в себя четыре PU, то конечная пара пригодных значений BV может включать в себя значения BV для первой и второй единиц PU текущей CU, несмотря на то, что значения BV для первой и второй PU текущей CU были обработаны до обработки значений BV для третьей и четвертой PU текущей CU.

На фиг. 17 в качестве примера показан другой прием для предсказания значений BV для единиц CU согласно примерным пинг-понговым подходам с использованием избирательного обновления буферов BV. Этот прием (1700) может выполнять кодер изображений или кодер видео, например, описанный со ссылками на фиг. 3, или фигуры 5a-5b, либо другой кодер. Либо этот прием (1700) может выполнять декодер изображений или декодер видео, например, описанный со ссылками на фиг. 4, или фиг. 6, либо другой декодер.

Кодер или декодер обрабатывает множество CU текущей картинки. Например, кодер обрабатывает множество единиц CU как часть кодирования и выводит кодированные данные как часть битового потока. Либо декодер обрабатывает множество единиц U как часть декодирования и выводит восстановленные значения отсчетов. Любая данная CU из упомянутого множества CU может включать в себя одну PU ли множество PU.

Для запуска обработки пара пригодных значений BV представляет собой первое начальное пригодное значение BV, BVinit_0 (PBV0) и второе начальное пригодное значение BV, BVinit_1 (PBV1), которые хранятся в буферах 0 и 1 для BV соответственно.

Для данной CU из множества CU кодер или декодер проверяет (1710), имеет ли данная CU одну или множество единиц PU. Если данная CU имеет одну PU, то кодер или декодер предсказывает (1720) значение BV для этой одной PU, используя одно из пары пригодных значений BV: PBV0 или PBV1 (на фиг. 17 это BVinit_0 и BVinit_1), где значение флага указывает на выбранное значение из пары пригодных значений BV. Кодер или декодер обрабатывает (1730) значение BV для этой одной PU, используя предсказанное значение BV. Кодер или декодер избирательно обновляет буферы 0 и 1 для BV в зависимости от того, равно ли значение BV для данной PU (BV_{PU0}) значению BV, запомненному в буфере 0 для BV (BVinit_0). Например, кодер или декодер сравнивает BV_{PU0} со значением BV, запомненным в буфере 0 (BVinit_0). Если BV_{PU0} отличается от значения BV, запомненного в буфере 0 (BVinit_0), то кодер или декодер запоминает значение BV из буфера 0 (BVinit_0) в буфере 1 и запоминает BV_{PU0} в буфере 0 для BV.

После обработки (1730) при обновлении буферов BV в буфере 0 хранится значение BV для упомянутой одной PU (BV_{PU0}), а в буфере 1 хранится BVinit_0. Кодер или декодер проверяет (1740), продолжать ли обработку следующей CU данной картинки. Если да, то пара буферизированных значений BV рассматривается в качестве начальных пригодных значений BV (BVinit_0 и BVinit_1) для следующей CU.

В противном случае, когда данная CU содержит множество PU, кодер или декодер выполняет предсказание BV и обработку для единиц PU данной CU. Для каждой из множества единиц PU данной CU кодер или декодер предсказывает (1750) значение BV для данной PU, используя одно из пары пригодных значений BV (PBV0 или PBV1) (изначально это BVinit_0 и BVinit_1, которые позднее обновляются). Значение флага для данной PU указывает на выбор, сделанный из пары пригодных значений BV. Кодер или декодер обрабатывает (1760) значение BV для данной PU, используя предсказанное значение BV. Если данное PU является первой или второй PU в заданной CU, то кодер или декодер избирательно обновляет буферы 0 и 1 для BV в зависимости от того, равно ли значение BV для данной PU (BV_{PUx}) значению BV, запомненному в буфере 0 (PBV0). Например, кодер или декодер сравнивает BV_{PUx} с PBV0. Если BV_{PUx} отличается от PBV0, то кодер или декодер запоминает PBV0 в буфере 1 и запоминает BV_{PUx} в буфере 0 для BV.

Кодер или декодер проверяет (1770), является ли данная PU последней PU в данной CU. Если нет, то кодер или декодер продолжает обработку, выполняя предсказание BV (1750) для следующей PU данной CU. Например, единицы PU содержат четыре блока размером $N \times N$ два блока размером $N \times 2N$ или два блока размером $2N \times N$ единицы CU размером $2N \times 2N$.

После обработки (1760) для последней PU данной CU после обновления обоих буферов BV в буфере 0 хранится значение BV для второй PU (BV_{PU1}) данной CU, а в буфере 1 хранится значение BV для первой PU (BV_{PU0}) данной CU. Кодер или декодер проверяет (1740), продолжать ли обработку следующей CU данной картинки. Если да, то пара буферизированных значений BV (BV_{PU1} и BV_{PU0}) будет рассматриваться в качестве начальных пригодных значений BV (BVinit_0 и BVinit_1) для следующей CU.

IX. Опции на стороне кодера для режима внутреннего предсказания с блочным копированием

В этом разделе представлены различные нововведения, касающиеся опций на стороне кодера, для внутреннего предсказания с блочным копированием («BC»). Некоторые из этих нововведений относятся к параллельному выполнению оценки блочного вектора («BV») и принятию решений о разделении того или иного блока. Другие нововведения относятся к избирательному слиянию блоков в один более крупный блок во время оценки BV. Ряд других нововведений относится к оценке значений отсчетов в области перекрытия текущего блока во время оценки BV. В общем случае эти нововведения повышают эффективность кодирования внутреннего предсказания с BC.

Как часть оценки BV, кодер может использовать любой из нескольких подходов.

Кодер может использовать полный поиск, оценивая каждое пригодное значение BV, допустимое в области поиска. Либо кодер может использовать частичный поиск, оценивая только некоторые из пригодных значений BV, допустимых в области поиска. Например, кодер может начать частичный поиск с предсказанным значением BV для текущего блока (например, предсказанном на основе значений BV для одного или нескольких соседних блоков). После оценки пригодного значения BV в начальной позиции для частичного поиска кодер может оценить одно или несколько других пригодных значений BV, увеличивая расстояние от стартовой позиции (например, в соответствии со схемой спирального поиска или какой-либо иной схемой). При оценке данного пригодного значения BV кодер может сравнить все значения отсчетов в области внутреннего предсказания с текущим блоком. Либо кодер может выполнить оценку поднабора значений отсчетов (то есть, подотсчеты, значения которых оцениваются). При сравнении значений отсчетов в области внутреннего предсказания и текущего блока для определения затрат на искажения кодер может вычислить среднеквадратическую ошибку, сумму квадратов отклонений, сумму абсолютных отклонений или некоторый иной показатель искажений. Кодер также может определить затраты на битрейт, связанные с кодированием пригодного значения BV.

Во время кодирования на различных этапах кодер может выполнить оптимизацию соотношения «битрейт-искажения» («RDO»), при которой оцениваются затраты на битрейт и затраты на искажения для различных опций. В общем случае затраты на «битрейт-искажения» для той или иной опции определяются как $D + \lambda R$ (или $R + \lambda D$), где R представляет затраты на битрейт в битах кодированных данных, D представляет затраты на искажения при использовании такого показателя, как среднеквадратическая ошибка, или показателя перцептуального искажения, а λ - множитель Лагранжа (пример параметра взвешивания), который взвешивает затраты R на битрейт по сравнению с затратами D на искажения. Как правило, кодер выбирает опцию, которая обеспечивает минимальные затраты на соотношение «битрейт-искажения».

А. Избирательное слияние блоков в более крупный блок во время оценки BV

Во время оценки BV кодер может объединить блоки в более крупный блок. Таким образом кодер может сократить количество бит, используемых для передачи сигнальной информации о BV. В частности, когда внутреннее предсказание с BC выполняется на блочной (ТВ) основе, кодер может выполнить оценку BV на блочной (СВ) основе с последующим слиянием малых СВ в более крупный СВ, основываясь на разделении блока СВ на более мелкие блоки ТВ во время внутреннего предсказания с BC, чтобы избежать перекрытия блока ТВ с его областью внутреннего предсказания.

На фиг. 18 показан прием (1800) для избирательного слияния блоков в более крупный блок во время оценки BV. Прием (1800) может выполняться кодером, например, описанным со ссылками на фиг. 3 или фигуры 5a и 5b, либо другим кодером.

Согласно данному приему, когда кодер кодирует данные для картинки, используя

внутреннее предсказание с ВС для каждого из множества блоков, имеющих первый размер, кодер определяет (1810) значение BV, используя оценку BV. Первым размером может быть 8×8, 16×16, 32×32 или какой-либо другой размер. В общем случае для каждого из блоков, имеющих первый размер, оценка BV включает в себя определение затрат на пригодное значение BV. Эти затраты могут включать в себя затраты на битрейт и/или затраты на искажения. Идентифицированное значение BV относится к области внутреннего предсказания, которая не перекрывает упомянутый блок, имеющий первый размер.

Кодер избирательно объединяет (1820) два или более блоков в блок, имеющий второй размер, который больше первого размера. Вторым размером может быть 16×16, 32×32, 64×64 или какой-либо иной размер. Например, кодер сравнивает значения BV двух или более соседних блоков с множеством блоков, имеющих первый размер, и если сравниваемые значения BV идентичны, то объединяет эти два или более соседних блоков в блок, имеющий второй размер. Блоку, имеющему второй размер, присваивается идентичное значение BV. В отличие от блоков, имеющих первый размер, указанное значение BV может относиться к области внутреннего предсказания, которая перекрывает блок, имеющий второй размер.

Кодер может повторить прием (1800) для другого блока. Наконец, кодер выводит кодированные данные для текущей картинки.

В некоторых примерных реализациях каждый из множества блоков, имеющих первый размер, является блоком СВ, а блоком, имеющим второй размер, также является блок СВ. Однако, внутреннее предсказание с ВС выполняется на блочной (ТВ) основе. При оценке BV может использоваться полный поиск в области поиска для каждого из множества блоков, имеющих первый размер. Либо при оценке BV может использоваться частичный поиск в области поиска для каждого из множества блоков, имеющих первый размер, например, начиная с предсказанного значения BV для данного блока, с возможным завершением частичного поиска по меньшей мере частично на основе затрат на пригодное значение BV, которое было оценено.

На фиг. 19 показано преимуществ избирательного слияния (1900) блоков в более крупный блок во время оценки BV. На фиг. 19 четыре блока под номерами 0, 1, 2 и 3 (показаны сплошными линиями) имеют значения BV, которые относятся к соответствующим областям внутреннего предсказания 0', 1', 2' и 3' соответственно. То есть, значение BV относится к области 0' внутреннего предсказания для блока 0, относится к области 1' внутреннего предсказания для блока 1 и т.д. Эти области внутреннего предсказания показаны пунктирными линиями. Ни один из блоков 0, 1, 2 или 3 не перекрывает соответствующую область внутреннего предсказания. Однако более крупный блок, включающий в себя блоки 0, 1, 2 и 3 (показаны жирными линиями) перекрывает соответствующую область внутреннего предсказания. Если бы кодер выполнял оценку BV для более крупного блока, когда перекрытие области внутреннего предсказания и блока запрещено, кодер не смог бы идентифицировать значение BV, показанное на фиг. 19 в качестве разрешенного значения BV. В противном случае, когда кодер выполняет оценку BV для более мелких блоков 0, 1, 2 и 3, он оценивает одинаковое значение BV для всех четырех блоков и может объединить эти четыре блока в более крупный блок, имеющий то же значение BV. Во время кодирования или декодирования этот более крупный блок будет разделен на более мелкие блоки для выполнения реальных операций внутреннего предсказания с ВС. Например, блок СВ данного размера разделяют на более мелкие блоки ТВ для выполнения операций внутреннего предсказания с ВС на блочной (ТВ) основе без

перекрытия ТВ и его области внутреннего предсказания.

