

(19)



**Евразийское  
патентное  
ведомство**

(11) **035270**(13) **B1**(12) **ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ**

- |                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>(45) Дата публикации и выдачи патента<br/><b>2020.05.22</b></p> <p>(21) Номер заявки<br/><b>201791449</b></p> <p>(22) Дата подачи заявки<br/><b>2016.01.26</b></p> | <p>(51) Int. Cl. <b>H04N 19/503</b> (2014.01)<br/><b>H04N 19/51</b> (2014.01)<br/><b>H04N 19/593</b> (2014.01)<br/><b>H04N 19/70</b> (2014.01)<br/><b>H04N 19/52</b> (2014.01)<br/><b>H04N 19/61</b> (2014.01)<br/><b>H04N 19/176</b> (2014.01)<br/><b>H04N 19/56</b> (2014.01)<br/><b>H04N 19/46</b> (2014.01)<br/><b>H04N 19/573</b> (2014.01)<br/><b>H04N 19/105</b> (2014.01)</p> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

(54) **СПОСОБ И УСТРОЙСТВО КОДИРОВАНИЯ ВИДЕОДАНЫХ**

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>(31) <b>62/107,933; 15/005,564</b></p> <p>(32) <b>2015.01.26; 2016.01.25</b></p> <p>(33) <b>US</b></p> <p>(43) <b>2017.12.29</b></p> <p>(86) <b>PCT/US2016/014875</b></p> <p>(87) <b>WO 2016/123081 2016.08.04</b></p> <p>(71)(73) Заявитель и патентовладелец:<br/><b>КВЭЛКОММ ИНКОРПОРЕЙТЕД</b><br/><b>(US)</b></p> <p>(72) Изобретатель:<br/><b>Чэнь Ин, Ли Сян, Лю Хунбинь, Чэнь</b><br/><b>Цзяньлэ, Чжан Ли, Карчевич Марта</b><br/><b>(US)</b></p> <p>(74) Представитель:<br/><b>Медведев В.Н. (RU)</b></p> | <p>(56) CHEN JIANLE ET AL.: "Coding tools investigation for next generation video coding based on HEVC", OPTOMECHATRONIC MICRO/NANO DEVICES AND COMPONENTS III : 8 - 10 OCTOBER 2007, LAUSANNE, SWITZERLAND; [PROCEEDINGS OF SPIE, ISSN 0277-786X], SPIE, BELLINGHAM, WASH, vol. 9599, 22 September 2015 (2015-09-22), pages 95991B-95991B, XP060060836, DOI: 10.1117/12.2193681 ISBN: 978-1-62841-730-2, the whole document</p> <p>KARCZEWICZ M. ET AL.: "Study of coding efficiency improvements beyond HEVC", 113. MPEG MEETING; 19-10-2015 - 23-10-2015; GENEVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), no. m37102, 15 October 2015 (2015-10-15), XP030065470, the whole document</p> <p>"High Efficiency Video Coding (HEVC) Encoder Description v 16 (HM16)", 109. MPEG MEETING; 7-7-2014 - 11-7-2014; SAPPORO; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), no. N14703, 19 October 2014 (2014-10-19), XP030021439, the whole document</p> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
- (57) Устройство для кодирования видеоданных, выполненное с возможностью формирования для текущего блока видеоданных списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков к текущему блоку и, непосредственно после четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), кодирования индекса в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и кодирования текущего блока видеоданных с использованием информации о движении идентифицированного кандидата слияния.

**B1****035270****035270****B1**

Настоящая заявка испрашивает приоритет предварительной патентной заявки США 62/107,933, поданной 26 января 2015, все содержание которой включено в настоящий документ посредством ссылки.

### **Область техники**

Данное изобретение относится к кодированию видео.

### **Предшествующий уровень техники**

Возможности цифрового видео могут быть включены в широкий спектр устройств, включая цифровые телевизоры, системы цифрового прямого вещания, беспроводные широкоэмитательные системы, персональные цифровые помощники (PDA), портативные или настольные компьютеры, планшетные компьютеры, электронные книги, цифровые камеры, устройства цифровой записи, цифровые медиаплееры, видеоигровые устройства, игровые приставки, сотовые или спутниковые радиотелефоны, так называемые "смартфоны", устройства видеоконференцсвязи, устройства потоковой передачи видео и т.п. Цифровые видеоустройства реализуют методы кодирования видео, такие как те, которые описаны в стандартах, определенных MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, часть 10, усовершенствованное кодирование видео (AVC), ITU-T H.265, также упоминаемое как высокоэффективное кодирование видео (HEVC), и расширениях таких стандартов. Видеоустройства могут более эффективно передавать, принимать, кодировать, декодировать и/или сохранять цифровую видеoinформацию, применяя такие методы кодирования видео.

Методы кодирования видео включают в себя пространственное (внутрикадровое, "внутри картинки") предсказание и/или временное (межкадровое, "между картинками") предсказание для уменьшения или удаления избыточности, присущей видеопоследовательностям. Для блочного кодирования видео, видео вырезка (например, видеокادر или часть видеокадра) может быть разделена на видеоблоки, которые для некоторых методов также могут упоминаться как древовидные блоки, единицы кодирования (CU) и/или узлы кодирования. Видеоблоки во внутри-(интра-)кодируемой (I) вырезке изображения (картинки) кодируются с использованием пространственного предсказания относительно опорных выборок в соседних блоках в той же самой картинке. Видеоблоки во взаимно-(интер-)кодируемой (P или B) вырезке картинки могут использовать пространственное предсказание относительно опорных выборок в соседних блоках в той же самой картинке или временное предсказание относительно опорных выборок в других опорных картинках. Картинки могут упоминаться как кадры, и опорные картинки могут упоминаться как опорные кадры.

Пространственное или временное предсказание приводит к блоку предсказания для блока, подлежащего кодированию. Остаточные данные представляют пиксельные разности между исходным блоком, подлежащим кодированию, и блоком предсказания. Интер-кодированный блок кодируется в соответствии с вектором движения, который указывает на блок опорных выборок, формирующих блок предсказания, и остаточными данными, указывающими разницу между кодированным блоком и блоком предсказания. Интра-кодированный блок кодируется в соответствии с режимом интра-кодирования и остаточными данными. Для дальнейшего сжатия остаточные данные могут быть преобразованы из пиксельной области в область преобразования, что приводит к остаточным коэффициентам преобразования, которые затем могут быть квантованы. Квантованные коэффициенты преобразования, первоначально упорядоченные в двумерном массиве, могут сканироваться для получения одномерного вектора коэффициентов преобразования, и энтропийное кодирование может применяться для достижения еще большего сжатия.

### **Сущность изобретения**

В общем, настоящее раскрытие описывает методы, относящиеся к кодированию (например, кодированию или декодированию) информации о движении для блока видеоданных. Более конкретно, устройство кодирования видео (например, видеокодер или декодер) может быть сконфигурировано для кодирования информации о движении для текущего блока (например, текущей единицы предсказания (PU)) с использованием расширенного предсказания временного вектора движения (ATMVP). ATMVP обычно включает в себя использование временного вектора движения для идентификации соответствующей единицы предсказания, которая разбивается на под-PU. Вместо того чтобы разбивать текущую PU на под-PU, временной вектор движения может просто идентифицировать соответствующий блок, который разбит на под-PU, каждый из которых имеет свою собственную информацию о движении, и устройство кодирования видео может быть сконфигурировано, чтобы предсказывать соответствующие части текущего блока с использованием информации о движении соответствующих под-PU. Избегая активного разделения текущего блока, информация сигнализации служебной нагрузки может быть уменьшена для текущего блока, при этом все еще достигается мелкозернистое предсказание для частей текущего блока, что в противном случае может быть результатом разделения текущего блока на под-PU.

В одном примере способ кодирования видеоданных включает в себя формирование для текущего блока видеоданных списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков к текущему блоку и, непосредственно после четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), кодирование индекса в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и кодирование текущего блока видеоданных с

использованием информации о движении идентифицированного кандидата слияния.

В другом примере устройство для кодирования видеоданных включает в себя память, сконфигурированную для хранения видеоданных, и устройство кодирования видео, сконфигурированное для формирования для текущего блока видеоданных, списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков к текущему блоку и, непосредственно после четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), кодирования индекса в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и кодирования текущего блока видеоданных с использованием информации о движении идентифицированного кандидата слияния.

В другом примере устройство для кодирования видеоданных включает в себя средство для формирования для текущего блока видеоданных списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков к текущему блоку и, непосредственно после четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), средство для кодирования индекса в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и средство для кодирования текущего блока видеоданных с использованием информации о движении идентифицированного кандидата слияния.

В другом примере считываемый компьютером носитель хранения данных содержит сохраненные в нем инструкции, которые при их исполнении побуждают процессор формировать для текущего блока видеоданных список кандидатов слияния, включающий в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков в текущий блок и, непосредственно после четырех пространственных соседних кандидатов, кандидата слияния расширенного предсказания временного вектора движения (ATMVP), кодировать индекс в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и кодировать текущий блок видеоданных с использованием информации о движении идентифицированного кандидата слияния.

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

### **Краткое описание чертежей**

Фиг. 1 - блок-схема, иллюстрирующая примерную систему кодирования и декодирования видео, которая может реализовывать методы для расширенного предсказания временного вектора движения (ATMVP).

Фиг. 2 - блок-схема, иллюстрирующая пример видеокодера, который может реализовывать методы для расширенного предсказания временного вектора движения (ATMVP).

Фиг. 3 - блок-схема, иллюстрирующая пример видеокодера, который может реализовывать методы для расширенного предсказания временного вектора движения (ATMVP).

Фиг. 4 - концептуальная диаграмма, иллюстрирующая пространственных соседних кандидатов в высокоэффективном кодировании видео (HEVC).

Фиг. 5 - концептуальная диаграмма, иллюстрирующая временное предсказание вектора движения (TMVP) в HEVC.

Фиг. 6 - концептуальная диаграмма, иллюстрирующая примерную структуру предсказания для 3D-HEVC.

Фиг. 7 - концептуальная диаграмма, иллюстрирующая межвидовое предсказание движения на основе под-PU в 3D-HEVC.

Фиг. 8 - концептуальная диаграмма, иллюстрирующая предсказание движения на основе под-PU из опорной картинки.

Фиг. 9 - концептуальная диаграмма, иллюстрирующая релевантные картинки в ATMVP (аналогично TMVP).

Фиг. 10 - блок-схема последовательности операций, иллюстрирующая примерный способ добавления кандидата ATMVP к списку кандидатов во время процесса кодирования в соответствии с методами настоящего раскрытия.

Фиг. 11 - блок-схема последовательности операций, иллюстрирующая примерный способ добавления кандидата ATMVP к списку кандидатов во время процесса декодирования в соответствии с методами настоящего раскрытия.

### **Подробное описание**

В общем, это раскрытие относится к предсказанию вектора движения в видео кодеках. Более конкретно, расширенное предсказание временного вектора движения реализуется путем сбора векторов движения на уровне подблока (под-PU) для данного блока (единицы предсказания).

Стандарты кодирования видео включают в себя ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 или ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual и ITU-T H.264 (также известный как ISO/IEC MPEG-4 AVC), включая его расширения для масштабируемого кодирования видео (SVC) и многовидового кодирования видео (MVC). Один совместный проект MVC описан в "Advanced video coding for generic audiovisual services" (Расширенное кодирование видео для общих аудиовизуальных сервисов), ITU-T Рекомендация H.264, март 2010.

Кроме того, существует недавно разработанный стандарт кодирования видео, а именно, высокоэффективное кодирование видео (HEVC), разработанное Объединенной группой сотрудничества по кодированию видео (JCT-VC) ITU-T Группы экспертов по кодированию видео (VCEG) и ISO/IEC Группы экспертов по движущимся изображениям (MPEG). Недавний проект HEVC доступен из [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v34.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip). Стандарт HEVC также представлен совместно в Рекомендации ITU-T H.265 и Международном стандарте ISO/IEC 23008-2, которые озаглавлены "High efficiency video coding" (Высокоэффективное кодирование видео) и оба опубликованы в октябре 2014.

Информация о движении: для каждого блока может быть доступен набор информации о движении. Набор информации о движении содержит информацию о движении для прямого и обратного направлений предсказания. Здесь прямое и обратное направления предсказания представляют собой два направления предсказания режима двунаправленного предсказания, и термины "прямое" и "обратное" не обязательно имеют геометрическое значение; вместо этого они соответствуют списку 0 опорных картинок (RefPicList0) и списку 1 опорных картинок (RefPicList1) текущей картинке. Когда доступен только один список опорных картинок для картинки или вырезки, доступным является только RefPicList0, и информация о движении каждого блока вырезки всегда соответствует прямому направлению.

Для каждого направления предсказания информация о движении должна содержать опорный индекс и вектор движения. В некоторых случаях для простоты можно сослаться на сам вектор движения таким образом, что предполагается, что он имеет ассоциированный опорный индекс. Опорный индекс используется для идентификации опорной картинки в текущем списке опорных картинок (RefPicList0 или RefPicList1). Вектор движения имеет горизонтальный и вертикальный компонент.

Порядковый отсчет картинки (POC) широко используется в стандартах кодирования видео, чтобы идентифицировать порядок отображения картинки. Хотя бывают случаи, когда две картинки в одной последовательности кодированного видео могут иметь одно и то же значение POC, обычно это не имеет места в последовательности кодированного видео. Когда в битовом потоке присутствуют несколько последовательностей кодированного видео, картинки с одинаковым значением POC могут быть ближе друг к другу с точки зрения порядка декодирования. Значения POC картинок обычно используются для построения списка опорных картинок, получения наборов опорных картинок, как в HEVC и масштабировании вектора движения.

Структура макроблока (MB) в расширенном кодировании видео (AVC) (H.264): В H.264/AVC, каждый интер-макроблок (MB) может быть разделен четырьмя разными способами:

Один 16×16 раздел MB

Два 16×8 раздела MB

Два 8×16 раздела MB

Четыре 8×8 раздела MB

Различные разделы MB в одном MB могут иметь разные значения опорного индекса для каждого направления (RefPicList0 или RefPicList1).