В. Параллельная оценка блочного вектора и принятие решений о разделении блока

Кодер может параллельно выполнять оценку BV и принимать решение о разделении для текущего блока. В частности, при выполнении операций внутреннего предсказания с ВС на блочной (ТВ) основе кодер может параллельно оценивать пригодное значение BV и возможно принимать решения о разделении для текущего блока.

На фиг. 20 представлена блок-схема, иллюстрирующая обобщенный прием для параллельного выполнения оценки BV и принятия решений о разделении для того или иного блока. Этот прием (2000) может выполняться кодером, например, кодером, описанным со ссылками на фиг. 3 или фигуры 5a и 5b, либо другим кодером.

Когда кодер выполняет кодирование данных для картинки, используя внутреннее предсказание с ВС, он кодирует (2110) данные для текущего блока. В качестве части кодирования (2010) кодер выполняет параллельно операции оценки BV и операции по принятию решения о разделении блока. В некоторых примерных реализациях текущий блок представляет собой блок СВ, соответствующий по размеру блоку ТВ, и внутреннее предсказание с ВС выполняется на блочной (ТВ)основе. Например, как часть кодирования (2010) кодер выполняет прием (2100), показанный на фиг. 21, для параллельной оценки пригодного значения BV и принятия решений о возможном разделении для текущего блока.

На фиг. 21 представлена блок-схема, иллюстрирующая примерный прием для параллельной оценки пригодного значения BV и принятия решения о разделении для того ли иного блока. Способ (2100) может выполняться кодером, таким как кодер, описанный со ссылками на фиг. 3 или фигуры 5a и 5b, либо другим кодером.

Кодер идентифицирует (2110) пригодное значение BV для текущего блока, используя операции оценки BV. Текущий блок имеет первый размер. Например, этот первый размер составляет 8×8, 16×16, 32×32, 64×64 или какой-либо другой размер.

Кодер проверяет (2120), следует ли разделить текущий блок. Например, кодер оценивает, приводит ли внутреннее предсказание с ВС при использовании данного пригодного значения BV для текущего блока к перекрытию этого текущего блока и области внутреннего предсказания, относящейся к данному пригодному значению BV.

В зависимости от результата проверки (2120) кодер избирательно разделяет (2130) текущий блок на множество блоков, имеющих каждый второй размер, который меньше первого размера. Например, вторым размером является 4×4, 8×8, 16×16, 32×32, или некоторый другой размер. Затем (при разделении текущего блока) кодер повторяет (2140) прием (2100) для каждого из блоков меньшего размера. То есть, для каждого из блоков, имеющих второй размер, кодер повторяет операции идентификации, оценки, избирательного разделения и т.д. для блока, имеющего второй размер и рассматриваемого в качестве текущего блока.

С другой стороны, если текущий блок не разделен, то кодер выполняет кодирование (2150) текущего блока и измеряет (2160) затраты на его кодирование. Например, при кодировании текущего блока кодер: (a) предсказывает значения отсчетов текущего блока, используя внутреннее предсказание с ВС; (b) определяет значения остатка для текущего блока, используя предсказанные значения отсчетов и значения отсчетов текущего блока; (c) в качестве опции, применяет частотное преобразование к значениям остатка для создания коэффициентов преобразования и выполняет квантование этих коэффициентов преобразования; (d) в качестве опции выполняет обратное квантование упомянутых коэффициентов преобразования и применяет обратное частотное преобразование для восстановления значений остатка; и (e) объединяет значения остатка

с предсказанными значениями отсчетов. Измеренные затраты могут представлять собой затраты на соотношение «битрейт-искажения», затраты на битрейт или затраты на искажения. Затраты на битрейт могут учитывать как затраты на сигнализацию о значении BV, так и затраты на передачу сигнальной информации о принятии решения по разделению блока.

В качестве альтернативы кодер оценивает пригодное значение BV и принимает решения о возможном разделении для текущего блока каким-либо иным образом. Например, вдобавок к рассмотрению перекрытия области внутреннего предсказания и блока кодер также учитывает, имеет ли блок на данный момент минимально допустимый размер. Если да, то этот блок не разделяют, но данное пригодное значение BV запрещается.

Обратимся к фиг. 20, где в качестве части операции кодирования (2010) кодер может записать затраты для пригодного значения BV. Затем кодер может оценить одно или несколько других пригодных значений BV. Например, кодер повторяет прием (2100) для другого пригодного значения BV. В конце концов, кодер выбирает один или несколько векторов BV и принимает решение о разделении для текущего блока (и блоков в составе данного текущего блока), что обеспечивает минимальные затраты (например, затраты на соотношение «битрейт-искажения», затраты на битрейт или затраты на искажения).

Кодер выводит (2020) кодированные данные для упомянутого блока. Кодер может повторить прием (2000) для другого блока.

С. Оценка восстановленных значений отсчетов в области перекрытия

В некоторых описанных здесь примерах сигнализация о векторах BV передается на уровне единиц CU и применяется для блоков на уровне CB. Перекрытие блока CB с соответствующей областью внутреннего предсказания не разрешается. В других описанных здесь примерах кодер использует значение BV для блоков меньшего размера на блочной (TB) основе, даже если значение BV передается в качестве сигнализации на более высоком синтаксическом уровне (например, для CU). Например, значение BV передается для CU с CB размером 32×32 , но это значение BV применяется к блокам на уровне блоков TB размером 4×4 или блокам TB размером 8×8 . Это позволяет использовать BV с меньшим значением при отсутствии перекрытия между блоком на уровне TB и областью внутреннего предсказания, которая копируется.

Когда данный блок разделяют на более мелкие блоки для выполнения операций внутреннего предсказания с BC, этот блок включает в себя область перекрытия, в которой значения отсчетов данного блока могут составлять часть областей внутреннего предсказания для одного или нескольких блоков меньшего размера. В этом разделе описываются подходы к оценке восстановленных значений отсчетов внутри области перекрытия. В частности, в этом разделе описаны подходы к оценке восстановленных значений отсчетов в области перекрытия блока, когда значение BV для этого блока применяется к блокам меньшего размера на блочной (TB) основе.

На фиг. 22а показан текущий блок (2210), имеющий размер $2m \times 2n$. Текущий блок (2210) включает в себя четыре блока (2221, 2222, 2223, 2224) размером $m \times n$ для операций внутреннего предсказания с BC. Например, текущий блок (2210) представляет собой блок единицы CU размером 8×8 , а четыре блока (2221, 2222, 2223, 2224) меньшего размера представляют собой блоки размером 4×4 с блоками TB размером 4×4 . Либо текущий блок (2210) представляет собой блок единицы CU размером 16×16 , а четыре блока (2221, 2222, 2223, 2224) меньшего размера представляют собой блоки размером 8×8 с блоками TB размером 8×8 . При использовании операций предсказания для блоков

размером $m \times n$ в порядке зигзагообразного сканирования (на фиг. 22а блок 0, блок 1, блок 2, затем блок 3) область внутреннего предсказания для первого блока (2221) размером $m \times n$ находится вне текущего блока (2210) размером $2m \times 2n$. Однако для второго блока (2222) размером $m \times n$ область внутреннего предсказания может перекрывать первый блок (2221) размером $m \times n$ частично или полностью. Для третьего блока (2223) размером $m \times n$ область внутреннего предсказания может перекрывать первый блок (2221) размером $m \times n$ или второй блок (2222) размером $m \times n$ частично или полностью. И для четвертого блока (2224) размером $m \times n$ область внутреннего предсказания может перекрывать любой из первых трех блоков (2221, 2222, 2223) размером $m \times n$ текущего блока (2210) частично или полностью.

На фиг. 22b показана область (2230) перекрытия текущего блока 2210. Область (2230) перекрытия покрывает первые три блока (2221, 2222, 2223) размером $m \times n$ в порядке зигзагообразного сканирования, поскольку значения отсчетов в этих трех блоках размером $m \times n$ могут составлять часть области внутреннего предсказания по меньшей мере для одного блока размером $m \times n$ текущего блока (2210). При использовании другого размера для меньшего блока (например, блоки размером 4×4 блока размером 16×16 или блока размером 32×32) область перекрытия может покрывать еще большую часть текущего блока.

Во время оценки BV, когда кодер оценивает пригодное значение BV для текущего блока, разделенного на меньшие блоки для выполнения операций внутреннего предсказания с ВС, кодер может вычислить действительные восстановленные значения отсчетов в области перекрытия, когда кодер применяет пригодное значение BV к соответствующим блокам меньшего размера в порядке зигзагообразного сканирования. Для этого может потребоваться выполнение интенсивных вычислений, поскольку операции кодирования и операции восстановления применяются для каждого пригодного значения BV для блока меньшего размера. Вместо этого, кодер может оценить восстановленные значения отсчетов в области перекрытия. В частности, когда операции внутреннего предсказания с ВС выполняются для блоков меньшего размера на блочной (ТВ) основе, кодер может оценить восстановленные значения отсчетов в области перекрытия текущего блока, которая включает в себя множество более мелких блоков.

На фиг. 23 представлена блок-схема, иллюстрирующая примерный прием для оценки BV, где кодер оценивает восстановленные значения отсчетов в области перекрытия. Этот прием (2300) может выполняться кодером, таким как кодер, описанный со ссылками на фиг. 3 или фигуры 5a и 5b, либо другим кодером.

Для текущего блока картинки кодер оценивает (2310) восстановленные значения отсчетов в области перекрытия текущего блока. Текущий блок имеет множество блоков меньшего размера. Например, текущий блок имеет первый размер (например, 32×32 , 16×16 или 8×8), а каждый из множества блоков имеет второй размер, меньший первого размера (например, 16×16 , 8×8 или 4×4). Текущим блоком может быть блок единицы CU, и тогда каждый из множества блоков меньшего размера может находиться на уровне блока ТВ единицы TU. Область перекрытия покрывает части текущего блока, которые находятся в возможных областях внутреннего предсказания по меньшей мере для одного из множества блоков меньшего размера.

Согласно первому подходу к оценке восстановленных значений отсчетов в области перекрытия кодер использует соответствующие исходные значения отсчетов в качестве восстановленных значений отсчетов соответственно в области перекрытия. Вне текущего блока при оценке BV могут использоваться исходные значения отсчетов в картинке. В качестве альтернативы вне текущего блока при оценке BV могут использоваться

восстановленные значения отсчетов в картинке (то есть, восстановленные значения отсчетов вне текущего блока; исходные значения отсчетов в области перекрытия текущего блока). В этом случае исходные значения отсчетов могут использоваться в области внутреннего предсказания, когда ни одно из действительных восстановленных значений отсчетов недоступно во время оценки BV, а в противном случае используются действительные восстановленные значения отсчетов.

Согласно второму подходу к оценке восстановленных значений отсчетов в области перекрытия кодер обрабатывает исходные значения отсчетов в области перекрытия. В частности, кодер применяет частотное преобразование, квантование, обратное квантование и обратное частотное преобразование к исходным значениям отсчетов в области перекрытия, выполняя аппроксимацию восстановленных значений отсчетов в области перекрытия. При квантовании и обратном квантовании возможно использование параметра квантования, который был применен (или потенциально должен быть применен) в соседних блоках текущего блока.

Согласно третьему подходу к оценке восстановленных значений отсчетов в области перекрытия кодер использует операции внутреннего предсказания с ВС при оценке восстановленных значений отсчетов в области перекрытия. Кодер предсказывает значения отсчетов в области перекрытия, используя по меньшей мере один предиктор BV (например, значение BV для предшествующего блока в картинке, например, соседнего блока, или предиктор BV по умолчанию). Кодер определяет значения остатков на основе разностей между предсказанными значениями отсчетов и соответствующими исходными значениями отсчетов в области перекрытия. К значениям остатков кодер применяет частотное преобразование, квантование, обратное квантование и обратное частотное преобразование. При квантовании и обратном квантовании может использоваться параметр квантования, примененный (или потенциально подлежащий применению) в соседних блоках текущего блока. Кодер объединяет результаты обработки (то есть, восстановленные значения отсчетов) с предсказанными значениями отсчетов, что порождает аппроксимацию восстановленных значений отсчетов в области перекрытия.

В одной версии третьего подхода кодер предсказывает значения отсчетов в области перекрытия, используя по меньшей мере один предиктор BV (например, значение BV для предшествующего блока в картинке, например, соседнего блока, либо предиктор BV по умолчанию), но пропускает обработку значений остатков. Оцененные восстановленные значения отсчетов в области перекрытия представляют собой предсказанные значения отсчетов.

В другой версии третьего подхода кодер может оценить значимость значений остатков для области перекрытия (например, наличие ненулевых значений коэффициентов преобразования после преобразования, квантования и т.д.). Например, кодер может оценить, имеют ли значения остатков малую величину (по отдельности или вместе) либо проверить некоторое другое условие для значений остатков. Если значения остатков имеют малую значимость, кодер может пропустить обработку остатков и просто использовать предсказанные значения отсчетов в качестве оцененных восстановленных значений отсчетов в области перекрытия. В противном случае (значения остатков имеют большую значимость), кодер может выполнить кодирование, затем восстановить значение остатков, и затем объединить их с предсказанными значениями остатков для создания оцененных восстановленных значений остатков в области перекрытия, как было описано выше.

Кодер выполняет (2320) оценку BV для определения значения BV для текущего блока.

При оценке BV используют по меньшей мере некоторые из оцененных восстановленных значений отсчетов в области перекрытия текущего блока. При оценке BV также используют восстановленные значения отсчетов (или исходные значения отсчетов в одном из подходов) в картинке вне текущего блока.

5 Оценка BV может включать в себя для каждого из одного или нескольких пригодных значений BV для текущего блока, выполнение операций согласования блоков для более мелких блоков соответственно, и пригодных областей внутреннего предсказания согласно пригодному значению BV. Например, для данного пригодного значения BV для текущего блока, имеющего четыре более мелких блока, кодер выполняет операции
10 согласования блоков для четырех более мелких блоков и их соответствующих пригодных областей внутреннего предсказания в соответствии с данным пригодным значением BV.

В качестве альтернативы оценка BV может включать в себя: (1) для каждого из множества более мелких блоков определение значений BV для более мелкого блока, а
15 затем (2) синтез значений BV для множества более мелких блоков с образованием значения BV для данного текущего блока. Например, для текущего блока, имеющего четыре более мелких блока, кодер последовательно определяет значение BV для первого более мелкого блока, значение BV для второго более мелкого блока, значение BV для третьего более мелкого блока и значение BV для четвертого более мелкого блока.
20 После первого более мелкого блока оценка BV может включать в себя оценку областей внутреннего предсказания в области перекрытия текущего блока. Затем кодер синтезирует значения BV для четырех более мелких блоков, образуя значение BV для данного текущего блока. Например, кодер может: (а) усреднить значения BV для множества более мелких блоков; (b) вычислить медиану значений BV для множества
25 более мелких блоков; (с) идентифицировать значение BV для текущего блока в окрестности значения BV для множества более мелких блоков; (d) идентифицировать значение BV для текущего блока, используя RDO для пригодных значений BV в окрестности значений BV для множества более мелких блоков, или (е) синтезировать значения BV для множества более мелких блоков с образованием значения BV для
30 текущего блока каким-либо другим путем.

После идентификации значения BV для текущего блока посредством оценки BV, кодер может выполнить внутреннее предсказание с BC для текущего блока, используя значение BV для данного текущего блока. Действуя таким образом, кодер выполняет операции BC на блочной основе для множества более мелких блоков. Таким образом
35 кодер может выполнять оценку BV на уровне текущего блока, но выполнять компенсацию BV на уровне более мелких блоков. Например, кодер может выполнить оценку BV для блока размером 8×8 единицы CU размером 8×8, но операции внутреннего предсказания с BC (и операции восстановления) может выполнить на блочной (TB) основе для четырех блоков размером 4×4 в порядке зигзагообразного сканирования.
40 Блоки размером 4×4 используют одно и то же значение BV (CU размером 8×8), и кодер выполняет преобразования с размерностью 4×4, квантование, обратное квантование, обратные преобразования с размерностью 4×4 и т.д. для последующих блоков размером 4×4 значений остатков. Сначала восстанавливается первый блок размером 4×4 (для TB 0), который затем может быть использован в области внутреннего предсказания для
45 второго блока размером 4×4 (для TB 1). Потом восстанавливается второй блок размером 4×4, который затем может быть использован в области внутреннего предсказания для третьего блока размером 4×4 (для TB 2). Для четвертого блока размером 4×4 (для TB 3) может быть использован любой из первых трех блоков размером 4×4 в области

внутреннего предсказания.

На стороне декодера, как было описано выше, декодер может проверить, приводит ли значение BV к перекрытию области внутреннего предсказания с текущим блоком. Если да, то данный блок (на уровне ТВ) разделяют на четыре более мелких блока (на уровне более мелкого ТВ), затем четыре более мелких блока последовательно декодируют (с использованием операций внутреннего предсказания с ВС и восстановления) в зигзагообразном порядке. Таким образом восстановленные значения отсчетов для одного из более мелких блоков доступны для использования в области внутреннего предсказания для следующего более мелкого блока в зигзагообразном порядке. Если флаг блока преобразования единицы TU равен нулю, то внутреннее предсказание с ВС применяется только на уровне CU, и запрещается декорреляция цветности.

Д. Альтернативы и версии

Многие из предшествующих примеров обращены к значениям BV и операциям внутреннего предсказания с ВС для блоков яркости. Для видео в формате YUV 4:4:4 компоненты цветности имеют такое же разрешение, как компонента яркости. Значение BV, используемое для текущего блока яркости может быть использованы для соответствующих блоков цветности. При разделении текущего блока яркости на более мелкие блоки яркости для операций внутреннего предсказания с ВС соответствующие блоки цветности также разделяют на более мелкие блоки цветности таким же образом, как для операций внутреннего предсказания с ВС.

Для видео в формате YUV 4:2:0 компоненты цветности подвергаются понижающей дискретизации с коэффициентом два по горизонтали и коэффициентом два по вертикали. Например, если текущий блок яркости имеет размер 16×16, то соответствующие блоки цветности имеют размер 8×8. Значение BV, используемое для текущего блока яркости, может быть использовано для соответствующих блоков цветности после соответствующих операций масштабирования и округления (и/или отбрасывания). В общем случае, когда текущий блок яркости разделяют на более мелкие блоки яркости для операций внутреннего предсказания с ВС, соответствующие блоки цветности также разделяют на более мелкие блоки цветности аналогичным образом для операций внутреннего предсказания с ВС.

Однако в некоторых реализациях это не всегда имеет место. В некоторых реализациях, если блок ТВ яркости видео в формате YUV 4:2:0 имеет минимальную размерность преобразования, блоки ТВ яркости, имеющие эту минимальную размерность преобразования, используются для эквивалентов цветности блока ТВ яркости и его трех соседних блоков ТВ яркости для того же СВ. Например, если текущий ТВ яркости имеет минимальный размер 4×4 (и связан с СВ яркости размером 8×8), то соответствующие блоки СВ цветности имеют размер 4×4. Каждый СВ цветности размером 4×4 имеет единственный ТВ цветности размером 4×4. Блоки ТВ цветности размером 2×2, как можно ожидать при использовании понижающей дискретизации для формата YUV 4:2:0, не поддерживаются. В указанных реализациях разделение текущего блока яркости на более мелкие блоки яркости для операций внутреннего предсказания с ВС не поддерживается, если более мелкие блоки яркости будут иметь минимальную размерность преобразования, поскольку при применении к соответствующим блокам цветности разделение блока приводит к получению блоков цветности с размерностью преобразования, меньшим минимального. В указанных реализациях разделение текущего блока яркости (и соответствующих блоков цветности) на более мелкие блоки для операций внутреннего предсказания с ВС все же разрешается для более крупных

размеров.

Для видео в формате YUV 4:2:2 компоненты цветности подвергаются понижающей дискретизации с коэффициентом два по горизонтали. Например, если текущий блок яркости имеет размер 16×16, то соответствующие блоки цветности имеют размер 8×16.

5 Значение BV, используемое для текущего блока яркости, может быть использовано для соответствующих блоков цветности после соответствующих операций масштабирования и округления (и/или отбрасывания). В общем случае, когда текущий блок яркости разделяют на более мелкие блоки яркости для операций внутреннего предсказания с ВС, соответствующие блоки цветности также разделяют на более мелкие

10 блоки цветности аналогичным образом для операций внутреннего предсказания с ВС.

Однако в некоторых реализациях это не всегда имеет место. В некоторых реализациях, если блок ТВ яркости видео в формате YUV 4:2:2 имеет минимальную размерность преобразования, блоки ТВ яркости, имеющие эту минимальную размерность преобразования, используются для эквивалентов цветности блока ТВ яркости и одного

15 соседнего блока ТВ яркости в том же СВ. Например, если текущий ТВ яркости имеет минимальный размер 4×4 (и связан с СВ яркости размером 8×8), то соответствующие блоки СВ цветности размером 4×8 имеют каждый два ТВ цветности размером 4×4.

Блоки ТВ цветности размером 2×4, как можно ожидать при использовании понижающей дискретизации для формата YUV 4:2:2, не поддерживаются. Вместо этого каждый из

20 блоков ТВ цветности размером 4×4 в блоке СВ цветности размером 4×8 включает в себя значения остатков для двух областей цветности размером 2×4. В указанных реализациях разделение текущего блока яркости на более мелкие блоки яркости для операций внутреннего предсказания с ВС как было описано выше (при использовании соответствующего разделения блоков цветности на более мелкие блоки цветности для

25 операций внутреннего предсказания с ВС, как было описано выше) разрешается для размеров блоков, превышающих минимальную размерность преобразования. Если более мелкие блоки имеют минимальную размерность преобразования, разделение текущего блока яркости на более мелкие блоки яркости для операций внутреннего предсказания с ВС поддерживается, если блоки цветности значений остатков

30 перекомпонованы так, как показано на фиг. 24.