Когда MB не разбит на четыре 8×8 раздела MB, он имеет только один вектор движения для каждого раздела MB в каждом направлении.

Когда MB разбит на четыре 8×8 раздела MB, каждый 8×8 раздел MB может быть дополнительно разбит на подблоки, каждый из которых может иметь отличающийся вектор движения в каждом направлении. Существует четыре разных способа получения подблоков из 8×8 раздела MB:

Один 8×8 подблок

Два 8×4 подблока

Два 4×8 подблока

Четыре 4×4 подблока

Каждый подблок может иметь отличающийся вектор движения в каждом направлении. Поэтому вектор движения присутствует на уровне, более высоком, чем уровень подблока.

Временной прямой режим в AVC: в AVC временной прямой режим может быть разрешен на уровне MB или уровне раздела MB для пропуска или прямого режима в B-вырезках. Для каждого раздела MB векторы движения блока, расположенного рядом с текущим разделом MB в RefPicList1[0] текущего блока, используются для вывода векторов движения. Каждый вектор движения в рядом расположенном блоке масштабируется на основе расстояний POC.

Пространственный прямой режим в AVC: в AVC прямой режим также может предсказывать информацию о движении из пространственных соседей.

Структура единицы кодирования (CU) в высокоэффективном кодировании видео (HEVC): в HEVC наибольшая единица кодирования в вырезке называется блоком дерева кодирования (CTB) или единицей дерева кодирования (CTU). CTB содержит квадродерево, узлы которого являются единицами кодирования.

Размер CTB может составлять от  $16 \times 16$  до  $64 \times 64$  в основном профиле HEVC (хотя технически размеры  $8 \times 8$  CTB могут поддерживаться). Единица кодирования (CU) может быть того же размера CTB, если даже равна  $8 \times 8$ . Каждая единица кодирования кодируется одним режимом. Когда CU является взаимно-(интер)-кодированной, она может дополнительно разбиваться на 2 или 4 единицы предсказания (PU) или становиться только одной PU, когда дальнейшее деление не применяется. Когда две PU присутствуют в одной CU, они могут быть прямоугольниками половинного размера или двумя размерами прямоугольника с размером  $\frac{1}{4}$  или  $\frac{3}{4}$  CU.

Когда CU является интер-кодированной, один набор информации о движении присутствует для каждой PU. Кроме того, каждая PU кодируется с уникальным режимом интер-предсказания для получения набора информации о движении.

Предсказание движения в HEVC: В стандарте HEVC существуют два режима интер-предсказания, называемых режимом слияния (пропуск рассматривается как частный случай слияния) и расширенным режимом предсказания вектора движения (AMVP), соответственно, для единицы предсказания (PU).

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

Список кандидатов MV содержит до 5 кандидатов для режима слияния и только два кандидата для режима AMVP. Кандидат слияния может содержать набор информации о движении, например векторы движения, соответствующие как спискам опорных картинок (списку 0 и списку 1), так и опорным индексам. Если кандидат слияния идентифицируется индексом слияния, опорные картинки используются для предсказания текущих блоков, а также определяются ассоциированные векторы движения. Однако в режиме AMVP для каждого потенциального направления предсказания из списка 0 или списка 1 необходимо явно сигнализировать опорный индекс вместе с индексом MVP в список кандидатов MV, поскольку кандидат AMVP содержит только вектор движения. В режиме AMVP предсказанные векторы движения могут быть дополнительно уточнены.

Как можно видеть выше, кандидат слияния соответствует полному набору информации о движении, в то время как кандидат AMVP содержит только один вектор движения для конкретного направления предсказания и опорного индекса.

Кандидаты для обоих режимов выводятся аналогично из тех же пространственных и временных соседних блоков.

Структура под-PU для 2D видеокодека, особенно относящаяся к расширенному TMVP, может столкнуться со следующими проблемами. Процесс предсказания временного вектора движения на основе под-PU может быть реализован путем определения такого процесса как дополнительного кандидата, а именно кандидата ATMVP.

Однако для такого кандидата ATMVP существуют следующие проблемы проектирования.

1. Хотя кандидат ATMVP может быть вставлен в качестве дополнительного кандидата как TMVP, позиция такого кандидата ATMVP, а также взаимодействие с кандидатом TMVP для достижения более высокой эффективности кодирования неизвестна.

2. Непонятно, как определить доступность кандидата ATMVP; это было бы сложной задачей, если бы все векторы движения всех под-PU следовало проверять, чтобы определить, является ли кандидат ATMVP недоступным и, следовательно, может быть вставлен в кандидата ATMVP.

3. Может потребоваться процесс обрезки для кандидата ATMVP; однако обрезка для такого кандидата может быть сложной.

4. Различные другие детали проектирования для кандидата ATMVP для достижения наилучшего компромисса между эффективностью и сложностью кодирования остаются неизвестными.

На фиг. 1 показана блок-схема, иллюстрирующая пример системы 10 кодирования и декодирования видео, которая может использовать методы для реализации расширенного предсказания временного вектора движения (ATMVP). Как показано на фиг. 1, система 10 включает в себя устройство-источник 12, которое обеспечивает закодированные видеоданные, которые позднее будут декодированы устройством-получателем 14. В частности, устройство-источник 12 предоставляет видеоданные устройству-получателю 14 посредством считываемого компьютером носителя 16. Устройство-источник 12 и устройство-получатель 14 могут содержать любое из широкого спектра устройств, включая настольные компьютеры, ноутбуки (например, портативные компьютеры), планшетные компьютеры, телевизионные приставки, телефонные трубки, такие, как так называемые "смарт"-фоны, так называемые "смарт"-планшеты, телевизоры, камеры, устройства отображения, цифровые медиаплееры, видеоигровые приставки, устройство потоковой передачи видео и т.п. В некоторых случаях устройство-источник 12 и устройство-получатель 14 могут быть оснащены для осуществления беспроводной связи.

Устройство-получатель 14 может принимать кодированные видеоданные, подлежащие декодированию, через считываемый компьютером носитель 16. Считываемый компьютером носитель 16 может содержать любой тип носителя или устройства, способного перемещать закодированные видеоданные из устройства-источника 12 в устройство-получатель 14. В одном примере считываемый компьютером носитель 16 может содержать среду связи, чтобы позволить устройству-источнику 12 передавать закодированные видеоданные непосредственно в устройство-получатель 14 в режиме реального времени. Закодированные видеоданные могут модулироваться в соответствии со стандартом связи, таким как протокол беспроводной связи, и передаваться на устройство-получатель 14. Среда связи может содержать любую беспроводную или проводную среду связи, такую как радиочастотный (RF) спектр или одна или несколько физических линий передачи. Среда связи может быть частью сети на основе пакетов, такой как локальная сеть, сеть широкого охвата или глобальная сеть, такая как Интернет. Среда связи может включать в себя маршрутизаторы, коммутаторы, базовые станции или любое другое оборудование, которое может быть полезно для облегчения связи от устройства-источника 12 к устройству-получателю 14.

В некоторых примерах кодированные данные могут выводиться из интерфейса 22 вывода в устройство хранения.

Аналогичным образом кодированные данные могут быть доступны из устройства хранения через интерфейс ввода. Устройство хранения может включать в себя любое из множества распределенных или локально доступных носителей хранения данных, таких как накопитель на жестком диске, Blu-ray диски, DVD, CD-ROM, флэш-память, энергозависимая или энергонезависимая память или любые другие подходящие цифровые носители хранения для хранения закодированных видеоданных. В другом примере, устройство хранения может соответствовать файловому серверу или другому промежуточному устройству хранения, которое может хранить закодированное видео, сгенерированное устройством-источником 12. Устройство-получатель 14 может осуществлять доступ к сохраненным видеоданным из устройства хранения посредством потоковой передачи или загрузки. Файловым сервером может быть любой тип сервера, способный хранить закодированные видеоданные и передавать эти закодированные видеоданные на устройство-получатель 14. Примеры файловых серверов включают в себя веб-сервер (например, для веб-сайта), FTP-сервер, сетевые устройства хранения (NAS) или локальный дисковый накопитель. Устройство-получатель 14 может осуществлять доступ к закодированным видеоданным через любое стандартное соединение для передачи данных, включая соединение с Интернетом. Это может включать в себя беспроводной канал (например, соединение Wi-Fi), проводное соединение (например, DSL, кабельный модем и т.д.) или комбинацию обоих, которая подходит для доступа к закодированным видеоданным, хранящимся на файловом сервере. Передача закодированных видеоданных из устройства хранения может быть потоковой передачей, передачей загрузки или их комбинацией.

Методы настоящего раскрытия не являются обязательно ограниченными беспроводными приложениями или настройками. Эти методы могут применяться к кодированию видео в поддержку любого из множества мультимедийных приложений, таких как эфирные телевизионные трансляции, передачи кабельного телевидения, передачи спутникового телевидения, потоковые передачи видео по Интернету, такие как динамическая адаптивная потоковая передача по HTTP (DASH), цифровое видео, которое кодируется на носителе хранения данных, декодирование цифрового видео, хранящегося на носителе данных, или другие приложения. В некоторых примерах система 10 может быть сконфигурирована для поддержки односторонней или двусторонней передачи видео, для поддержки таких приложений, как потоковое видео, воспроизведение видео, видеовещание и/или видеотелефония.

В примере согласно фиг. 1 устройство-источник 12 включает в себя источник 18 видео, видеокодер 20 и интерфейс 22 вывода. Устройство-получатель 14 включает в себя интерфейс 28 ввода, видеодекодер 30 и устройство 32 отображения. В соответствии с этим раскрытием видеокодер 20 устройства-источника 12 может быть сконфигурирован для применения методов для расширенного предсказания временного вектора движения (ATMVP). В других примерах устройство-источник и устройство-получатель могут включать в себя другие компоненты или устройства. Например, устройство-источник 12 может принимать видеоданные от внешнего источника 18 видео, такого как внешняя камера. Аналогичным образом устройство-получатель 14 может взаимодействовать с внешним устройством отображения, а не включать в себя встроенное устройство отображения.

Проиллюстрированная система 10 согласно фиг. 1 является всего лишь одним примером. Методы расширенного предсказания временного вектора движения (ATMVP) могут выполняться любым устройством кодирования и/или декодирования цифрового видео. Хотя, как правило, методы этого раскрытия выполняются устройством кодирования видео, методы также могут выполняться кодером/декодером видео, обычно называемым "кодеком". Кроме того, методы этого раскрытия могут также выполняться видеопроцессором. Устройство-источник 12 и устройство-получатель 14 являются просто примерами таких устройств кодирования, в которых устройство-источник 12 генерирует кодированные видеоданные для передачи в устройство-получатель 14. В некоторых примерах устройств 12, 14 могут функционировать, по существу, симметрично, так что каждое из устройств 12, 14 включает в себя компоненты кодирования и декодирования видео. Следовательно, система 10 может поддерживать одностороннюю или двустороннюю передачу видео между видеоустройствами 12, 14, например, для потоковой передачи ви-

део, воспроизведения видео, видеовещания или видеотелефонии.

Источник 18 видео устройства-источника 12 может включать в себя устройство съемки видео, такое как видеокамера, видеоархив, содержащий ранее снятое видео, и/или интерфейс ввода видео для приема видео от поставщика видеоконтента. В качестве дополнительной альтернативы источник 18 видео может генерировать данные на основе компьютерной графики в качестве исходного видео или комбинацию видео в реальном времени, архивированного видео и видео, генерируемого компьютером. В некоторых случаях, если источник 18 видео представляет собой видеокамеру, устройство-источник 12 и устройство-получатель 14 могут образовывать так называемые камерофоны или видеофоны. Как упомянуто выше, однако методы, описанные в настоящем раскрытии, могут быть применимы к кодированию видео в общем и могут применяться к беспроводным и/или проводным приложениям. В каждом случае снимаемое, предварительно снятое или генерируемое компьютером видео может быть закодировано видеокодером 20. Затем закодированная видеoinформация может выводиться интерфейсом 22 вывода на считываемый компьютером носитель 16.

Считываемый компьютером носитель 16 может включать в себя переходные (временные) среды, такие как беспроводное вещание или проводная сетевая передача, или носители хранения данных (то есть невременные носители данных), такие как жесткий диск, флэш-накопитель, компакт-диск, цифровой видеодиск, Blu-ray диск или другие считываемые компьютером носители. В некоторых примерах, сетевой сервер (не показан) может принимать закодированные видеоданные из устройства-источника 12 и предоставлять закодированные видеоданные устройству-получателю 14, например, посредством сетевой передачи. Аналогично вычислительное устройство производства по выпуску носителей, такого как оборудование штамповки дисков, может принимать данные закодированного видео из устройства-источника 12 и создавать диск, содержащий данные закодированного видео. Следовательно, считываемый компьютером носитель 16 можно понимать как включающий в себя одну или несколько считываемых компьютером сред различных форм в различных примерах.

Интерфейс 28 ввода устройства-получателя 14 принимает информацию от считываемого компьютером носителя 16. Информация считываемого компьютером носителя 16 может включать в себя синтаксическую информацию, определенную видеокодером 20, которая также используется видеодекодером 30, которая включает в себя синтаксические элементы, которые описывают характеристики и/или обработку блоков и других кодированных единиц, например GOP. Устройство 32 отображения отображает данные декодированного видео пользователю и может содержать любое из множества устройств отображения, таких как электронно-лучевая трубка (CRT), жидкокристаллический дисплей (LCD), плазменный дисплей, дисплей на органических светоизлучающих диодах (OLED) или другой тип устройства отображения.

Видеокодер 20 и видеодекодер 30 могут работать в соответствии со стандартом кодирования видео, таким как стандарт высокоэффективного кодирования видео (HEVC), расширения стандарта HEVC или последующие стандарты, такие как ITU-T H.266. В качестве альтернативы видеокодер 20 и видеодекодер 30 могут работать в соответствии с другими проприетарными или промышленными стандартами, такими как стандарт ITU-T H.264, альтернативно называемый MPEG-4, часть 10, расширенное кодирование видео (AVC), или расширения таких стандартов. Однако методы настоящего раскрытия не ограничиваются каким-либо конкретным стандартом кодирования. Другие примеры стандартов кодирования видео включают MPEG-2 и ITU-T H.263. Хотя это не показано на фиг. 1, в некоторых аспектах, видеокодер 20 и видеодекодер 30 могут быть объединены с аудиокодером и декодером и могут включать в себя соответствующие блоки MUX-DEMUX или другое аппаратное и программное обеспечение для обработки кодирования как аудио, так и видео в общем потоке данных или отдельных потоках данных. Если применимо, устройства MUX-DEMUX могут соответствовать протоколу мультимплексора ITU H.223 или другим протоколам, таким как протокол пользовательских дейтаграмм (UDP).

Видеокодер 20 и видеодекодер 30, каждый, могут быть реализованы как любая из множества подходящих схем кодера, таких как один или несколько микропроцессоров, процессоров цифровых сигналов (DSP), специализированных интегральных схем (ASIC), программируемых пользователем вентильных матриц (FPGA), дискретная логика, программное обеспечение, аппаратные средства, встроенное программное обеспечение или любые их комбинации. Когда методы частично реализованы в программном обеспечении, устройство может хранить инструкции для программного обеспечения на подходящем, невременном считываемом компьютером носителе и исполнять инструкции на аппаратных средствах с использованием одного или нескольких процессоров для выполнения методов настоящего раскрытия. Каждый из видеокодера 20 и видеодекодера 30 может быть включен в один или несколько кодеров или декодеров, каждый из которых может быть интегрирован как часть объединенного кодера/декодера (кодека) в соответствующем устройстве.

JCT-VC работает над разработкой стандарта HEVC. Усилия по стандартизации HEVC основаны на развивающейся модели устройства кодирования видео, которая называется тестовой моделью HEVC (HM). HM предполагает несколько дополнительных возможностей устройств кодирования видео по сравнению с существующими устройствами согласно, например, ITU-T H.264/AVC. Например, в то время как H.264 обеспечивает девять режимов кодирования с интра-предсказанием, HEVC HM может обес-

печивать до тридцати трех режимов кодирования с интра-предсказанием.

В общем, рабочая модель НМ описывает, что видеокадр или картинку можно разделить на последовательность древовидных блоков или наибольших единиц кодирования (LCU), которые включают в себя выборки как яркости, так и цветности. Синтаксические данные в потоке битов могут определять размер для LCU, которая является наибольшей единицей кодирования по количеству пикселей. Вырезка включает в себя ряд последовательных древовидных блоков в порядке кодирования. Видеокадр или картинка могут быть разделены на одну или несколько вырезок. Каждый древовидный блок может быть разделен на единицы кодирования (CU) в соответствии с квадродеревом. В общем, структура данных квадродерева включает в себя один узел на каждую CU, с корневым узлом, соответствующим древовидному блоку. Если CU разделена на четыре под-CU, узел, соответствующий CU, включает в себя четыре листовых узла, каждый из которых соответствует одной из под-CU.

Каждый узел структуры данных квадродерева может предоставлять синтаксические данные для соответствующей CU. Например, узел в квадродереве может включать в себя флаг разделения, указывающий, является ли CU, соответствующая узлу, разделенной на под-CU. Синтаксические элементы для CU могут быть определены рекурсивно и могут зависеть от того, разделена ли CU на под-CU. Если CU не разделяется дальше, то она упоминается как листовая CU. В настоящем раскрытии, четыре суб-CU листовой CU также будут упоминаться как листовые CU, даже если нет явного разделения исходной листовой CU. Например, если CU с размером 16×16 не будет разделяться дальше, четыре 8×8 суб-CU также будут называться листовыми CU, хотя 16×16 CU не была разделена.

CU имеет аналогичное назначение, что и у макроблока стандарта H.264, за исключением того, что CU не имеет отличия в размерах. Например, древовидный блок можно разделить на четыре дочерних узла (также называемых под-CU), и каждый дочерний узел, в свою очередь, может быть родительским узлом и может разбиваться на другие четыре дочерних узла. Конечный неразделенный дочерний узел, называемый листовым узлом квадродерева, содержит узел кодирования, также называемый листовой CU. Синтаксические данные, ассоциированные с кодированным битовым потоком, могут определять максимальное количество раз разделения древовидного блока, называемое максимальной глубиной CU, и также могут определять минимальный размер узлов кодирования. Соответственно, битовый поток также может определять наименьшую единицу кодирования (SCU). Это раскрытие использует термин "блок" для ссылки на любую из CU, PU или TU в контексте HEVC или аналогичных структур данных в контексте других стандартов (например, макроблоков и их подблоков в H.264/AVC).

CU включает в себя узел кодирования и единицы предсказания (PU) и единицы преобразования (TU), ассоциированные с узлом кодирования. Размер CU соответствует размеру узла кодирования и должен иметь квадратную форму. Размер CU может варьироваться от 8×8 пикселей до размера древовидного блока максимум 64×64 пикселей или больше. Каждая CU может содержать одну или несколько PU и одну или несколько TU. Синтаксические данные, ассоциированные с CU, могут описывать, например, разбиение CU на одну или несколько PU. Режимы разбиения могут различаться между тем, пропущена ли CU или закодирована в прямом режиме, закодирована в режиме интра-предсказания или закодирована в режиме интер-предсказания. PU могут быть разбиты на неквадратные формы. Синтаксические данные, ассоциированные с CU, могут также описывать, например, разбиение CU на одну или несколько TU в соответствии с квадродеревом. TU может быть квадратной или не квадратной (например, прямоугольной) по форме.

Стандарт HEVC допускает преобразования в соответствии с TU, которые могут отличаться для разных CU. TU обычно имеют размер, основанный на размере PU в данной CU, определенной для разделенной LCU, хотя это может быть не всегда так. TU обычно имеют тот же размер или меньше, чем PU. В некоторых примерах остаточные выборки, соответствующие CU, могут быть подразделены на более мелкие единицы с использованием структуры квадродерева, известной как "остаточное квадродерево" (RQT). Листовые узлы RQT могут упоминаться как единицы преобразования (TU). Значения пиксельной разности, ассоциированные с TU, могут быть преобразованы для получения коэффициентов преобразования, которые могут быть квантованы.

Листовая CU может включать в себя одну или несколько единиц предсказания (PU). В общем, PU представляет собой пространственную область, соответствующую всей или части соответствующей CU, и может включать в себя данные для извлечения опорной выборки для PU. Кроме того, PU включает данные, относящиеся к предсказанию. Например, когда PU закодирована в интра-режиме, данные для PU могут быть включены в остаточное квадродерево (RQT), что может включать в себя данные, описывающие режим интра-предсказания для TU, соответствующей PU. В качестве еще одного примера, когда PU кодируется в интер-режиме, PU может включать в себя данные, определяющие один или несколько векторов движения для PU. Данные, определяющие вектор движения для PU, могут описывать, например, горизонтальный компонент вектора движения, вертикальный компонент вектора движения, разрешение для вектора движения (например, точность в одну четвертую пиксела или одну восьмую пиксела), опорную картинку, на которую указывает вектор движения, и/или список опорных картинок (например, список 0, список 1 или список C) для вектора движения.

Листовая CU, имеющая одну или несколько PU, может также включать в себя одну или несколько единиц преобразования (TU). Единицы преобразования могут быть указаны с использованием RQT (также называемого структурой квадродерева TU), как обсуждалось выше. Например, флаг разбиения может указывать, разделена ли листовая CU на четыре единицы преобразования. Затем каждая единица преобразования может быть далее разделена на дальнейшие под-TU. Когда TU далее не разделяется, она может называться листовой TU. Как правило, для интра-кодирования, все листовые TU, принадлежащие листовой CU, совместно используют один и тот же режим интра-предсказания. То есть, один и тот же режим интра-предсказания обычно применяется для вычисления предсказанных значений для всех TU листовой CU. Для интра-кодирования видеокодер может вычислять остаточное значение для каждой листовой TU, используя режим интра-предсказания, в качестве разности между частью CU, соответствующей TU, и исходным блоком. TU не обязательно ограничивается размером PU. Таким образом, TU могут быть больше или меньше, чем PU. Для интра-кодирования PU может располагаться рядом с соответствующей листовой TU для той же самой CU. В некоторых примерах, максимальный размер листовой TU может соответствовать размеру соответствующей листовой CU.

Кроме того, TU листовых CU могут также быть ассоциированы с соответствующими структурами данных квадродерева, упоминаемыми как остаточные квадродеревья (RQT). То есть листовая CU может включать в себя квадродерево, указывающее, как листовая CU разбивается на TU. Корневой узел квадродерева TU обычно соответствует листовой CU, а корневой узел квадродерева CU обычно соответствует древовидному блоку (или LCU). TU RQT, которые не разделены, называются листовыми TU. В общем, настоящее раскрытие использует термины CU и TU для ссылки на листовую CU и листовую TU, соответственно, если не указано иное.

Последовательность видео обычно включает в себя серию видеок кадров или картинок. Группа картинок (GOP) обычно содержит серию из одной или нескольких видео картинок. GOP может включать синтаксические данные в заголовке GOP, заголовке одной или нескольких картинок или в другом месте, которые описывают ряд картинок, включенных в GOP. Каждая вырезка картинки может включать в себя синтаксические данные вырезки, которые описывают режим кодирования для соответствующей вырезки. Видеокодер 20 обычно работает с видеоблоками в отдельных видео вырезках, чтобы кодировать видеоданные. Видеоблок может соответствовать узлу кодирования в CU. Видеоблоки могут иметь фиксированные или различные размеры и могут отличаться по размеру в соответствии с указанным стандартом кодирования.

В качестве примера, HM поддерживает предсказание в различных размерах PU. Предполагая, что размер конкретной CU равен  $2N \times 2N$ , HM поддерживает интра-предсказание в размерах PU  $2N \times 2N$  или  $N \times N$  и интер-предсказание в симметричных размерах PU  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$  или  $N \times N$ . HM также поддерживает асимметричное разбиение для интер-предсказания в размерах PU  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  и  $nR \times 2N$ . В асимметричном разбиении одно направление CU не разделяется, а другое направление разделяется на 25 и 75%. Часть CU, соответствующая разделу 25%, обозначается посредством "п", за которым следует указание "Вверх", "Вниз", "Влево" или "Вправо". Так, например, " $2N \times nU$ " относится к  $2N \times 2N$  CU, которая разделена горизонтально с  $2N \times 0,5N$  PU сверху и  $2N \times 1,5N$  PU снизу.

В настоящем раскрытии " $N \times N$ " и " $N$  на  $N$ " могут использоваться взаимозаменяемо, чтобы ссылаться на пиксельные размеры видеоблока в терминах вертикальных и горизонтальных размеров, например  $16 \times 16$  пикселей или 16 на 16 пикселей. В общем случае, блок  $16 \times 16$  будет иметь 16 пикселей в вертикальном направлении ( $y=16$ ) и 16 пикселей в горизонтальном направлении ( $x=16$ ). Аналогично, блок  $N \times N$  обычно имеет  $N$  пикселей в вертикальном направлении и  $N$  пикселей в горизонтальном направлении, где  $N$  представляет неотрицательное целочисленное значение. Пиксели в блоке могут быть расположены в строках и столбцах. Кроме того, блоки не обязательно должны иметь одинаковое количество пикселей в горизонтальном направлении и в вертикальном направлении. Например, блоки могут содержать  $N \times M$  пикселей, где  $M$  не обязательно равно  $N$ .

Вслед за кодированием с интра-прогнозированием или интер-предсказанием с использованием PU в CU видеокодер 20 может вычислять остаточные данные для TU в CU. PU могут содержать синтаксические данные, описывающие способ или режим генерации предсказанных пиксельных данных в пространственной области (также называемой пиксельной областью), и TU могут содержать коэффициенты в области преобразования после применения преобразования, например дискретного косинусного преобразования (DCT), целочисленного преобразования, вейвлет-преобразования или концептуально подобного преобразования в остаточные видеоданные. Остаточные данные могут соответствовать пиксельным разностям между пикселями не закодированной картинки и значениями предсказания, соответствующими PU. Видеокодер 20 может формировать TU, включая остаточные данные для CU, а затем преобразовывать TU для получения коэффициентов преобразования для CU.

После любых преобразований для получения коэффициентов преобразования видеокодер 20 может выполнять квантование коэффициентов преобразования. Квантование обычно относится к процессу, в котором коэффициенты преобразования квантуются, чтобы, возможно, уменьшить количество данных, используемых для представления коэффициентов, обеспечивая дополнительное сжатие. Процесс кванто-

вания может уменьшить битовую глубину, ассоциированную с некоторыми или всеми коэффициентами. Например,  $n$ -битовое значение может быть округлено до  $m$ -битового значения во время квантования, где  $n$  больше, чем  $m$ .

После квантования видеокодер может сканировать коэффициенты преобразования, создавая одномерный вектор из двумерной матрицы, включающей в себя квантованные коэффициенты преобразования. Сканирование может быть создано так, чтобы размещать более высокоэнергетичные (и, следовательно, более низкочастотные) коэффициенты в передней части массива и размещать более низкоэнергетичные (и, следовательно, более высокочастотные) коэффициенты в задней части массива. В некоторых примерах видеокодер 20 может использовать predetermined порядок сканирования для сканирования квантованных коэффициентов преобразования для создания преобразованного в последовательную форму вектора, который может энтропийно кодироваться. В других примерах видеокодер 20 может выполнять адаптивное сканирование. После сканирования квантованных коэффициентов преобразования для формирования одномерного вектора видеокодер 20 может энтропийно кодировать одномерный вектор, например, согласно контекстно-адаптивному кодированию с переменной длиной (CAVLC), контекстно-адаптивному двоичному арифметическому кодированию (CABAC), контекстно-адаптивному двоичному арифметическому кодированию на основе синтаксиса (SBAC), энтропийному кодированию с разбиением на интервалы вероятности (PIPE) или другой методологии энтропийного кодирования. Видеокодер 20 может также энтропийно кодировать синтаксические элементы, ассоциированные с закодированными видеоданными для использования видеодекодером 30 при декодировании видеоданных.