На фиг. 24 показаны блоки с минимальной размерностью преобразования 4×4 для видео в формате YUV 4:2:2. Блок (2410) яркости размером 8×8 включает в себя четыре блока (2421, 2422, 2423, 2424) яркости размером 4×4, которые пронумерованы как 0... 3. Соответствующий блок (2430) цветности размером 4×8 включает в себя

35 соответствующие компоненты цветности, которые подвергаются понижающей дискретизации с коэффициентом два по горизонтали. Размер 2×4 преобразования не поддерживается.

Согласно текущей черновой версии стандарта HEVC, для блока цветности размером 4×8 блоки 0 и 1 размером 2×4 обрабатываются вместе в качестве ТВ цветности размера

40 4×4 с использованием размера преобразования 4×4 для кодирования и декодирования остатков, а блоки 2 и 3 размером 2×4 обрабатываются вместе в качестве ТВ цветности размером 4×4 с размерностью преобразования 4×4 для кодирования и декодирования остатков. В результате, поскольку блок 0 размером 2×4 не будет восстановлен, пока не восстановлен блок 1 размером 2×4, операции внутреннего предсказания с ВС для

45 блока 1 размером 2×4 не обращаются к восстановленным значениям отсчетов в блоке 0 размером 2×4. Аналогичным образом, операции внутреннего предсказания с ВС для блока 3 размером 2×4 не обращаются к восстановленным значениям отсчетов в блоке 2 размером 2×4.

Как показано на фиг. 24, блоки размером 2×4 блока (2440) цветности размером 4×8 могут быть перекомпонованы с целью кодирования и декодирования остатков. В этом случае блоки 0 и 2 размером 2×4 обрабатываются вместе в качестве блока ТВ цветности размером 4×4 с размерностью преобразования 4×4 для кодирования и декодирования остатков, а блоки 1 и 3 размером 2×4 обрабатываются вместе в качестве ТВ цветности размером 4×4 с размерностью преобразования 4×4 для кодирования и декодирования остатков. Операции внутреннего предсказания с ВС тем не менее выполняются и в «неперекомпонованных» блоках размером 2×4 блока 2430 цветности размером 4×8 . При декодировании блока цветности размером 4×8 сначала может быть выполнено внутреннее предсказание с ВС для блоков 0 и 2 размером 2×4 , после чего следует декодирование остатков для блоков 0 и 2 размером 2×4 и восстановление значений отсчетов для блоков 0 и 2 размером 2×4 . Затем восстанавливаются блоки 1 и 3 размером 2×4 . Таким образом, блок 0 размером 2×4 может быть восстановлен до восстановления блока 1 размером 2×4 , так что операции внутреннего предсказания с ВС для блока 1 размером 2×4 могут обращаться к восстановленным значениям отсчетов в блоке 0 размером 2×4 . Аналогичным образом, операции внутреннего предсказания с ВС для блока 3 размером 2×4 могут обращаться к восстановленным значениям отсчетов в блоке 2 размером 2×4 . Перекомпоновка блоков цветности размером 2×4 значений остатков происходит в кодере или в декодере.

При использовании блока (2440) цветности размером 4×8 , содержащего перекомпонованные блоки размером 2×4 для кодирования и декодирования остатков, область перекрытия поддерживается для левой половины блока яркости и соответствующих блоков цветности (показана серым цветом). В ходе оценки BV кодер может оценить восстановленные значения отсчетов в области перекрытия (как было описано выше) и оценить пригодные значения BV, относящиеся к областям внутреннего предсказания, которые частично или полностью покрывают область перекрытия.

При таком подходе допускаются значения В с ненулевыми горизонтальными компонентами BV, но значения BV имеют нулевые вертикальные компоненты BV. Это подходит к контенту снимка экрана, например, который, как правило, отличается непрерывностью по горизонтали для значений отсчетов.

В примерных реализациях при внутреннем предсказании с ВС используются значения BV с целочисленной точностью отсчетов для блоков яркости и для блоков цветности. Дробные смещения (и дробная интерполяция на восстановленных значениях отсчетов) для значений BV не используются. Когда данные цветности для картинки имеют уменьшенное разрешение по отношению к данным яркости для этой картинки (например, при использовании формата YUV 4:2:0 или формата YUV 4:2:2), значение BV для блока яркости перед тем, как оно применяется к соответствующему блоку цветности, можно масштабировать в сторону уменьшения для корректировки отличия в разрешении цветности, но при этом оно округляется и/или усекается до целочисленного значения точности отсчета.

В качестве альтернативы, значения BV имеют целочисленную точность отсчетов для блоков яркости, но могут иметь дробную точность отсчетов для соответствующих блоков цветности. В частности, когда данные цветности для картинки имеют уменьшенное разрешение по отношению к данным яркости для этой картинки, значение BV из блока яркости, прежде чем оно будет применено к соответствующему блоку цветности, может быть масштабировано в сторону уменьшения для корректировки отличия в разрешении цветности. Для формата YUV 4:2:0 вертикальная и горизонтальная компоненты значения BV могут быть уменьшены вдвое и усечены или округлены,

порождая значение с целочисленной точностью отсчетов или дробной точностью отсчетов, например, с точностью до половины отсчета. Например, значение 3 компоненты BVY, равное 30 для блока яркости масштабируется до значения компоненты BVY, равного 1,5 для соответствующего блока цветности. Для формата YUV 4:2:2 горизонтальная компонента значения BV может быть уменьшена вдвое и усечена или округлена, порождая значение либо с целочисленной точностью, либо с дробной точностью, например, с точностью до половины отсчета. Когда значение BV характеризуется дробной точностью, в качестве интерполяции на восстановленных значениях отсчетов можно использовать билинейную интерполяцию, бикубическую интерполяцию или интерполяцию другого вида.

В качестве альтернативы, значения BV могут иметь предсказание с дробными отсчетами для блоков яркости и соответствующих блоков цветности. Опять же, когда значение BV характеризуется дробной точностью отсчетов, в качестве интерполяции на восстановленных значениях отсчетов можно использовать билинейную интерполяцию, бикубическую интерполяцию или интерполяцию другого вида.

В контексте многих возможных вариантов осуществления, для которых могут быть применимы принципы раскрытого изобретения, следует понимать, что показанные варианты осуществления являются лишь предпочтительными примерами изобретения, и их не следует рассматривать как ограничение объема изобретения. Скорее, объем изобретения определяется нижеследующей формулой изобретения. Таким образом, авторы изобретения считают, что изобретением является все, что не выходит за рамки объема, определяемого приведенной формулой изобретения.

(57) Формула изобретения

1. Способ кодирования данных, выполняемый в вычислительном устройстве, которое реализует видеокодер или кодер изображений, содержащий:

кодирование картинки для получения кодированных данных, при этом кодирование включает в себя этапы, на которых:

оценивают восстановленные значения отсчетов в области перекрытия текущего блока картинки, где текущий блок имеет множество более мелких блоков, причем область перекрытия покрывает части текущего блока, которые потенциально являются областями внутреннего предсказания для по меньшей мере одного из этого множества более мелких блоков, при этом при оценивании восстановленных значений отсчетов в области перекрытия используют соответствующие исходные значения отсчетов из области перекрытия, чтобы оценивать восстановленные значения отсчетов соответственно в области перекрытия без использования восстановленных значений отсчетов из-за пределов области перекрытия вместо восстановленных значений отсчетов соответственно в области перекрытия, и

выполняют оценку блочного вектора (BV) для определения значения BV для текущего блока, где блочный вектор указывает смещение от текущего блока к области картинки, которая включает в себя значения отсчетов, используемых для предсказания, причем при оценке BV используют, по меньшей мере, некоторые из оцененных восстановленных значений отсчетов в области перекрытия текущего блока; и

вывод кодированных данных в качестве части битового потока.

2. Способ по п. 1, в котором упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя этап, на котором используют соответствующие исходные значения отсчетов из области перекрытия в качестве восстановленных значений отсчетов соответственно в области перекрытия.

3. Способ по п. 1, в котором упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя этапы, на которых:

обрабатывают соответствующие исходные значения отсчетов из области перекрытия с использованием частотного преобразования, квантования, обратного квантования и обратного частотного преобразования; и

используют обработанные значения отсчетов в качестве восстановленных значений отсчетов соответственно в области перекрытия.

4. Способ по п. 1, в котором упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя этапы, на которых:

предсказывают значения отсчетов в области перекрытия с использованием по меньшей мере одного предиктора BV;

определяют значения остатков на основе разностей между предсказанными значениями отсчетов и соответствующими исходными значениями отсчетов из области перекрытия;

обрабатывают значения остатков с использованием частотного преобразования, квантования, обратного квантования и обратного частотного преобразования;

объединяют обработанные значения остатков с предсказанными значениями отсчетов; и

используют результаты объединения в качестве восстановленных значений отсчетов соответственно в области перекрытия.

5. Способ по п. 1, в котором упомянутая оценка BV включает в себя, для каждого из одного или более пригодных значений BV для текущего блока, этап, на котором выполняют операции согласования блоков между упомянутыми более мелкими блоками соответственно и пригодными областями внутреннего предсказания в соответствии с пригодным значением BV.

6. Способ по п. 1, в котором упомянутая оценка BV включает в себя этапы, на которых:

для каждого из упомянутого множества более мелких блоков, определяют значение BV для этого более мелкого блока; и

синтезируют значения BV для упомянутого множества более мелких блоков в значение BV для текущего блока.

7. Способ по п. 1, дополнительно содержащий выполнение внутреннего предсказания с блочным копированием (BC) для текущего блока с использованием значения BV для текущего блока, включая выполнение операций BC на поблочной основе для

упомянутого множества более мелких блоков.

8. Способ по п. 1, в котором текущий блок является блоком кодирования единицы кодирования, при этом каждый из множества блоков является блоком преобразования единицы преобразования.

9. Вычислительное устройство, выполненное с возможностью кодирования данных, при этом вычислительное устройство содержит:

кодер, сконфигурированный выполнять операции для получения кодированных данных для картинки, причем кодер представляет собой видеокодер или кодер изображений, каковые операции включают в себя:

оценивание восстановленных значений отсчетов в области перекрытия текущего блока картинки, где текущий блок имеет множество более мелких блоков, причем область перекрытия покрывает части текущего блока, которые потенциально являются областями внутреннего предсказания для по меньшей мере одного из этого множества более мелких блоков, при этом при оценивании восстановленных значений отсчетов в

области перекрытия используются соответствующие исходные значения отсчетов из области перекрытия, чтобы оценивать восстановленные значения отсчетов соответственно в области перекрытия без использования восстановленных значений отсчетов из-за пределов области перекрытия вместо восстановленных значений отсчетов

5 соответственно в области перекрытия, и

выполнение оценки блочного вектора (BV) для определения значения BV для текущего блока, где блочный вектор указывает смещение от текущего блока к области картинки, которая включает в себя значения отсчетов, используемых для предсказания, причем при оценке BV используются, по меньшей мере, некоторые из оцененных

10 восстановленных значений отсчетов в области перекрытия текущего блока; и

буфер, приспособленный для сохранения кодированных данных в качестве части битового потока.