Для выполнения CABAC видеокодер 20 может назначать контекст в рамках контекстной модели символу, который должен быть передан. Контекст может относиться, например, к тому, являются ли соседние значения символа отличными от нуля или нет.

Для выполнения CAVLC видеокодер 20 может выбирать код переменной длины для передаваемого символа. Кодовые слова в VLC могут быть сконструированы таким образом, что относительно короткие коды соответствуют более вероятным символам, тогда как более длинные коды соответствуют менее вероятным символам. Таким образом, использование VLC может обеспечить экономию битов по сравнению, например, с использованием кодовых слов равной длины для каждого передаваемого символа. Определение вероятности может быть основано на контексте, назначенном символу.

В соответствии с методами настоящего раскрытия видеокодер 20 и видеодекодер 30 могут быть сконфигурированы для выполнения любых или всех из следующих методов, показанных в нумерованном списке ниже, отдельно или в любой комбинации.

1. Помещение кандидата ATMVP, если он вставлен, например, в список кандидатов слияния
  - a. Предполагается, что противопоставленные кандидаты и кандидат TMVP вставляются в список кандидатов слияния в определенном порядке. Кандидат ATMVP может быть вставлен в любую относительно фиксированную позицию этих кандидатов:
    - i) в одном варианте, например, кандидат ATMVP может быть вставлен в список кандидатов слияния после первых двух пространственных кандидатов, например A1 и B1;
    - ii) в одном варианте, например, кандидат ATMVP может быть вставлен после первых трех пространственных кандидатов, например A1 и B1 и B0;
    - iii) в одном варианте, например, кандидат ATMVP может быть вставлен после первых четырех кандидатов, например, A1, B1, B0 и A0;
    - iv) в одном варианте, например, кандидат ATMVP может быть вставлен непосредственно перед кандидатом TMVP;
    - v) в одном варианте, например, кандидат ATMVP может быть вставлен сразу после кандидата TMVP.
  - b. В качестве альтернативы, позиция кандидата ATMVP в списке кандидатов может сигнализироваться в потоке битов. Позиции других кандидатов, включая кандидата TMVP, могут дополнительно сигнализироваться.

2. Проверка доступности кандидата ATMVP может применяться путем доступа к одному набору информации о движении. Когда такой набор информации недоступен, например один блок является интра-кодированным, весь кандидат ATMVP считается недоступным. В этом случае ATMVP не будет вставлен в список слияния.

a. Центральная позиция или центральная под-PU используется исключительно для проверки доступности кандидата ATMVP. Когда используется центральная под-PU, центральная под-PU выбирается так, чтобы она покрывала центральную позицию (например, позицию центра 3 с относительной координатой  $(W/2, H/2)$  к верхней левой выборке PU, где  $W \times H$  является размером PU). Такая позиция или центральная под-PU может использоваться вместе с временным вектором для идентификации соответствующего блока в исходной картинке движения. Определяется набор информации движения из блока, который покрывает центральную позицию соответствующего блока.

3. Репрезентативный набор информации о движении для кодированной PU ATMVP из под-PU.

a. Чтобы сформировать кандидата ATMVP, сначала формируется репрезентативный набор информации о движении.

b. Такой репрезентативный набор информации о движении может быть получен из фиксированной позиции или фиксированной под-PU. Он может быть выбран таким же образом, как и набор информации о движении, используемый для определения доступности кандидата ATMVP, как описано в пункте 2.

c. Когда под-PU идентифицировала свой собственный набор информации о движении и он недоступен, он устанавливается равным репрезентативному набору информации о движении.

d. Если репрезентативный набор информации о движении установлен как таковой для под-PU, то на стороне декодера не требуется дополнительного хранения движения для текущей STU или вырезки в сценарии наихудшего случая.

e. Такой репрезентативный набор информации о движении используется во всех сценариях, когда процессы декодирования требуют, чтобы вся PU была представлена одним набором информации о движении, включая обрезку, так что этот процесс используется для генерирования комбинированных двуправленно предсказанных кандидатов слияния.

4. Кандидат ATMVP обрезается кандидатом TMVP, и могут учитываться взаимодействия между TMVP и ATMVP; подробные методы перечислены ниже.

a. Обрезка кандидата на основе под-PU, например кандидата ATMVP нормальным кандидатом, может быть выполнена с использованием репрезентативного набора информации о движении (как в пункте 3) для такого кандидата на основе под-PU. Если такой набор информации о движении совпадает с обычным кандидатом слияния, два кандидата считаются одинаковыми.

b. В качестве альтернативы, кроме того, выполняется проверка, чтобы определить, содержит ли ATMVP несколько разных наборов информации о движении для нескольких под-PU; если идентифицировано по меньшей мере два разных набора, кандидат на основе под-PU не используется для обрезки, то есть считается отличным от любого другого кандидата; в противном случае его можно использовать для обрезки (например, можно обрезать в процессе обрезки).

c. В качестве альтернативы, кроме того, кандидат ATMVP может быть обрезан пространственными кандидатами, например только левым и верхним, с позициями, обозначенными как A1 и B1.

d. В качестве альтернативы, только один кандидат формируется из временной опоры, являясь кандидатом ATMVP или кандидатом TMVP. Когда ATMVP доступно, кандидатом является ATMVP; в противном случае кандидатом является TMVP. Такой кандидат вставляется в список кандидатов слияния в позиции, аналогичной позиции TMVP. В этом случае максимальное количество кандидатов может оставаться неизменным:

i) в качестве альтернативы, TMVP всегда отключается, даже если ATMVP недоступно;

ii) в качестве альтернативы, TMVP используется только тогда, когда ATMVP недоступно.

e. В качестве альтернативы, когда ATMVP доступно и TMVP недоступно, в качестве кандидата TMVP используется один набор информации о движении одной под-PU. В этом случае, кроме того, процесс обрезки между ATMVP и TMVP не применяется.

f. Альтернативно или дополнительно, временной вектор, используемый для ATMVP, может также использоваться для TMVP, так что нижнюю правую позицию или позицию центра 3, как используется для текущего TMVP в HEVC, не требуется использовать

i) в качестве альтернативы, позиция, идентифицированная временным вектором, и нижняя правая позиция и позиция центра 3 совместно рассматриваются для обеспечения доступного кандидата TMVP.

5. Множество проверок доступности для ATMVP поддерживается, чтобы повысить шансы для кандидата ATMVP быть более точным и эффективным. Когда текущий кандидат ATMVP из исходной картинки движения, как идентифицировано первым временным вектором (например, как показано на фиг. 9), недоступен, другие картинки могут рассматриваться в качестве исходной картинки движения. Когда рассматривается другая картинка, она может быть ассоциирована с другим вторым временным вектором или может быть ассоциирована просто со вторым временным вектором, масштабированным из первого временного вектора, который указывает на недоступного кандидата ATMVP.

a. Второй временной вектор может идентифицировать кандидата ATMVP во второй исходной картинке движения, и может применяться такая же проверка доступности. Если кандидат ATMVP, полученный из второй исходной картинки движения, доступен, кандидат ATMVP выводится, и не требуется проверять никакие другие картинки; в противном случае необходимо проверить другие картинки в качестве исходных картинок движения.

b. Картинки, которые необходимо проверить, могут быть указаны в списках опорных картинок текущей картинки в заданном порядке. Для каждого списка, картинки проверяются в порядке возрастания опорного индекса. Сначала проверяется список X, а затем следуют картинки в списке Y (являющемся 1-X)

i) список X выбирается так, что список X является списком, который содержит рядом расположенную картинку, используемую для TMVP;

ii) альтернативно, X просто устанавливается равным 1 или 0.

c. Картинки, которые нужно проверить, являются теми, которые идентифицированы векторами движения пространственных соседей с заданным порядком.

6. Раздел PU, к которому применяется текущее ATMVP, может быть разделами  $2N \times 2N$ ,  $N \times N$ ,  $2N \times N$ ,

$N \times 2N$  или асимметричным разделом движения (AMP), таким как  $2N \times N/2$ .

a. В качестве альтернативы, кроме того, если разрешены другие размеры разделов, ATMVP также может поддерживаться, и такой размер может включать, например,  $64 \times 8$ .

b. Альтернативно, режим может применяться только к определенным разделам, например,  $2N \times 2N$ .

7. Кандидат ATMVP маркируется как другой тип кандидата слияния.

8. При идентификации вектора (временного вектора, как на первом этапе) из соседей несколько соседних позиций, например те, которые используются при построении списка кандидатов слияния, могут проверяться по порядку. Для каждого из соседей векторы движения, соответствующие списку опорных картинок 0 (списку 0) или списку опорных картинок 1 (списку 1), могут проверяться по порядку. Когда доступны два вектора движения, сначала могут проверяться векторы движения в списке X, а затем в списке Y (при Y, равном  $1-X$ ), так что список X является списком, который содержит рядом расположенную картинку, используемую для TMVP. В ATMVP используется временной вектор как сдвиг любой центральной позиции под-PU, причем компоненты временного вектора могут требовать смещения к целым числам. Такая смещенная центральная позиция используется для идентификации наименьшей единицы, которой может быть выделен вектор движения, например с размером  $4 \times 4$ , которая покрывает текущую центральную позицию.

a. Альтернативно, векторы движения, соответствующие списку 0, могут быть проверены перед теми, которые соответствуют списку 1.

b. Альтернативно, векторы движения, соответствующие списку 1, могут быть проверены перед теми, которые соответствуют списку 0.

c. Альтернативно, все векторы движения, соответствующие списку X во всех пространственных соседях, проверяются по порядку, за которым следуют векторы движения, соответствующие списку Y (при Y, равном  $1-X$ ). Здесь список "X" может быть списком, который указывает, где находится расположенная рядом картинка, или просто устанавливается как 0 или 1.

d. Порядок пространственных соседей может быть таким же, как используется в режиме слияния HEVC.

9. Когда на первом этапе идентификации временного вектора не включается информация, идентифицирующая опорную картинку, исходная картинка движения, как показано на фиг. 9, может просто устанавливаться на фиксированную картинку, например рядом расположенную картинку, используемую для TMVP.

a. В таком случае вектор можно идентифицировать только из векторов движения, которые указывают на такую фиксированную картинку.

b. В таком случае, вектор можно идентифицировать только из векторов движения, которые указывают на любую картинку, но дополнительно масштабированы к фиксированной картинке.

10. Когда на первом этапе идентификации вектора включается идентификация опорной картинки, исходной картинки движения, как показано на фиг. 9, для вектора движения кандидата может применяться одна или несколько из следующих дополнительных проверок.

a. Если вектор движения ассоциирован с картинкой или вырезкой, которая является интра-кодированной, такой вектор движения считается недоступным и не может быть использован для преобразования в вектор.

b. Если вектор движения идентифицирует интра-блок (например, путем суммирования текущей координаты центра с вектором движения) в ассоциированной картинке, такой вектор движения считается недоступным и не может быть использован для преобразования в вектор.

11. Когда на первом этапе идентификации вектора компоненты вектора могут быть установлены как (половина ширины текущей PU, половина высоты текущей PU), так что это идентифицирует позицию нижнего правого пиксела в исходной картинке движения. Здесь (x, y) указывает горизонтальный и вертикальный компоненты одного вектора движения.

a. В качестве альтернативы, компоненты вектора могут быть установлены как (сумма (половины ширины текущей PU, M), сумма (половины высоты текущей PU, N)), где функция сумма (a, b) возвращает сумму a и b. В одном примере, когда информация о движении сохранена в блоке  $4 \times 4$ , M и N, оба, установлены равными 2. В другом примере, когда информация о движении сохранена в блоке  $8 \times 8$ , M и N, оба, установлены равными 4.

12. Размер подблока/под-PU при применении ATMVP сигнализируется в наборе параметров, например наборе параметров последовательности набора параметров картинки. Размер варьируется от наименьшего размера PU до размера CTU. Размер также может быть предварительно определен или сигнализирован.

Размер может быть, например, размером  $4 \times 4$ . В качестве альтернативы, размер подблока/под-PU может быть получен на основе размера PU или CU. Например, подблок/под-PU может быть установлен равным  $\max(4 \times 4, (\text{ширина CU}) \gg M)$ . Значение M может быть предопределено или сигнализировано в битовом потоке.

13. Максимальное количество кандидатов слияния может быть увеличено на 1 ввиду того, что

ATMVP может рассматриваться как новый кандидат слияния. Например, по сравнению с HEVC, которое после обрезки принимает до 5 кандидатов в списке кандидатов слияния, максимальное количество кандидатов слияния может быть увеличено до 6.

a. В качестве альтернативы, для ATMVP может быть выполнена обрезка обычным кандидатом TMVP или унификация с обычным кандидатом TMVP, так что максимальное количество кандидатов слияния может сохраняться без изменений.

b. В качестве альтернативы, когда ATMVP идентифицировано как доступное, пространственный соседний кандидат исключается из списка кандидатов слияния, например исключается последний пространственный кандидат в порядке выборки.

14. Когда учитывается несколько пространственных соседних векторов движения для получения временного вектора, сходство вектора движения можно рассчитать на основе соседних векторов движения текущей PU, а также соседних векторов движения, идентифицированных конкретным временным вектором, установленным равным вектору движения. Тот, который приводит к наивысшему сходству движения, может быть выбран в качестве конечного временного вектора.

a. В одном варианте для каждого вектора движения из соседней позиции N вектор движения идентифицирует блок (такого же размера, что и текущая PU) в исходной картинке движения, причем его соседняя позиция N содержит набор информации о движении. Этот набор вектора движения сравнивается с набором информации о движении, как в соседней позиции N текущего блока.

b. В другом варианте для каждого вектора движения из соседней позиции N вектор движения идентифицирует блок в исходной картинке движения, причем его соседние позиции содержат множество наборов информации о движении. Эти множество наборов вектора движения сравниваются с множеством наборов информации о движении из соседних позиций текущей PU в тех же относительных позициях. Вычисляется подобие информации о движении. Например, текущая PU имеет следующие наборы информации о движении из A1, B1, A0 и B0, обозначенные как MIA1, MIB1, MIA0 и MIB0. Для временного вектора TV это идентифицирует блок, соответствующий PU в исходной картинке движения. Такой блок имеет информацию о движении из тех же относительных позиций A1, B1, A0 и B0 и обозначается как TMIA1, TMIB1, TMIA0 и TMIB0. Подобие движения, как определено посредством TV, вычисляется как  $MStv = \sum_{N \in \{A1, B1, A0, B0\}} [(MVSim([(MI)]_N), [(TMI)]_N)]$ , где MVSим определяет подобие между двумя наборами информации о движении.

c. В обоих из вышеуказанных случаев может использоваться подобие движения MVSим, причем два входных параметра представляют собой два набора информации о движении, каждый из которых содержит до двух векторов движения и двух опорных индексов. Каждая пара векторов движения в списке X фактически ассоциирована с опорными картинками в другом списке X других картинок, текущей картинки и исходной картинки движения. Для каждого из двух векторов движения MVXN и TMVXN (при X, равно 0 или 1) разность векторов движения MVDXN может быть вычислена как MVXN-TMVXN. Затем вычисляется разность MVSимX, например, как  $abs([(MVDX)]_N[0]) + abs([(MVDX)]_N[1])$  или  $([(MVDX)]_N[0]) * [(MVDX)]_N[0] + [(MVDX)]_N[1] * [(MVDX)]_N[1]$ . Если оба набора информации о движении содержат доступные векторы движения, MVSим устанавливается равным  $MVSим0 + MVSим1$

i) чтобы иметь единообразное вычисление разности движений, оба вектора движения необходимо масштабировать к одной и той же фиксированной картинке, которая может быть, например, первой опорной картинкой RefPicListX [0] списка X текущей картинки;

ii) если доступность вектора движения в списке X из первого набора и доступность вектора движения в списке X из второго набора различны, т.е. один опорный индекс равен -1, а другой нет, то такие два набора информации о движении считаются не подобными в направлении X. Если два набора не одинаковы в обоих наборах, конечная функция MVSим может возвращать большое значение T, которое может, например, считаться бесконечным;

iii) в качестве альтернативы, для пары наборов информации о движении, если один предсказан из списка X (X равно 0 или 1), но не списка Y (Y равно 1-X), а другой имеет тот же статус, может быть использовано взвешивание между 1 и 2 (например, MVSим равно  $MVSимX * 1,5$ ). Когда один набор предсказан только из списка X, а другой предсказан только из списка Y, MVSим устанавливается на большее значение T;

iv) альтернативно, для любого набора информации о движении, если доступен один вектор движения, будут созданы оба вектора движения. В случае когда доступен только один вектор движения (соответствующий списку X), он масштабируется, чтобы сформировать вектор движения, соответствующий другому списку Y.

d. Альтернативно, вектор движения может быть измерен на основе разностей между соседними пикселями текущей PU и соседними пикселями блока (такого же размера, что и текущая PU), идентифицированного вектором движения. Вектор движения, который приводит к наименьшей разности, может быть выбран в качестве конечного временного вектора.

15. При выводе временного вектора текущего блока векторы движения и/или временные векторы из соседних блоков, которые кодированы с помощью ATMVP, могут иметь более высокий приоритет, чем векторы движения из других соседних блоков.

a. В одном примере сначала проверяются только временные векторы соседних блоков, и первый доступный может быть установлен на временной вектор текущего блока. Только когда таких временных векторов нет, далее проверяются нормальные векторы движения. В этом случае необходимо сохранить временные векторы для кодированных блоков ATMVP.

b. В другом примере сначала проверяются только векторы движения из кодированных соседних блоков ATMVP, и первый доступный может быть установлен на временной вектор текущего блока. Только когда таких временных векторов нет, далее проверяются нормальные векторы движения.

c. В другом примере сначала проверяются только векторы движения из кодированных соседних блоков ATMVP, и первый доступный может быть установлен на временной вектор текущего блока. Если такие векторы движения недоступны, проверка временного вектора продолжается аналогично тому, как в пункте 15a.

d. В другом примере сначала проверяются временные векторы из соседних блоков, первый доступный может быть установлен на временной вектор текущего блока. Если такие векторы движения недоступны, проверка временного вектора продолжается аналогично тому, как в пункте 15b.

e. В другом примере сначала проверяются временные векторы и векторы движения кодированных соседних блоков ATMVP, первый доступный может быть установлен на временной вектор текущего блока. Только когда такие временные векторы и векторы движения отсутствуют, далее проверяются нормальные векторы движения.

16. Когда учитывается несколько пространственных соседних векторов движения, чтобы получить временной вектор, вектор движения может быть выбран так, чтобы он минимизировал искажение, которое вычисляется из пиксельной области, например. сопоставление шаблонов может использоваться для получения временного вектора, так что тот, который приводит к минимальной стоимости сопоставления, выбирается в качестве конечного временного вектора.

17. Вывод набора информации о движении из соответствующего блока (в исходной картинке движения) выполняется таким образом, что когда вектор движения доступен в соответствующем блоке для любого списка X (обозначим вектор движения MVX) для текущей под-PU кандидата ATMVP, вектор движения считается доступным для списка X (путем масштабирования MVX). Если вектор движения недоступен в соответствующем блоке для любого списка X, вектор движения считается недоступным для списка X.

a. Альтернативно, когда вектор движения в соответствующем блоке недоступен для списка X, но доступен для списка 1-X (обозначим 1-X через Y и обозначим вектор движения как MVY), вектор движения по-прежнему считается доступным для списка X (путем масштабирования MVY к целевой опорной картинке в списке X).

b. Альтернативно или дополнительно, когда оба вектора движения в соответствующем блоке для списка X и списка Y (равного 1-X) доступны, векторы движения из списка X и списка Y не обязательно использовать, чтобы непосредственно масштабировать и генерировать два вектора движения текущей под-PU путем масштабирования

i) в одном примере при формулировании кандидата ATMVP проверка с низкой задержкой, как это делается в TMVP, применяется к каждой под-PU. Если для каждой картинке (обозначенной refPic) в каждом списке опорных картинок текущей вырезки значение порядкового отсчета картинке (POC) refPic меньше, чем POC текущей вырезки, то текущая вырезка рассматривается в режиме низкой задержки. В этом режиме низкой задержки векторы движения из списка X и списка Y масштабируются для генерации векторов движения текущей под-PU для списка X и списка Y, соответственно. Когда не в режиме низкой задержки, только один вектор движения MVZ из MVX или MVY выбирается и масштабируется для генерации двух векторов движения для текущей под-PU. Подобно TMVP, в таком случае Z устанавливается равным `collocated_from_10_flag`, что означает, что он зависит от того, находится ли расположенная рядом картинка как в TMVP в списке X или списке Y текущей картинке. Альтернативно, Z устанавливается следующим образом: если исходная картинка движения идентифицирована из списка X, Z устанавливается в X. Альтернативно, кроме того, когда исходные картинки движения относятся к обоим спискам опорных картинок, и `RefPicList0[id×0]` является исходной картинкой движения, которая первой присутствует в списке 0, и `RefPicList(1) [id×1]` является исходной картинкой движения, которая первой присутствует в списке 1, Z устанавливается равным 0, если `id×0` меньше или равно `id×1`, и устанавливается равным 1 в противном случае.

18. Исходная картинка движения может сигнализироваться, например, генерироваться видеокodeром 20 в кодированном битовом потоке. Более подробно, флаг, указывающий, является ли исходная картинка движения картинкой из списка 0 или списка 1, сигнализируется для B-вырезки. Альтернативно, кроме того, опорный индекс в список 0 или список 1 текущей картинки может сигнализироваться для идентификации исходной картинки движения.

19. При идентификации временного вектора вектор считается недоступным (таким образом, могут рассматриваться другие), если он указывает на интра-кодированный блок в ассоциированной исходной картинке движения.

Реализация различных методов настоящего раскрытия обсуждается ниже. Предполагается, что ATMVP реализовано поверх HEVC версии 1. Сжатие движения может не применяться к опорным картинкам, и могут быть включены более мелкие блоки с двунаправленной компенсацией движения.

Сигнализация ATMVP в SPS:

atmvp\_sub\_pu\_size может присутствовать в SPS;

atmvp\_sub\_pu\_size может указывать размер под-PU для PU, закодированной в режиме ATMVP. Он находится в диапазоне от 2 до 6 включительно. Размер под-PU для ATMVP (spuWidth, spuHeight) выводится как  $\min(w, 1 \ll \text{atmvp\_sub\_pu\_size})$  на  $\min(h, 1 \ll \text{atmvp\_sub\_pu\_size})$ , где wxh - размер текущей PU.

В качестве альтернативы, как ширина, так и высота размеров под-PU сигнализируются отдельно в SPS.

В качестве альтернативы, размеры под-PU сигнализируются в отличие от размера CTU или наименьшего размера блока кодирования.

Переменная atmvpEnableFlag выводится равной 1, если atmvp\_sub\_pu\_size меньше, чем размер CTU (например, 6, как в HEVC версии 1), и 0 - в противном случае.

Сигнализация ATMVP в заголовке вырезки: five\_minus\_max\_num\_merge\_cand задает максимальное количество кандидатов MVP слияния, поддерживаемых в вырезке, вычтенное из 5. Максимальное количество кандидатов MVP слияния MaxNumMergeCand определяется как:  $\text{MaxNumMergeCand} = (\text{atmvpEnableFlag} \ll 6) - \text{five\_minus\_max\_num\_merge\_cand}$  (7-41).

Значение five\_minus\_max\_num\_merge\_cand должно быть ограничено так, чтобы MaxNumMergeCand находилось в диапазоне от 1 до  $(\text{atmvpEnableFlag} \ll 6)$  включительно.

Альтернативно, five\_minus\_max\_num\_merge\_cand изменяется на six\_minus\_max\_num\_merge\_cand и семантика является следующей: six\_minus\_max\_num\_merge\_cand задает максимальное количество кандидатов MVP слияния, поддерживаемых в вырезке, вычтенное из 6. Максимальное количество кандидатов MVP слияния MaxNumMergeCand выводится как  $\text{MaxNumMergeCand} = 6 - \text{six\_minus\_max\_num\_merge\_cand}$  (7-41).

Альтернативно, max\_num\_merge\_cand\_minus1 напрямую сигнализируется.

В некоторых примерах другие изменения синтаксиса не нужны, и кандидат ATMVP идентифицируется с помощью merge\_idx, который может находиться в диапазоне от 0 до 5 включительно.

Процессы декодирования, относящиеся к ATMVP.

Следующие процессы декодирования могут быть реализованы, например, видеодекодером 30 для формулирования кандидата ATMVP и включения его в качестве части списка кандидатов слияния:

Идентификация временного вектора первого этапа:

Установить переменную mtSrcPOC в значение POC рядом расположенной картинке, используемой в TMVP, tV в нулевой вектор, и atmvpAvaFlag устанавливается в 0.

Для каждой позиции N пространственных соседних позиций, являющихся A1, B1, B0, A0 и B2, применить следующее:

- dir устанавливается равным collocated\_from\_10\_flag;

- для X, равного dir до (1-dir) включительно, если текущая вырезка представляет собой B-вырезку, или просто X равно 0, если текущая вырезка не является B-вырезкой, применяется следующее:

когда соседний блок N доступен, и он не является интра-кодированным, и RefIdxX[N] больше или равно 0 (обозначим через MVLX[N] и RefIdxX[N] вектор движения и опорный индекс соседнего блока N, соответствующего RefPicListX), следующие этапы применяются в следующем порядке:

- mtSrcPOC устанавливается равным значению POC для RefPicListX[RefIdxX[N]];

- tV устанавливается равным MVLX[N];

- atmvpAvaFlag устанавливается в 1;

- завершить этот процесс.

Идентификация доступного источника движения:

Установить список картинок как CanPicATMVP, чтобы быть пустым списком.

CanPicATMVP[0] устанавливается как картинка со значением POC, равным mtSrcPOC.

i устанавливается равным 1;

MotionSrcPic является пустой картинкой и может обновляться, как указано ниже.

Для каждого доступного списка X опорных картинок применяется следующее:

- dir устанавливается в collocated\_from\_10\_flag;

для X, равного dir до (1-dir), включительно, если текущая вырезка является B-вырезкой, или просто X равно 0, если текущая вырезка не является B-вырезкой, применяется следующее:

для каждого idx от 0 до num\_ref\_active\_1X\_minus1;

CanPicATMVP[i++] = RefPicListX[idx];

пусть (CurrPosX, CurrPosY) - координата позиции верхнего левого пиксела текущей PU.

Для n, равного от 0 до i, включительно, применяется следующее:

- если n не равно 0, масштабировать tV к картинке CanPicATMVP[n], чтобы получить tScaledVector, где релевантными картинками для tV являются текущая картинка и CanPicATMVP[0], и релевантными картинками для целевого вектора tScaledVector являются текущая картинка и CanPicATMVP[i];

- в противном случае (n равно 0) tScaledVector устанавливается равным tV.

- Получить информацию о движении блока, соответствующего центральной под-PU, из CanPicATMVP[n] следующим образом:

```
centerPosX=CurrPosX+((tScaledVector[0]+2)>>2);
```

```
centerPosY=CurrPosY+((tScaledVector[1]+2)>>2);
```

пусть (centerPosX, centerPosY) - это позиция, которая идентифицирует соответствующий блок центральной под-PU, и размер текущей PU определяется как ширина на высоту.

```
centerPosX+=CurrPosX+((width/spuWidth)>>1)*spuWidth+ (min(spuWidth,width)>>1);
```

```
centerPosY+=CurrPosY+((height/spuHeight)>>1)*spuHeight+ (min(spuHeight,height)>>1);
```

Вызвать процесс извлечения информации о движении, который захватывает информацию о движении, с картинкой mtnSrcPic, равной CanPicATMVP[n], и позицией (posX, posY). равной (centerPosX, centerPosY), в качестве входа, и флагом доступности движения под-PU SubPumtnAvaFlag, парой опорных индексов sColRefIdx0 и sColRefIdx1 и парой векторов движения sColMV0, sColMV1 в качестве выхода.