10. Вычислительное устройство по п. 9, в котором упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя

15 использование соответствующих исходных значений отсчетов из области перекрытия в качестве восстановленных значений отсчетов соответственно в области перекрытия.

11. Вычислительное устройство по п. 9, в котором упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя:

обработку соответствующих исходных значений отсчетов из области перекрытия с использованием частотного преобразования, квантования, обратного квантования и обратного частотного преобразования; и

использование обработанных значений отсчетов в качестве восстановленных значений отсчетов соответственно в области перекрытия.

12. Вычислительное устройство по п. 9, в котором упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя:

25 предсказание значений отсчетов в области перекрытия с использованием по меньшей мере одного предиктора BV;

определение значений остатков на основе разностей между предсказанными значениями отсчетов и соответствующими исходными значениями отсчетов из области

30 перекрытия;

обработку значений остатков с использованием частотного преобразования, квантования, обратного квантования и обратного частотного преобразования;

объединение обработанных значений остатков с предсказанными значениями отсчетов; и

35 использование результатов объединения в качестве восстановленных значений отсчетов соответственно в области перекрытия.

13. Вычислительное устройство по п. 9, в котором упомянутая оценка BV включает в себя, для каждого из одного или более пригодных значений BV для текущего блока, выполнение операций согласования блоков между упомянутыми более мелкими блоками

40 соответственно и пригодными областями внутреннего предсказания в соответствии с пригодным значением BV.

14. Вычислительное устройство по п. 9, в котором упомянутая оценка BV включает в себя:

для каждого из упомянутого множества более мелких блоков, определение значения

45 BV для этого более мелкого блока; и

синтезирование значений BV для упомянутого множества более мелких блоков в значение BV для текущего блока.

15. Вычислительное устройство по п. 9, в котором операции дополнительно содержат

выполнение внутреннего предсказания с блочным копированием (BC) для текущего блока с использованием значения BV для текущего блока, включая выполнение операций BC на поблочной основе для упомянутого множества более мелких блоков.

16. Вычислительное устройство по п. 9, при этом текущий блок является блоком кодирования единицы кодирования, при этом каждый из множества блоков является блоком преобразования единицы преобразования.

17. Машиночитаемый носитель, на котором сохранены машиноисполняемые команды для предписания вычислительному устройству, запрограммированному ими, выполнять операции, содержащие:

кодирование картинки для получения кодированных данных, при этом кодирование включает в себя:

оценивание восстановленных значений отсчетов в области перекрытия текущего блока картинки, где текущий блок имеет множество более мелких блоков, причем область перекрытия покрывает части текущего блока, которые потенциально являются областями внутреннего предсказания для по меньшей мере одного из этого множества более мелких блоков, при этом при оценивании восстановленных значений отсчетов в области перекрытия используются соответствующие исходные значения отсчетов из области перекрытия, чтобы оценивать восстановленные значения отсчетов соответственно в области перекрытия без использования восстановленных значений отсчетов из-за пределов области перекрытия вместо восстановленных значений отсчетов соответственно в области перекрытия, и

выполнение оценки блочного вектора (BV) для определения значения BV для текущего блока, где блочный вектор указывает смещение от текущего блока к области картинки, которая включает в себя значения отсчетов, используемых для предсказания, причем при оценке BV используются, по меньшей мере, некоторые из оцененных восстановленных значений отсчетов в области перекрытия текущего блока; и вывод кодированных данных в качестве части битового потока.

18. Машиночитаемый носитель по п. 17, при этом упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя использование соответствующих исходных значений отсчетов из области перекрытия в качестве восстановленных значений отсчетов соответственно в области перекрытия.

19. Машиночитаемый носитель по п. 17, при этом упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя:

обработку соответствующих исходных значений отсчетов из области перекрытия с использованием частотного преобразования, квантования, обратного квантования и обратного частотного преобразования; и

использование обработанных значений отсчетов в качестве восстановленных значений отсчетов соответственно в области перекрытия.

20. Машиночитаемый носитель по п. 17, при этом упомянутое оценивание восстановленных значений отсчетов в области перекрытия включает в себя:

предсказание значений отсчетов в области перекрытия с использованием по меньшей мере одного предиктора BV;

определение значений остатков на основе разностей между предсказанными значениями отсчетов и соответствующими исходными значениями отсчетов из области перекрытия;

обработку значений остатков с использованием частотного преобразования, квантования, обратного квантования и обратного частотного преобразования;

объединение обработанных значений остатков с предсказанными значениями

отсчетов; и

использование результатов объединения в качестве восстановленных значений отсчетов соответственно в области перекрытия.

5

10

15

20

25

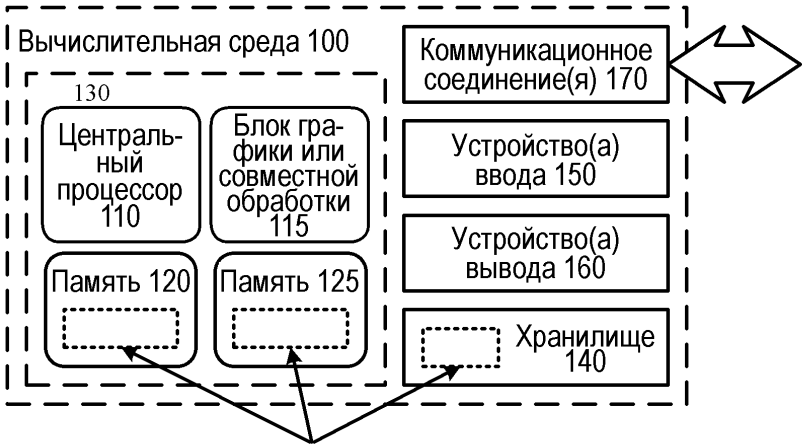
30

35

40

45

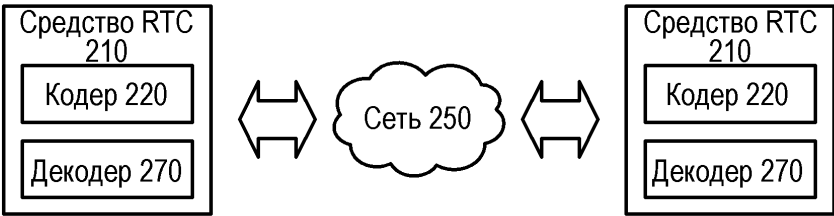
ФИГ. 1



Программное обеспечение 180, реализующее одно или несколько нововведений для внутреннего предсказания с блочным ко-пированием