Если SubPumtnAvaFlag равен 1, применяется следующее:

- MotionSrcPic устанавливается в CanPicATMVP[n];

- tV устанавливается как tScaledVector;

- завершить этот цикл.

Процесс извлечения информации о движении.

Входом этого процесса являются картинка mtnSrcPic и позиция (posX, posY) с картинкой, и выходом этого процесса являются флаг доступности движения mtnAvaFlag, пара опорных индексов refIdx0 и refIdx1 и пара векторов движения, mv0, mv1;

(posX, posY) сначала обрезается, чтобы находиться внутри картинки mtnSrcPic.

Идентифицируется блок 4x4 (или другого наименьшего размера, который хранит информацию о движении) blkT, содержащий позицию (posX, posY);

mtnAvaFlag устанавливается равным 0.

Если blkT не является интра-кодированным и его информация о движении содержит blkTRefIdx0, blkTRefIdx1, blkTMv0 и blkTMv1, применяется следующее:

- когда либо blkTRefIdx0, либо blkTRefIdx1 больше или равно 0, mtnAvaFlag устанавливается равным 1, и следующее применяется для X, равного 0 и 1;

refIdxX устанавливается равным blkTRefIdxX;

mvX устанавливается равным blkTMvX.

Генерация движения под-PU для ATMVP:

если SubPumtnAvaFlag равно 1, вызывается следующий процесс.

Для каждой из под-PU (например, в порядке растрового сканирования) применяется следующее:

- обозначить горизонтальный индекс и вертикальный индекс текущей под-PU как k и l, соответственно, где k находится в диапазоне от 0 до width/spuWidth-1 включительно, и l находится в диапазоне от 0 до height/spuHeight-1 включительно. Например, если 16x16 PU разделена на четыре 8x8 суб-PU, значения (k, l) четырех суб-PU в порядке растрового сканирования равны: (0, 0), (1, 0), (0, 1) и (1, 1) соответственно.

- Координаты под-PU (tempPosX, tempPosY) вычисляются как (tempPosX, tempPosY)=(CurrPosX, CurrPosY)+(k\*spuWidth, l\*spuHeight).

```
- tempPosX+=((tV[0]+2)>>2);
```

```
- tempPosY+=((tV[1]+2)>>2);
```

- вызвать процесс извлечения информации о движении, который захватывает информацию о движении с картинкой mtnSrcPic, равной MotionSrcPic, и позицией (posX, posY), равной (tempPosX, tempPosY), в качестве входа и флагом доступности движения под-PU currSubPumtnAvaFlag, парой опорных индексов currSubRefIdx0 и currSubRefIdx1 и парой векторов движения currSubMV0, currSubMV1 в качестве выхода.

- Когда currSubPumtnAvaFlag равен 0, для X, равного 0 и 1 включительно, currSubRefIdxX устанавливается равным cColRefIdxX и currSubMVX устанавливается равным cColMVX.

- Для X, равного 0 и 1 включительно, масштабировать вектор движения currSubMVX к целевой опорной картинке по умолчанию для текущей картинки, которая является RefPicListX[0], аналогично тому, как в TMVP. Обозначим полученный опорный индекс и вектор движения для текущей под-PU как cSpuRefIdxX и cSpuMVX, и они выводятся следующим образом:

```
cSpuRefIdxX=(currSubRefIdxX>=0?0:-1);
```

cSpuMVX устанавливается как масштабированный вектор currSubMVX, подобно тому, как в TMVP.

Репрезентативный набор информации движения, aRefIdxX и aMVX (для X, равного 0 или 1) для этого кандидата ATMVP выводится следующим образом:

```
aRefIdxX=(cColRefIdxX>=0?0:-1);
```

aMVX устанавливается как масштабированный вектор cColMVX, подобно тому, как в TMVP.

Альтернативно, процесс выборки движения и процесс масштабирования движения одинаковы (или аналогичны), как в TMVP HEVC версии 1, то есть в подпункте 8.5.3.2.8 HEVC версии 1: "Derivation process for collocated motion vectors" (Процесс вывода для соотнесенных векторов движения) применяется для замены выделенного текста в этом подразделе. В этом случае процесс выборки движения и масштабирования движения, как в TMVP (подпункт 8.5.3.2.8), заменяют процесс выборки движения и процесс масштабирования движения, определенные выше (в том числе, как указано курсивом).

Вставка кандидата ATMVP в список кандидатов слияния.

Когда SubPmtnAvaFlag равно 1, кандидат ATMVP вставляется в список кандидатов слияния после того, как кандидат A0 (или альтернативно B0) будет протестирован и, возможно, вставлен в список кандидатов слияния.

Информация о движении для этого кандидата считается сформированной посредством aRefIdxX и aMVX (с X равным 0 или 1).

Когда кандидат TMVP доступен, он дополнительно сравнивается с репрезентативной информацией кандидата ATMVP (aRefIdxX и aMVX); только если кандидат TMVP имеет refIdxX, не равный aRefIdxX, или вектор движения, не равный aMVX (при этом X равен 0 или 1), он также добавляется в список кандидатов слияния.

Когда все кандидаты, включая кандидата ATMVP, рассматриваются прозрачно, как представлено одним набором информации о движении (до двух опорных индексов и двух ассоциированных векторов движения), используется репрезентативная информация для кандидата ATMVP. Например, в процессе вывода для комбинированных двунаправленно предсказанных кандидатов слияния используется репрезентативная информация о движении ATMVP.

Кроме того, каждый кандидат слияния снабжается тегом (может быть флагом или типом), указывающим, является ли такой кандидат кандидатом ATMVP.

Кроме того, для кандидата ATMVP, наборы информации о движении, обозначенные выше как cSpuRefIdxX и cSpuMVX для X, равного 0 и 1, и для каждой под-PU должны быть сохранены для декодирования текущей PU.

Компенсация движения, основанная на кандидате ATMVP.

Когда текущая PU кодирована с режимом слияния и merge\_id задает кандидата, указанного как кандидат ATMVP, для каждой под-PU, информация о движении cSpuRefIdxX и cSpuMVX (для X, равного 0 и 1) выводится и используется для выполнения компенсации движения для текущей под-PU. После выполнения компенсации движения остаточное декодирование и другие процессы выполняются так же, как и другие интер-режимы.

Видеокодер 20 может дополнительно отправлять синтаксические данные, такие как синтаксические данные на основе блока, синтаксические данные на основе кадра и синтаксические данные на основе GOP, в видеокодер 30, например, в заголовке кадра, заголовке блока, заголовке вырезки или заголовке GOP. Синтаксические данные GOP могут описывать несколько кадров в соответствующей GOP, а синтаксические данные кадра могут указывать режим кодирования/предсказания, используемый для кодирования соответствующего кадра.

Видеокодер 20 и видеокодер 30, каждый, могут быть реализованы как любая из множества подходящих схем кодера или декодера, если это применимо, таких как один или несколько микропроцессоров, процессоров цифровых сигналов (DSP), специализированных интегральных схем (ASIC), программируемых пользователем вентильных матриц (FPGA), дискретных логических схем, программное обеспечение, аппаратные средства, встроенное программное обеспечение или любые их комбинации. Каждый из видеокодера 20 и видеокодера 30 может быть включен в один или несколько кодеров или декодеров, каждый из которых может быть интегрирован как часть объединенного видеокодера/декодера (кодека). Устройство, включающее в себя видеокодер 20 и/или видеокодер 30, может содержать интегральную схему, микропроцессор и/или устройство беспроводной связи, такое как сотовый телефон.

Фиг. 2 является блок-схемой, иллюстрирующей пример видеокодера 20, который может реализовывать методы для расширенного предсказания временного вектора движения (ATMVP). Видеокодер 20 может выполнять интра- и интер-кодирование видеоблоков в видео вырезках. Интра-кодирование основывается на пространственном предсказании для уменьшения или удаления пространственной избыточности в видео в данном видеокadre или картинке. Интер-кодирование основывается на временном предсказании для уменьшения или удаления временной избыточности в видео в смежных кадрах или картинках видео последовательности. Интер-режим (I-режим) может относиться к любому из нескольких режимов кодирования на пространственной основе. Интер-режимы, такие как однонаправленное предсказание (P-режим) или двунаправленное (B-режим), могут относиться к любому из нескольких режимов кодирования на временной основе.

Как показано на фиг. 2, видеокодер 20 принимает текущий видеоблок в видеокadre, подлежащем кодированию. В примере согласно фиг. 2 видеокодер 20 включает в себя модуль 40 выбора режима, память 64 опорных картинок, сумматор 50, модуль 52 обработки преобразования, модуль 54 квантования и модуль 56 энтропийного кодирования. Модуль 40 выбора режима, в свою очередь, включает в себя модуль 44 компенсации движения, модуль 42 оценки движения, модуль 46 интра-предсказания и модуль 48

разбиения. Для восстановления видеоблока видеокодер 20 также включает в себя модуль 58 обратного квантования, модуль 60 обратного преобразования и сумматор 62. Фильтр деблокирования (не показан на фиг. 2) может также быть включен для фильтрации границ блоков для удаления артефактов блочности из восстановленного видео. При желании фильтр деблокирования обычно фильтрует выход сумматора 62. В дополнение к фильтру деблокирования также могут использоваться дополнительные фильтры (в контуре или после контура). Такие фильтры не показаны для краткости, но, если желательно, могут фильтровать выход сумматора 50 (как фильтр в контуре).

Во время процесса кодирования видеокодер 20 принимает видеокادر или вырезку, подлежащий(ую) кодированию. Кадр или вырезка может разделяться на несколько видеоблоков. Модуль 42 оценки движения и модуль 44 компенсации движения выполняют кодирование с интер-предсказанием принятого видеоблока относительно одного или нескольких блоков в одном или нескольких опорных кадрах для обеспечения временного предсказания. Модуль 46 интра-предсказания может альтернативно выполнять кодирование с интра-предсказанием принятого видеоблока относительно одного или более соседних блоков в том же кадре или вырезке в качестве блока, подлежащего кодированию, для обеспечения пространственного предсказания. Видеокодер 20 может выполнять несколько проходов кодирования, например, для выбора соответствующего режима кодирования для каждого блока видеоданных.

Кроме того, модуль 48 разбиения может разбивать блоки видеоданных на подблоки на основе оценки предыдущих схем разбиения в предыдущих проходах кодирования. Например, модуль 48 разбиения может сначала разбить кадр или вырезку на LCU и разбить каждую из LCU на под-CU на основе анализа скорости-искажения (например, оптимизации скорости-искажения). Модуль 40 выбора режима может дополнительно создавать структуру данных квадродерева, указывающую разбиение LCU на под-CU. CU листовых узлов квадродерева могут включать одну или несколько PU и одну или несколько TU.

Модуль 40 выбора режима может выбирать один из режимов кодирования, интра- или интер-кодирование, например, на основе результатов ошибок, и предоставляет результирующий интра- или интер-кодированный блок на сумматор 50 для генерации остаточного блока данных и на сумматор 62 для восстановления кодированного блока для использования в качестве опорного кадра. Модуль 40 выбора режима также предоставляет синтаксические элементы, такие как векторы движения, указатель интра-режима, информацию о разбиении и другую такую синтаксическую информацию, в модуль 56 энтропийного кодирования.

Модуль 42 оценки движения и модуль 44 компенсации движения могут быть высоко интегрированными, но проиллюстрированы отдельно для концептуальных целей. Оценка движения, выполняемая модулем 42 оценки движения, представляет собой процесс генерации векторов движения, которые оценивают движение для видеоблоков. Например, вектор движения может указывать смещение PU видеоблока в текущем видеокadre или картинке относительно блока предсказания в опорном кадре (или другой кодируемой единице) относительно текущего блока, кодируемого в текущем кадре (или другой кодируемой единице). Блок предсказания представляет собой блок, который, как установлено, близко соответствует кодируемому блоку с точки зрения пиксельной разности, что может быть определено суммой абсолютной разности (SAD), суммой квадратичной разности (SSD) или другими метриками разности. В некоторых примерах, видеокодер 20 может вычислять значения для суб-целочисленных пиксельных позиций опорных картинок, сохраненных в памяти 64 опорных картинок. Например, видеокодер 20 может интерполировать значения четверть пиксельных позиций, одной восьмой пиксельных позицией или других дробных пиксельных позиций опорной картинки. Следовательно, модуль 42 оценки движения может выполнять поиск движения относительно полнопиксельных позиций и дробнопиксельных позиций и выводить вектор движения с дробнопиксельной точностью.

Модуль 42 оценки движения вычисляет вектор движения для PU видеоблока в интер-кодированной вырезке путем сравнения позиции PU с позицией блока предсказания опорной картинки. Опорная картинка может быть выбрана из первого списка опорных картинок (список 0) или второго списка опорных картинок (список 1), каждый из которых идентифицирует одну или более опорных картинок, сохраненных в памяти 64 опорных картинок. Модуль 42 оценки движения отправляет вычисленный вектор движения в модуль 56 энтропийного кодирования и модуль 44 компенсации движения.

Компенсация движения, выполняемая модулем 44 компенсации движения, может включать в себя извлечение или генерацию блока предсказания на основе вектора движения, определяемого модулем 42 оценки движения. Вновь модуль 42 оценки движения и модуль 44 компенсации движения могут быть функционально интегрированными в некоторых примерах. После приема вектора движения для PU текущего видеоблока модуль 44 компенсации движения может находить блок предсказания, на который указывает вектор движения, в одном из списков опорных картинок. Сумматор 50 формирует остаточный видеоблок путем вычитания пиксельных значений блока предсказания из пиксельных значений текущего видеоблока, который кодируется, формируя пиксельные разностные значения, как обсуждается ниже. В общем, модуль 42 оценки движения выполняет оценку движения относительно компонентов яркости, и модуль 44 компенсации движения использует векторы движения, вычисленные на основе компонентов яркости, как для компонентов цветности, так и для компонентов яркости. Модуль 40 выбора режима может также генерировать синтаксические элементы, ассоциированные с видеоблоками и видео вырезкой,

для использования видеodeкодером 30 при декодировании видеоблоков видео вырезки.

Видеокoder 20 может быть сконфигурирован для выполнения любого из различных методов этого раскрытия, рассмотренных выше со ссылкой на фиг. 1. Например, модуль 44 компенсации движения может быть сконфигурирован для кодирования информации о движении для блока видеоданных с использованием AMVP или режима слияния в соответствии с методами этого раскрытия.

Предполагая, что модуль 44 компенсации движения выбирает выполнение режима слияния, модуль 44 компенсации движения может формировать список кандидатов, включающий в себя набор кандидатов слияния. Модуль 44 компенсации движения может добавлять кандидатов в список кандидатов на основе конкретного предопределенного порядка. В одном примере модуль 44 компенсации движения добавляет кандидатов в список кандидатов в порядке A1, B1, B0, A0 и затем кандидата расширенного предсказания временного вектора движения (ATMVP). Модуль 44 компенсации движения может также добавлять дополнительных кандидатов и выполнять обрезку списка кандидатов, как обсуждалось выше. В конечном счете, модуль 40 выбора режима может определять, какой из кандидатов должен использоваться для кодирования информации о движении текущего блока, и кодировать индекс слияния, представляющий выбранный кандидата.

Кроме того, в некоторых примерах модуль 44 компенсации движения может сначала определить, доступен ли кандидат ATMVP. Например, блок компенсации движения может определять соответствующий блок для текущего блока в опорной картинке и определять, доступна ли информация о движении для соответствующего блока. Модуль 44 компенсации движения может затем определить, что кандидат ATMVP (то есть соответствующий блок) доступен, когда информация о движении доступна для соответствующего блока. В некоторых примерах модуль 44 компенсации движения может определять, что информация о движении доступна для соответствующего блока, когда весь соответствующий блок (например, блок центральной позиции, как показано на фиг. 5а ниже) предсказан без использования интра-предсказания, но недоступна, когда по меньшей мере часть соответствующего блока предсказана с использованием интра-предсказания.

Аналогичным образом в некоторых примерах модуль 44 компенсации движения может определять, какой из двух потенциальных кандидатов ATMVP должен быть использован в качестве кандидата ATMVP, который в конечном счете добавляется в список кандидатов. Например, модуль 44 компенсации движения может формировать первый временной вектор движения относительно текущего блока, который идентифицирует первого кандидата ATMVP в первой исходной картинке движения, то есть первой опорной картинке. Если информация о движении недоступна для первого кандидата ATMVP, модуль 44 компенсации движения может определять, доступна ли информация о движении для второго, другого кандидата ATMVP. Второй кандидат ATMVP может быть идентифицирован с использованием того же временного вектора движения, ссылающегося на вторую, другую опорную картинку, другого временного вектора движения, ссылающегося на ту же (то есть первую) опорную картинку, или другого временного вектора движения, ссылающегося на вторую, другую опорную картинку. Опорные картинки, которые должны проверяться, как обсуждалось выше, могут находиться в порядке возрастания опорных индексов в списке опорных картинок. Аналогично, если используются различные временные векторы движения, временные векторы движения могут выбираться в предопределенном порядке от временных векторов соседних блоков к текущему блоку.

Кроме того, модуль 44 компенсации движения может определять, доступен ли вектор движения для под-PU в кандидате ATMVP для конкретного списка опорных картинок. Если это так, то вектор движения считается доступным для этого списка опорных картинок. В противном случае вектор движения считается недоступным для этого списка опорных картинок. Альтернативно, если вектор движения доступен для другого списка опорных картинок, модуль 44 компенсации движения может изменять информацию о движении путем масштабирования вектора движения, чтобы указывать на целевой опорный кадр в первом списке опорных картинок, как обсуждалось выше.

Модуль 46 интра-предсказания может интра-предсказывать текущий блок, в качестве альтернативы интер-предсказанию, выполняемому модулем 42 оценки движения и модулем 44 компенсации движения, как описано выше. В частности, модуль 46 интра-предсказания может определять режим интра-предсказания для использования для кодирования текущего блока. В некоторых примерах модуль 46 интра-предсказания может кодировать текущий блок, используя различные режимы интра-предсказания, например, во время отдельных проходов кодирования, и модуль 46 интра-предсказания (или модуль 40 выбора режима в некоторых примерах) может выбирать подходящий режим интра-предсказания для использования из испытанных режимов.

Например, модуль 46 интра-предсказания может вычислять значения искажения-скорости с использованием анализа искажения-скорости для различных испытанных режимов интра-предсказания и выбирать режим интра-предсказания, имеющий наилучшие характеристики искажения-скорости среди проверенных режимов. Анализ скорости-искажения обычно определяет величину искажения (или ошибки) между закодированным блоком и исходным некодированным блоком, который кодировался, чтобы создать закодированный блок, а также битрейт (то есть количество битов), используемый, чтобы создать закодированный блок. Модуль 46 интра-предсказания может вычислять отношения из искажений и ско-

ростей для различных закодированных блоков, чтобы определять, какой режим интра-предсказания имеет наилучшее значение искажения-скорости для блока.

После выбора режима интра-предсказания для блока модуль 46 интра-предсказания может предоставлять информацию, указывающую выбранный режим интра-предсказания для блока, в модуль 56 энтропийного кодирования. Модуль 56 энтропийного кодирования может кодировать информацию, указывающую выбранный режим интра-предсказания. Видеокодер 20 может включать в передаваемый битовый поток данные конфигурации, которые могут включать в себя множество таблиц индексов режима интра-предсказания и множество таблиц индексов модифицированного режима интра-предсказания (также называемых таблицами отображения кодовых слов), определения контекстов кодирования для различных блоков и указания наиболее вероятного режима интра-предсказания, таблицу индексов режима интра-предсказания и таблицу индексов модифицированного режима интра-предсказания для использования для каждого из контекстов.

Видеокодер 20 формирует остаточный видео-блок путем вычитания данных предсказания из модуля 40 выбора режима из кодируемого исходного видеоблока. Сумматор 50 представляет компонент или компоненты, которые выполняют эту операцию вычитания. Модуль 52 обработки преобразования применяет преобразование, такое как дискретное косинусное преобразование (DCT) или концептуально подобное преобразование, к остаточному блоку, создавая видеоблок, содержащий остаточные значения коэффициента преобразования. Модуль 52 обработки преобразования может выполнять другие преобразования, которые концептуально подобны DCT. Также можно использовать вейвлет-преобразования, целочисленные преобразования, поддиапазонные преобразования или другие типы преобразований.

В любом случае модуль 52 обработки преобразования применяет преобразование к остаточному блоку, создавая блок остаточных коэффициентов преобразования. Преобразование может преобразовывать остаточную информацию из области пиксельных значений в область преобразования, например в частотную область. Модуль 52 обработки преобразования может отправлять полученные коэффициенты преобразования в модуль 54 квантования. Модуль 54 квантования квантует коэффициенты преобразования для дальнейшего уменьшения скорости передачи битов. Процесс квантования может уменьшить битовую глубину, ассоциированную с некоторыми или всеми коэффициентами. Степень квантования может быть изменена путем корректировки параметра квантования. В некоторых примерах модуль 54 квантования может затем выполнять сканирование матрицы, включающей в себя квантованные коэффициенты преобразования. Альтернативно, модуль 56 энтропийного кодирования может выполнять сканирование.

После квантования модуль 56 энтропийного кодирования энтропийно кодирует квантованные коэффициенты преобразования. Например, модуль 56 энтропийного кодирования может выполнять контекстно-адаптивное кодирование с переменной длиной (CAVLC), контекстно-адаптивное двоичное арифметическое кодирование (CABAC), основанное на синтаксисе контекстно-адаптивное двоичное арифметическое кодирование (SBAC), энтропийное кодирование с разделением на интервалы вероятности (PIPE) или другой метод энтропийного кодирования. В случае контекстно-зависимого энтропийного кодирования контекст может быть основан на соседних блоках. После энтропийного кодирования модулем 66 энтропийного кодирования закодированный битовый поток может быть передан на другое устройство (например, видеодекодер 30) или архивирован для последующей передачи или поиска.

Модуль 58 обратного квантования и модуль 60 обратного преобразования применяют обратное квантование и обратное преобразование, соответственно, для восстановления остаточного блока в пиксельной области, например для последующего использования в качестве опорного блока. Модуль 44 компенсации движения может вычислять опорный блок, добавляя остаточный блок к блоку предсказания одного из кадров памяти 64 опорных картинок. Модуль 44 компенсации движения может также применять один или несколько интерполяционных фильтров к восстановленному остаточному блоку для вычисления субцелочисленных пиксельных значений для использования при оценке движения. Сумматор 62 суммирует восстановленный остаточный блок со скомпенсированным по движению блоком предсказания, созданным модулем 44 компенсации движения, для создания восстановленного видеоблока для хранения в памяти 64 опорных картинок. Восстановленный видеоблок может использоваться модулем 42 оценки движения и модулем 44 компенсации движения в качестве опорного блока для интеркодирования блока в последующем видеокадре.

Таким образом, видеокодер 20 согласно фиг. 2 представляет собой пример устройства кодирования видео, сконфигурированного для формирования, для текущего блока видеоданных, списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков к текущему блоку и, непосредственно после четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), кодирования индекса в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и кодирования текущего блока видеоданных с использованием информации о движении идентифицированного кандидата слияния.

Фиг. 3 представляет собой блок-схему, иллюстрирующую пример видеодекодера 30, который мо-

жет реализовывать методы для расширенного предсказания временного вектора движения (ATMVP). В примере согласно фиг. 3 видеодекодер 30 включает в себя модуль 70 энтропийного декодирования, модуль 72 компенсации движения, модуль 74 интра-предсказания, модуль 76 обратного квантования, модуль 78 обратного преобразования, память 82 опорных картинок и сумматор 80. Видеодекодер 30 может в некоторых примерах выполнять проход декодирования, в общем обратный проходу кодирования, описанному относительно видеокодера 20 (фиг. 2). Модуль 72 компенсации движения может генерировать данные предсказания на основе векторов движения, принятых из модуля 70 энтропийного декодирования, в то время как модуль 74 интра-предсказания может генерировать данные предсказания на основе указателей режима интра-предсказания, принятых из модуля 70 энтропийного декодирования.

Во время процесса декодирования видеодекодер 30 принимает закодированный битовый поток видео, который представляет видеоблоки закодированной видеовырезки и ассоциированные с ним синтаксические элементы из видеокодера 20. Модуль 70 энтропийного декодирования видеодекодера 30 энтропийно декодирует битовый поток для генерации квантованных коэффициентов, векторов движения или указателей режима интра-предсказания и другие синтаксические элементы. Модуль 70 энтропийного декодирования пересылает векторы движения и другие синтаксические элементы в модуль 72 компенсации движения. Видеодекодер 30 может принимать синтаксические элементы на уровне видеовырезки и/или уровне видеоблока.

Когда видеовырезка кодируется как интра-кодированная (I) вырезка, модуль 74 интра-предсказания может генерировать данные предсказания для видеоблока текущей видеовырезки на основе сигнализированного режима интра-предсказания и данных из ранее декодированных блоков текущего кадра или картинки. Когда видеокادر закодирован как интер-кодированная вырезка (то есть B, P или GPB), модуль 72 компенсации движения создает блоки предсказания для видеоблока текущей видеовырезки на основе векторов движения и других синтаксических элементов, полученных от модуля 70 энтропийного декодирования. Блоки предсказания могут быть получены из одной из опорных картинок в одном из списков опорных картинок. Видеодекодер 30 может создавать списки опорных кадров, список 0 и список 1, используя методы построения по умолчанию на основе опорных картинок, хранящихся в памяти 82 опорных картинок.

Модуль 72 компенсации движения определяет информацию предсказания для видеоблока текущей видео вырезки, анализируя векторы движения и другие синтаксические элементы, и использует информацию предсказания для создания блоков предсказания для текущего декодируемого видеоблока. Например, модуль 72 компенсации движения использует некоторые из принятых синтаксических элементов для определения режима предсказания (например, интра- или интер-предсказания), использованного для кодирования видеоблоков видеовырезки, типа вырезки интер-предсказания (например, B-вырезка, P-вырезка или GPB-вырезка), информации о конструкции для одного или нескольких списков опорных картинок для вырезки, векторов движения для каждого интер-кодированного видеоблока вырезки, состояния интер-предсказания для каждого интер-кодированного видеоблока вырезки и другой информации для декодирования видеоблоков в текущей видеовырезке.

Модуль 72 компенсации движения также может выполнять интерполяцию на основе фильтров интерполяции. Модуль 72 компенсации движения может использовать фильтры интерполяции, как было использовано видеокодером 20 во время кодирования видеоблоков, для вычисления интерполированных значений для субцелочисленных пикселей опорных блоков. В этом случае модуль 72 компенсации движения может определять фильтры интерполяции, использованные видеокодером 20, из принятых синтаксических элементов и использовать фильтры интерполяции для создания блоков предсказания.

Видеодекодер 30 может быть сконфигурирован для выполнения любого из различных методов настоящего раскрытия, рассмотренных выше со ссылкой на фиг. 1. Например, модуль 72 компенсации движения может быть сконфигурирован для определения того, закодирована ли информация движения для блока видеоданных с использованием AMVP или режима слияния в соответствии с методами настоящего раскрытия. Более конкретно, модуль 70 энтропийного декодирования может декодировать один или несколько синтаксических элементов, представляющих то, как информация движения закодирована для текущего блока.