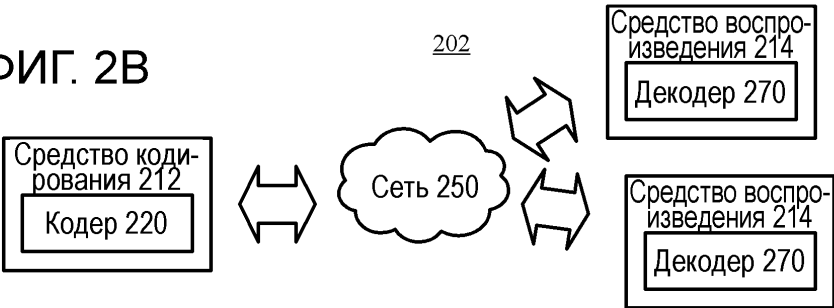
ФИГ. 2А

201

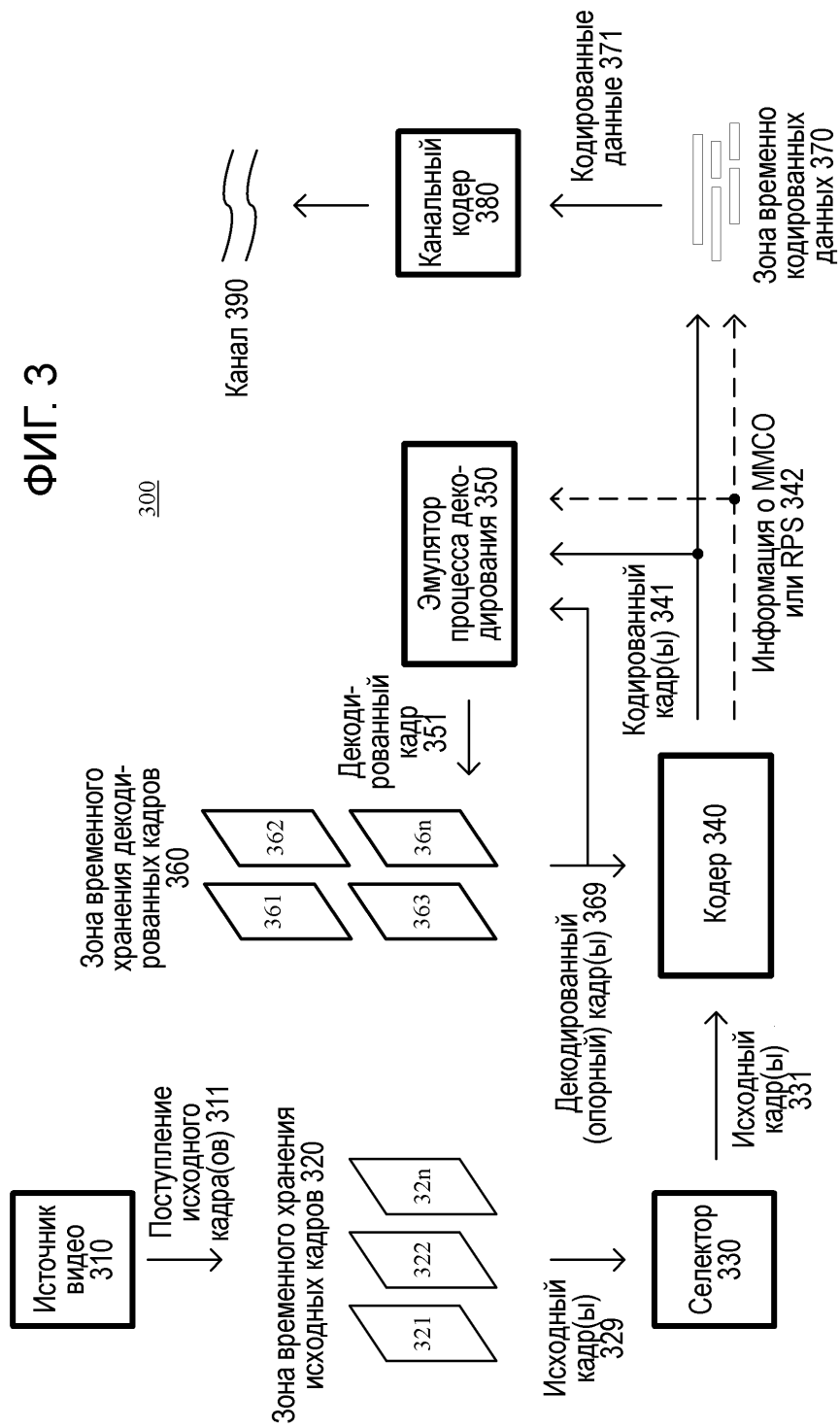


ФИГ. 2В

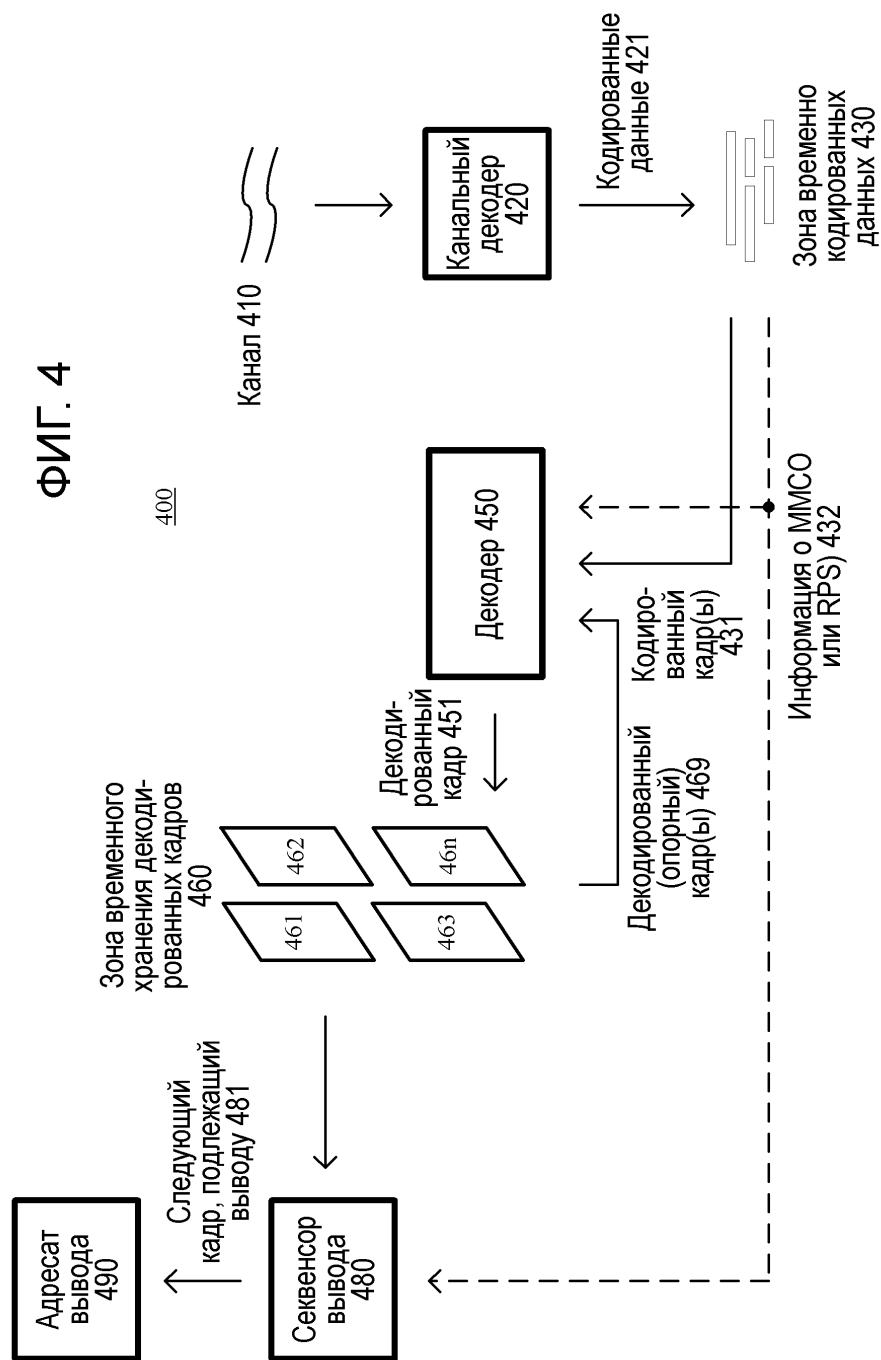
202



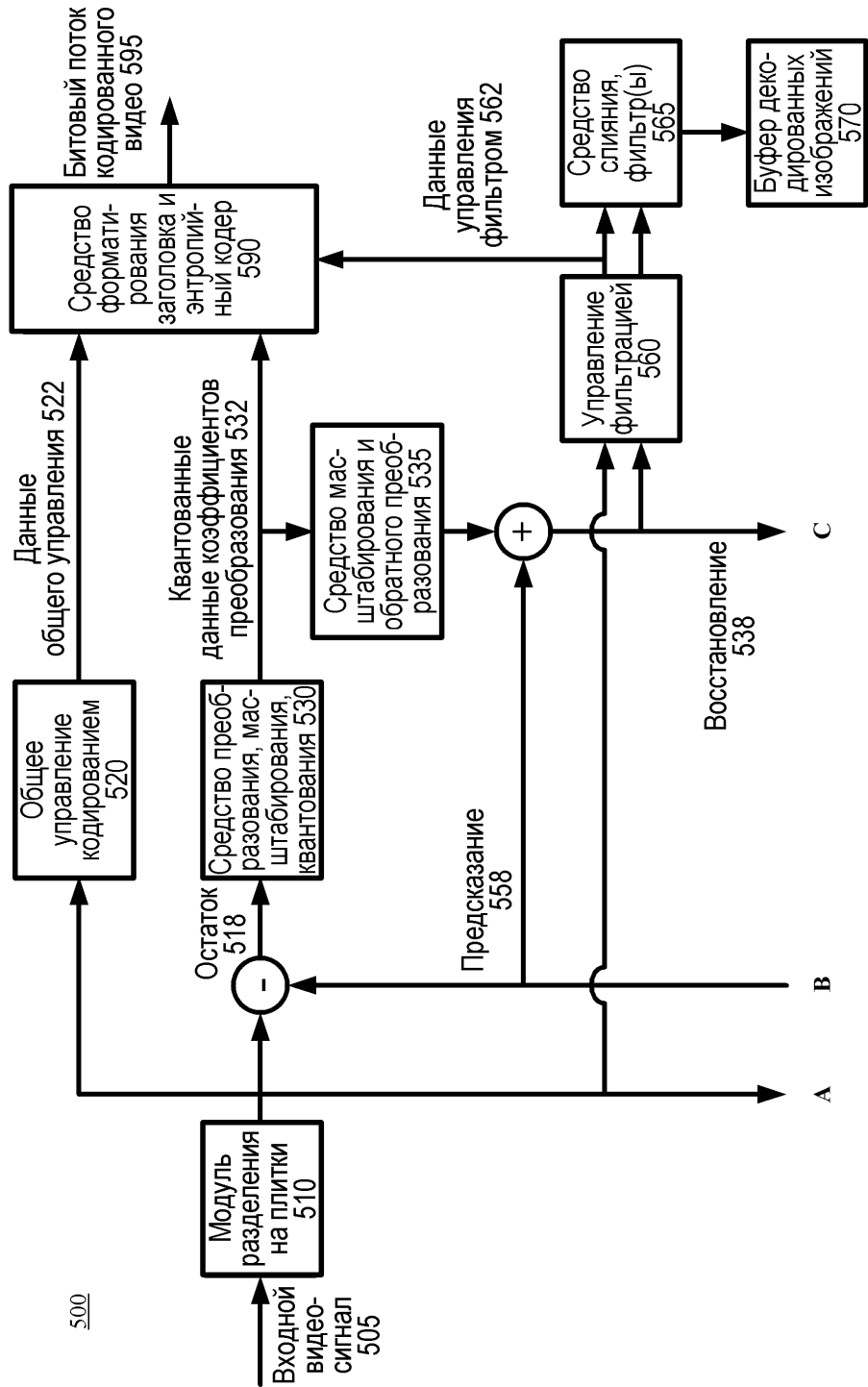
ФИГ. 3

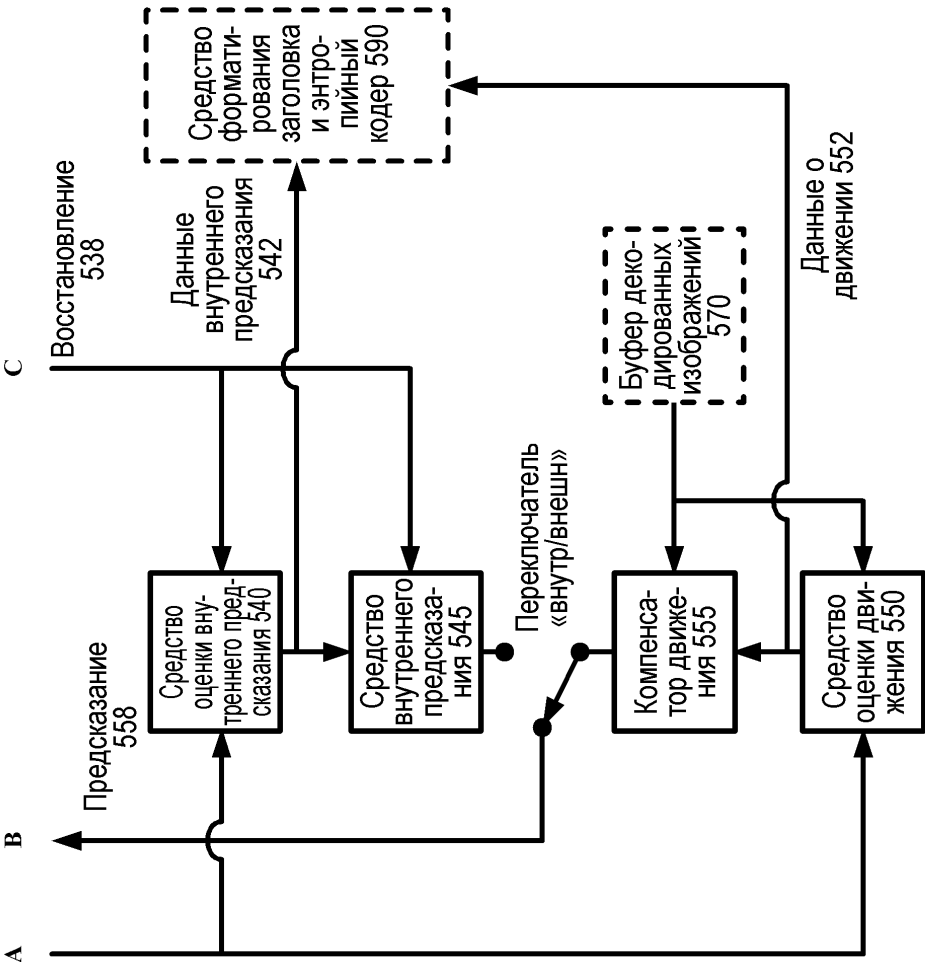


ФИГ. 4



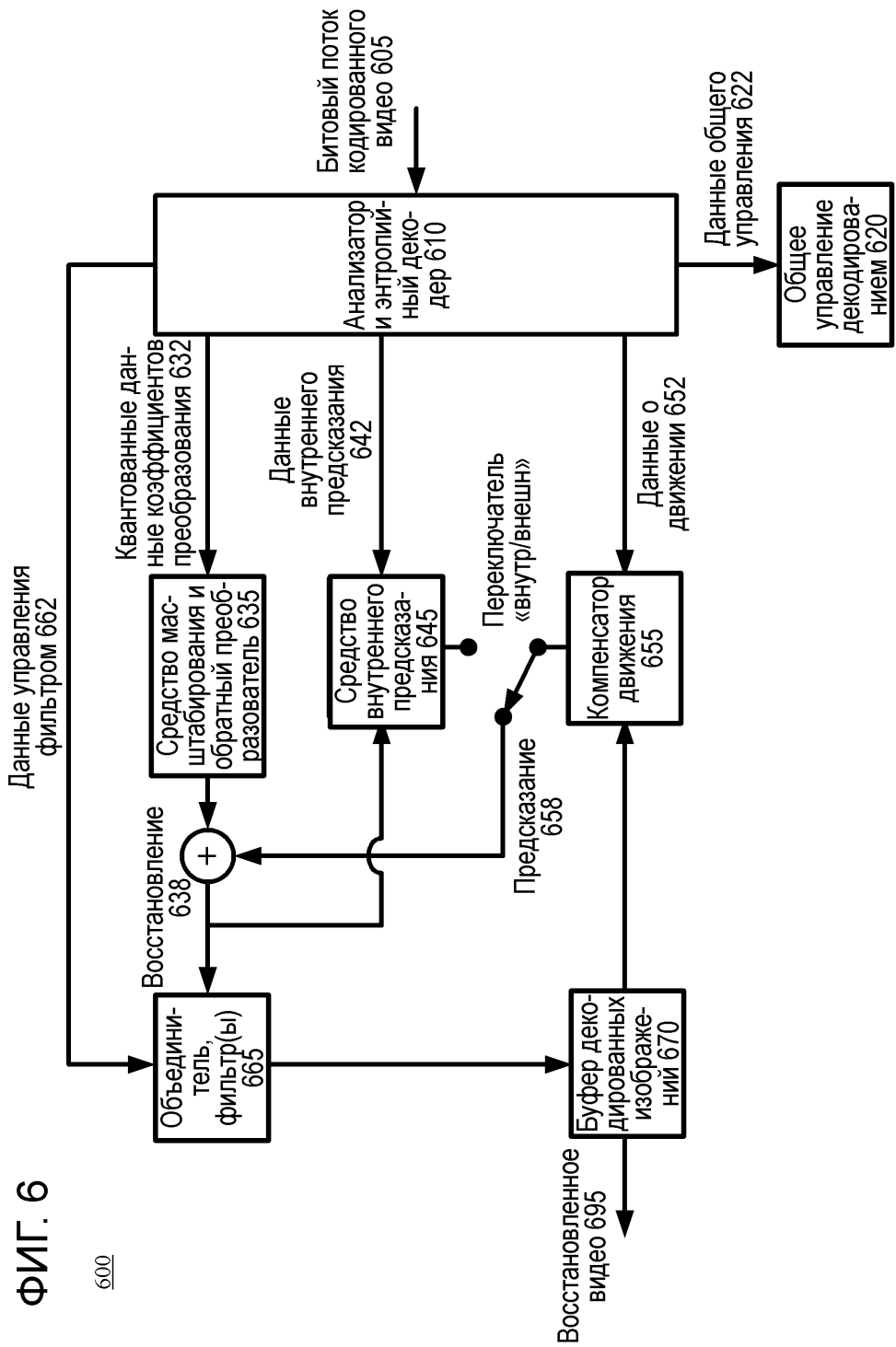
ФИГ. 5А





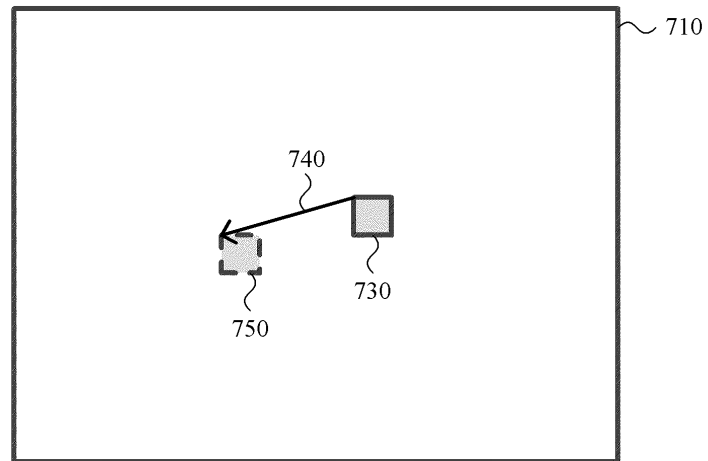
ФИГ. 5В

500

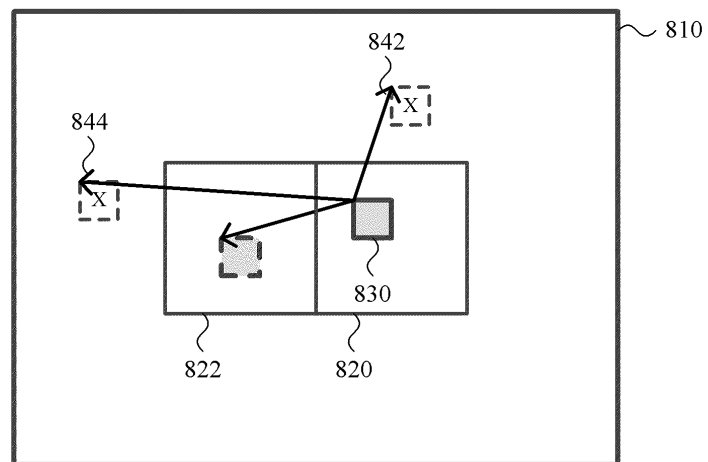


7/17

ФИГ. 7



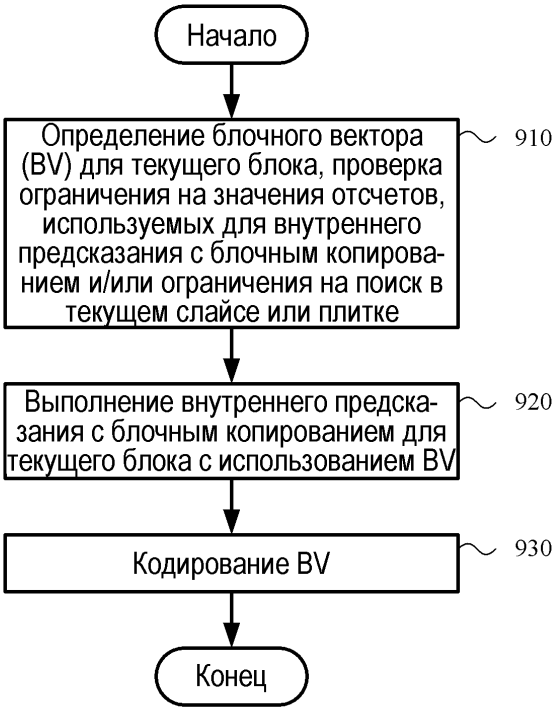
ФИГ. 8



8/17

ФИГ. 9

900



ФИГ. 10

1000



9/17

ФИГ. 11



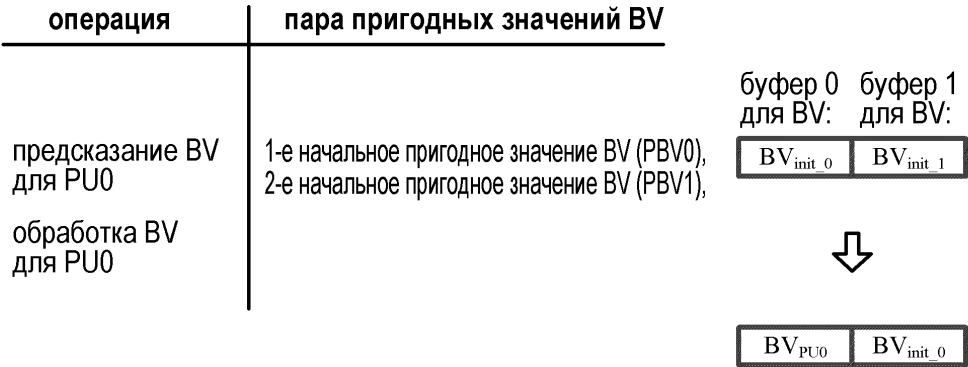
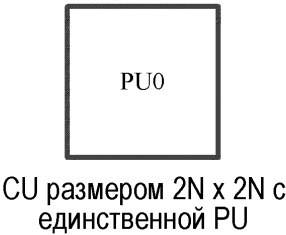
ФИГ. 12

1200

Значение индекса:	предиктор BV:
idx 0:	первое пригодное значение (BV_{init_0})
idx 1:	второе пригодное значение (BV_{init_0})

ФИГ. 13

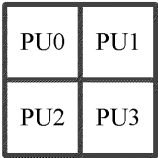
1300



10/17

ФИГ. 14

1400



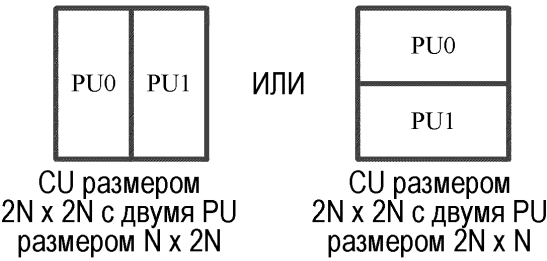
CU размером 2N x 2N с
четырьмя PU размером N x N

операция	пара пригодных значений BV			
		буфер 0 для BV: буфер 1 для BV:		
предсказание BV для PU0	1-е начальное пригодное значение BV (PBV0), 2-е начальное пригодное значение BV (PBV1),	<table><tr><td>BV_{init 0}</td><td>BV_{init 1}</td></tr></table>	BV _{init 0}	BV _{init 1}
BV _{init 0}	BV _{init 1}			
обработка BV для PU0		↓		
предсказание BV для PU1	BV для PU0, 1-е начальное пригодное значение BV (PBV1),	<table><tr><td>BV_{PU0}</td><td>BV_{init 0}</td></tr></table>	BV _{PU0}	BV _{init 0}
BV _{PU0}	BV _{init 0}			
обработка BV для PU1		↓		
предсказание BV для PU2	BV для PU1, BV для PU0	<table><tr><td>BV_{PU1}</td><td>BV_{PU0}</td></tr></table>	BV _{PU1}	BV _{PU0}
BV _{PU1}	BV _{PU0}			
обработка BV для PU2		↓		
предсказание BV для PU3	BV для PU1, BV для PU2	<table><tr><td>BV_{PU1}</td><td>BV_{PU0}</td></tr></table>	BV _{PU1}	BV _{PU0}
BV _{PU1}	BV _{PU0}			
обработка BV для PU3				

11/17

ФИГ. 14

1500

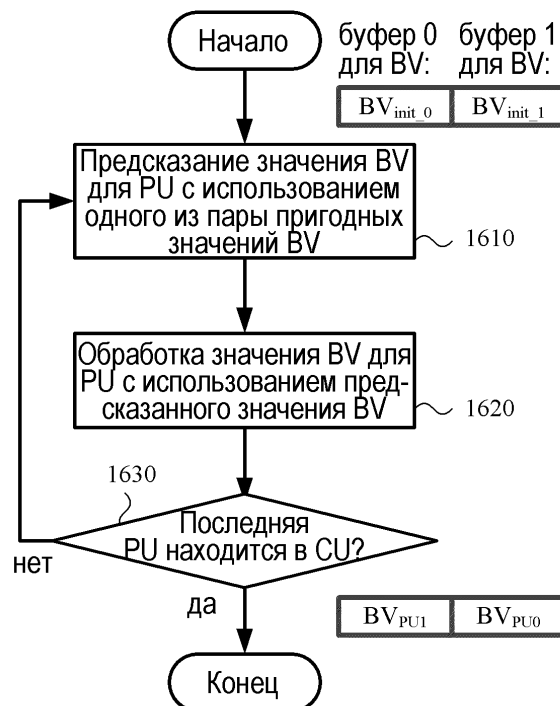


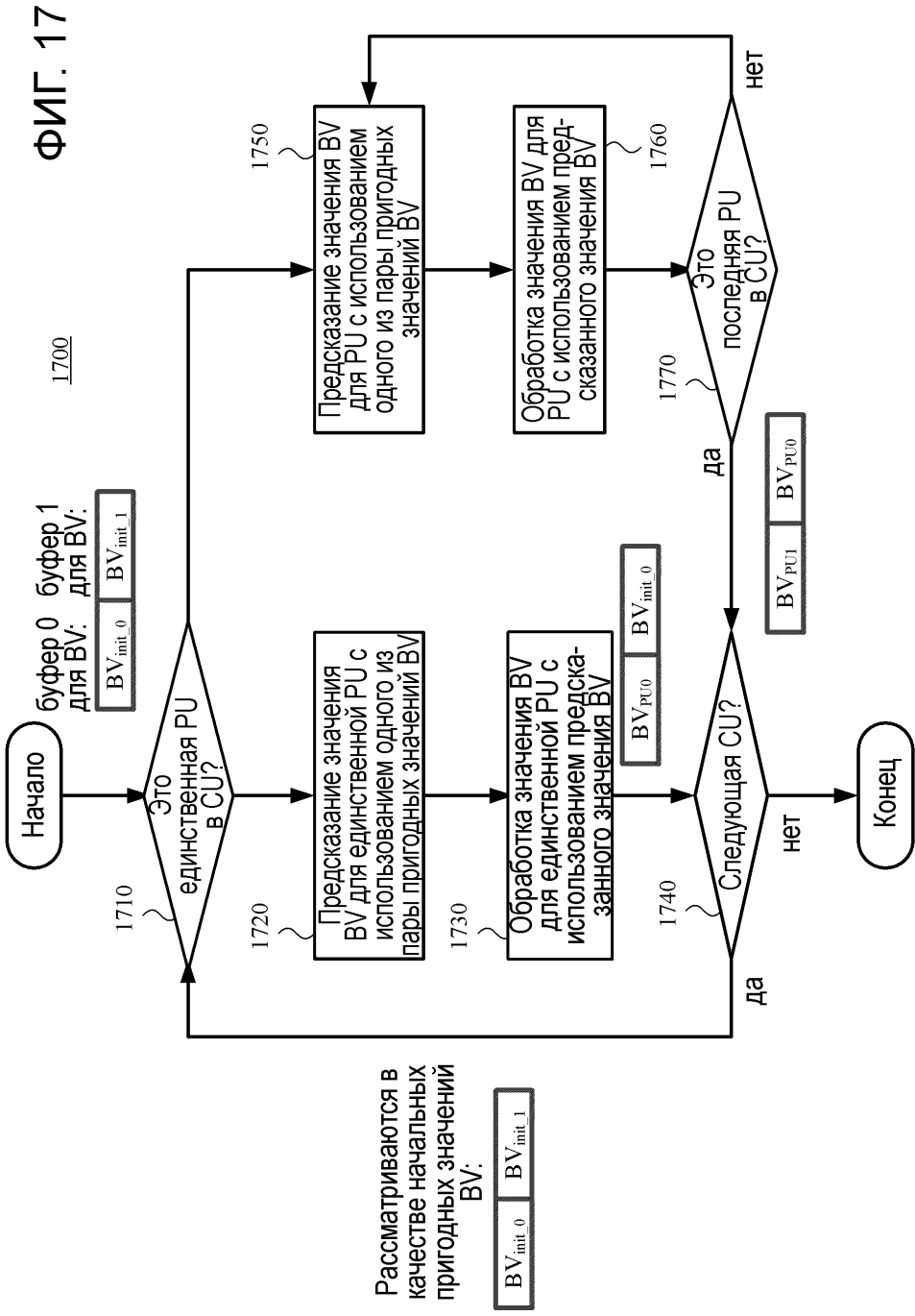
операция	пара пригодных значений BV	
предсказание BV для PU0	1-е начальное пригодное значение BV (PBV0), 2-е начальное пригодное значение BV (PBV1),	буфер 0 для BV: BV_{init 0} буфер 1 для BV: BV_{init 1}
обработка BV для PU0		↓
предсказание BV для PU1	BV для PU0, 1-е начальное пригодное значение BV (PBV0),	BV_{PU0} BV_{init 0}
обработка BV для PU1		↓
		BV_{PU1} BV_{PU0}

12/17

ФИГ. 16

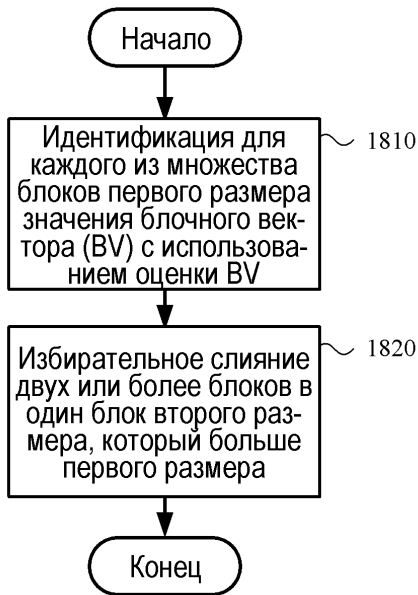
1600





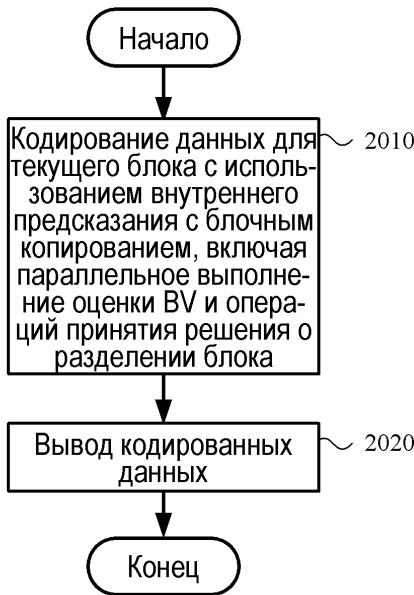
ФИГ. 18

1800



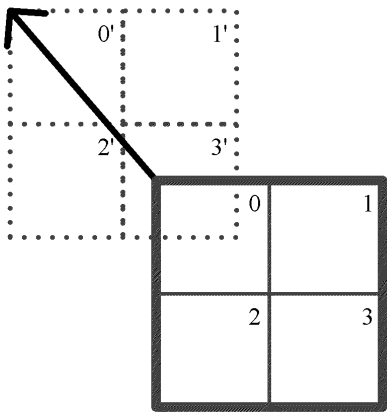
ФИГ. 20

2000



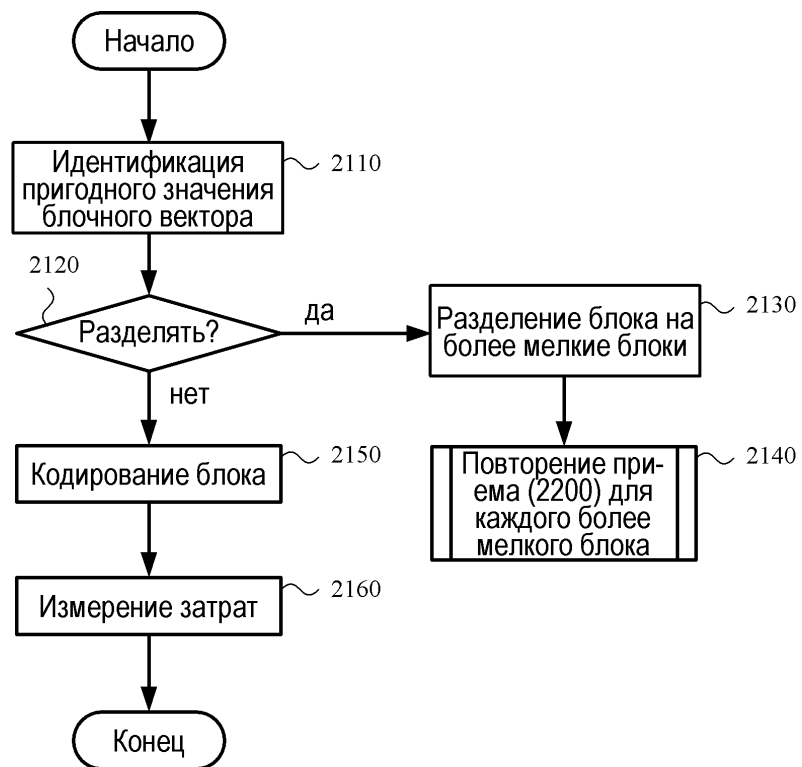
ФИГ. 19

1900

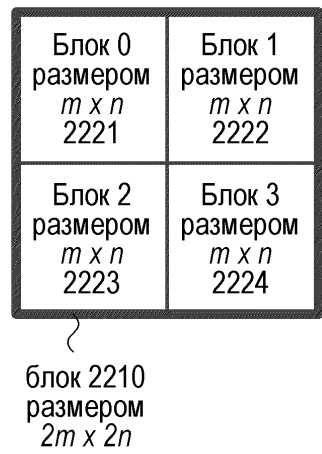


15/17

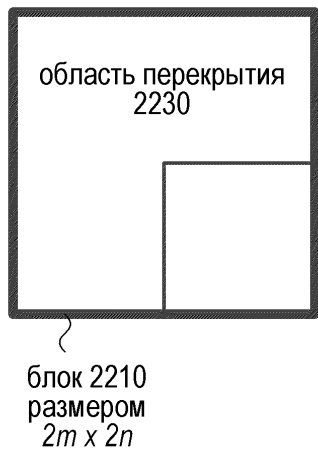
ФИГ. 21

2100

ФИГ. 22А

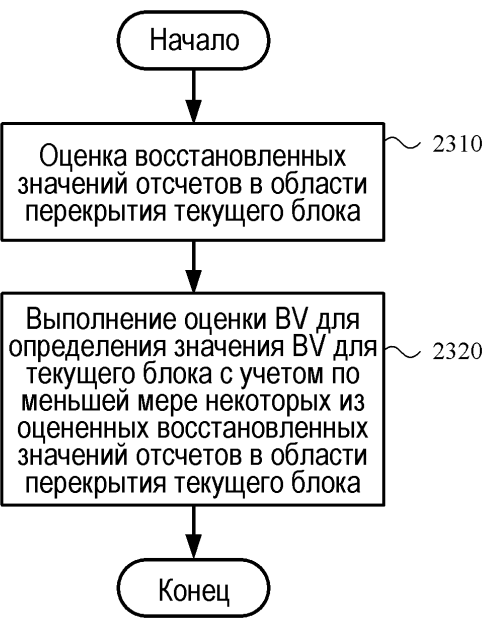


ФИГ. 22В



ФИГ. 23

2300



17/17

ФИГ. 24