Предполагая, что синтаксические элементы указывают, что выполняется режим слияния, модуль 72 компенсации движения может сформировать список кандидатов, включающий в себя набор кандидатов слияния. Модуль 72 компенсации движения может добавлять кандидатов в список кандидатов на основе конкретного предопределенного порядка. В одном примере модуль 72 компенсации движения добавляет кандидатов в список кандидатов в порядке A1, B1, B0, A0 и затем кандидата расширенного предсказания временного вектора движения (ATMVP). Модуль 72 компенсации движения может также добавлять дополнительных кандидатов и выполнять обрезку списка кандидатов, как обсуждалось выше. В конечном счете, модуль 72 компенсации движения может декодировать индекс слияния, представляющий, какой из кандидатов использован для кодирования информации движения для текущего блока.

Кроме того, в некоторых примерах модуль 72 компенсации движения может сначала определить, доступен ли кандидат ATMVP. Например, модуль компенсации движения может определять соответствующий блок для текущего блока в опорной картинке и определять, доступна ли информация движения

для соответствующего блока. Модуль 72 компенсации движения может затем определить, что кандидат ATMVP (то есть соответствующий блок) доступен, когда информация о движении доступна для соответствующего блока. В некоторых примерах модуль 72 компенсации движения может определить, что информация о движении доступна для соответствующего блока, когда весь соответствующий блок предсказан без использования интра-предсказания, но недоступна, когда по меньшей мере часть соответствующего блока предсказана с использованием интра-предсказания.

Аналогичным образом в некоторых примерах модуль 72 компенсации движения может определять, какого из двух потенциальных кандидатов ATMVP следует использовать в качестве кандидата ATMVP, в конечном счете добавляемого в список кандидатов. Например, модуль 72 компенсации движения может формировать первый временной вектор движения относительно текущего блока, который идентифицирует первого кандидата ATMVP в первой исходной картинке движения, то есть первой опорной картинке. Если информация о движении недоступна для первого кандидата ATMVP, модуль 72 компенсации движения может определить, доступна ли информация движения для второго, другого кандидата ATMVP. Второй кандидат на ATMVP может быть идентифицирован с использованием того же временного вектора движения, ссылающегося на вторую, другую опорную картинку, другого временного вектора движения, ссылающегося на ту же (то есть первую) опорную картинку, или другого временного вектора движения, ссылающегося на вторую, другую опорную картинку. Опорные картинки, которые должны проверяться, как обсуждалось выше, могут находиться в порядке возрастания опорных индексов в списке опорных картинок. Аналогично, если используются различные временные векторы движения, то временные векторы движения могут быть выбраны в предопределенном порядке из временных векторов соседних блоков к текущему блоку.

Кроме того, модуль 72 компенсации движения может определять, доступен ли вектор движения для под-PU в кандидате ATMVP для конкретного списка опорных картинок. Если это так, то вектор движения считается доступным для этого списка опорных картинок. В противном случае, вектор движения считается недоступным для этого списка опорных картинок. Альтернативно, если вектор движения доступен для другого списка опорных картинок, модуль 72 компенсации движения может изменять информацию о движении путем масштабирования вектора движения, чтобы указывать на целевую опорную картинку в первом списке опорных картинок, как обсуждалось выше.

Модуль 76 обратного квантования обратно квантует, т.е. деквантует квантованные коэффициенты преобразования, представленные в битовом потоке и энтропийно декодированные модулем 70 энтропийного декодирования. Процесс обратного квантования может включать в себя использование параметра квантования  $QP_{\gamma}$ , вычисленного видеодекодером 30 для каждого видеоблока в видеовырезке, чтобы определить степень квантования и, аналогично, степень обратного квантования, которая должна применяться.

Модуль 78 обратного преобразования применяет обратное преобразование, например обратное DCT, обратное целочисленное преобразование или концептуально подобный процесс обратного преобразования, к коэффициентам преобразования для создания остаточных блоков в пиксельной области.

После того, как модуль 72 компенсации движения генерирует блок предсказания для текущего видеоблока на основе векторов движения и других синтаксических элементов, видеодекодер 30 формирует декодированный видеоблок путем суммирования остаточных блоков из модуля 78 обратного преобразования с соответствующими блоками предсказания, сгенерированными блоком 72 компенсации движения. Сумматор 80 представляет компонент или компоненты, которые выполняют эту операцию суммирования. При желании фильтр деблокирования может также применяться для фильтрации декодированных блоков, чтобы удалить артефакты блочности. Другие контурные фильтры (либо в контуре кодирования, либо после контура кодирования) также могут использоваться для сглаживания пиксельных переходов или улучшения иным образом качества видео. Декодированные видеоблоки в данном кадре или картинке затем сохраняются в памяти 82 опорных картинок, в которой хранятся опорные картинки, используемые для последующей компенсации движения. Память 82 опорных картинок также хранит декодированное видео для последующего представления на устройстве отображения, таком как устройство 32 отображения согласно фиг. 1.

Таким образом, видеодекодер 30 представляет собой пример устройства кодирования видео, сконфигурированного для формирования, для текущего блока видеоданных, списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков к текущему блоку и, непосредственно после четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), кодирования индекса в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и кодирования текущего блока видеоданных с использованием информации о движении идентифицированного кандидата слияния.

Фиг. 4 представляет собой концептуальную диаграмму, иллюстрирующую пространственных соседних кандидатов в HEVC. Пространственные кандидаты MV получены из соседних блоков, показанных на фиг. 4, для конкретной PU (PU0), хотя способы генерации кандидатов из блоков различаются для

режимов слияния и AMVP.

В режиме слияния могут быть получены до четырех пространственных кандидатов MV в порядках, показанных на фиг. 4 (а) числами, и порядок следующий: левый (0, A1), верхний (1, B1), верхний правый (2, B0), нижний левый (3, A0) и верхний левый (4, B2), как показано на фиг. 4(а). То есть на фиг. 4 (а), блок 100 включает в себя PU0 104A и PU1 104B. Когда устройство кодирования видео должно кодировать информацию о движении для PU0 104A с использованием режима слияния, устройство кодирования видео добавляет информацию о движении из пространственных соседних блоков 108A, 108B, 108C, 108D и 108E в список кандидатов в этом порядке. Блоки 108A, 108B, 108C, 108D и 108E также могут упоминаться, соответственно, как блоки A1, B1, B0, A0 и B2, как в HEVC.

В режиме AMVP соседние блоки делятся на две группы: левую группу, включающую в себя блоки 0 и 1, и верхнюю группу, включающую в себя блоки 2, 3 и 4, как показано на фиг. 4 (b). Эти блоки обозначены соответственно как блоки 110A, 110B, 110C, 110D и 110E на фиг. 4 (b). В частности, на фиг. 4(b) блок 102 включает в себя PU0 106A и PU1 106B, а блоки 110A, 110B, 110C, 110D и 110E представляют пространственных соседей для PU0 106A. Для каждой группы потенциальный кандидат в соседнем блоке, ссылающийся на ту же самую опорную картинку, что и указанная сигнализированным опорным индексом, имеет наивысший приоритет для выбора, чтобы сформировать окончательного кандидата группы. Возможно, что все соседние блоки не содержат вектор движения, указывающий на ту же самую опорную картинку. Поэтому, если такой кандидат не может быть найден, первый доступный кандидат будет масштабироваться для формирования окончательного кандидата; таким образом, временные разности расстояний могут быть скомпенсированы.

На фиг. 5 представлена концептуальная диаграмма, иллюстрирующая предсказание временного вектора движения в HEVC. В частности, фиг. 5(a) иллюстрирует пример CU 120, включающей в себя PU0 122A и PU1 122B. PU0 122 A включает в себя центральный блок 126 для PU 122A и нижний правый блок 124 для PU0 122A. На фиг. 5(a) также показан внешний блок 128, для которого информация о движении может быть предсказана из информации о движении PU0 122A, как обсуждается ниже. Фиг. 5(b) иллюстрирует текущую картинку 130, включающую в себя текущий блок 138, для которого должна быть предсказана информация о движении. В частности, фиг. 5(b) показывает картинку 134, соотношенную с текущей картинкой 130 (включающую в себя блок 140, соотношенный с текущим блоком 138), текущую опорную картинку 132 и соотношенную опорную картинку 136. Соотношенный блок 140 предсказывается с использованием вектора 144 движения, который используется как предсказатель временного вектора движения (TMVP) 142 для информации о движении блока 138.

Устройство кодирования видео может добавить кандидата TMVP (например, кандидата 142 TMVP) в список кандидатов MV после любых кандидатов пространственного вектора движения, если TMVP включено и кандидат TMVP доступен. Процесс вывода вектора движения для кандидата TMVP одинаков для режимов слияния и AMVP. Однако целевой опорный индекс для кандидата TMVP в режиме слияния устанавливается в 0, согласно HEVC.

Местоположение первичного блока для вывода кандидата TMVP является нижним правым блоком вне соотношенной PU, как показано на фиг. 5(a) как блок 124 относительно PU0 122 A, чтобы компенсировать смещение к верхнему и левому блокам, используемым для генерации пространственных соседних кандидатов. Однако, если блок 124 расположен за пределами текущей строки СТВ или информация о движении недоступна для блока 124, то блок заменяется центральным блоком 126 PU, как показано на фиг. 5(a).

Вектор движения для кандидата 142 TMVP выводится из рядом расположенного блока 140 рядом расположенной картинкой 134, как указано в информации уровня вырезки.

Подобно временному прямому режиму в AVC вектор движения кандидата TMVP может подвергаться масштабированию вектора движения, которое выполняется для компенсации разностей расстояний порядковых отсчетов картинок (РОС) между текущей картинкой 130 и текущей опорной картинкой 132 и соотношенной картинкой 134 и соотношенной картинкой 136. То есть вектор 144 движения может масштабироваться для создания кандидата 142 TMVP на основе этих разностей РОС.

Несколько аспектов режимов слияния и AMVP для HEVC обсуждаются ниже.

#### Масштабирование вектора движения

Предполагается, что значение вектора движения пропорционально расстоянию между картинками во время представления. Вектор движения ассоциирует две картинки: опорную картинку и картинку, содержащую вектор движения (а именно, вмещающую картинку). Когда вектор движения используется видеокодером 20 или видеодекoderом 30 для предсказания другого вектора движения, расстояние между вмещающей картинкой и опорной картинкой вычисляется на основе значений порядкового отсчета картинок (РОС).

Для вектора движения, подлежащего предсказанию, его ассоциированная вмещающая картинка и опорная картинка различны. То есть существуют два значения разности РОС для двух различных векторов движения: первым вектором движения, подлежащим предсказанию, и вторым вектором движения, используемым для предсказания первого вектора движения. Кроме того, первая разность РОС представляет собой разность между текущей картинкой и опорной картинкой первого вектора движения, а вторая

разность РОС представляет собой разность между картинкой, содержащей второй вектор движения, и опорной картинкой, на которую ссылается второй вектор движения. Второй вектор движения можно масштабировать на основе этих двух расстояний РОС. Для пространственного соседнего кандидата вмещающие картинки для двух векторов движения одинаковы, тогда как опорные картинки различны. В HEVC масштабирование вектора движения применяется как к TMVP, так и AMVP для пространственных и временных соседних кандидатов.

Генерация искусственных кандидатов вектора движения: если список кандидатов вектора движения не завершен, искусственные кандидаты вектора движения могут генерироваться и вставляться в конце списка до тех пор, пока список не включит предопределенное количество кандидатов.

В режиме слияния существуют два типа искусственных кандидатов MV: комбинированные кандидаты, выведенные только для B-вырезов, и нулевые кандидаты, используемые только для AMVP, если первый тип не обеспечивает достаточных искусственных кандидатов.

Для каждой пары кандидатов, которые уже находятся в списке кандидатов и имеют необходимую информацию движения, двунаправленные комбинированные кандидаты вектора движения выводятся комбинацией вектора движения первого кандидата, ссылающегося на картинку в списке 0, и вектора движения второго кандидата, ссылающегося на картинку в списке 1.

#### Процесс обрезки для вставки кандидата

Кандидаты из разных блоков могут оказаться одинаковыми, что снижает эффективность списка кандидатов слияния/AMVP. Для решения этой проблемы может быть применен процесс обрезки. Согласно процессу обрезки устройство кодирования видео сравнивает одного кандидата с другими в текущем списке кандидатов, чтобы избежать вставки идентичного кандидата в определенной степени. Чтобы уменьшить сложность, применяется только ограниченное количество процессов обрезки вместо сравнения каждого потенциального кандидата со всеми другими существующими кандидатами, уже включенными в список.

Фиг. 6 иллюстрирует примерную структуру предсказания для 3D-HEVC. 3D-HEVC - это расширение 3D-видео HEVC, разрабатываемое JCT-3V. Некоторые методы, относящиеся к методам настоящего раскрытия, описаны со ссылкой на фиг. 6 и 7 ниже.

На фиг. 6 показана структура многовидового предсказания для трехвидового случая. V3 обозначает базовый вид, а картинка в небазовом виде (V1 или V5) может быть предсказана из картинок в зависимом (базовом) виде того же временного экземпляра.

Интер-видовое предсказание выборки (из восстановленных выборок) поддерживается в MV-HEVC, типичная структура предсказания которого показана на фиг. 8.

Как MV-HEVC, так и 3D-HEVC совместимы с HEVC таким образом, что базовый (текстурный) вид декодируется декодером HEVC (версии 1). Тестовая модель для MV-HEVC и 3D-HEVC описана в публикации: Zhang et al., "Test Model 6 of 3D-HEVC и MV-HEVC", документ JCT-3V ISO/IEC JTC1/SC29/WG11 N13940, доступной на вебсайте [mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding/test-model-6-3d-hevc-and-mv-hevc](http://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding/test-model-6-3d-hevc-and-mv-hevc) по состоянию на 26 января 2015.

В MV-HEVC текущая картинка в небазовом виде может быть предсказана как картинками в том же виде, так и картинками в опорном виде того же временного экземпляра путем помещения всех этих картинок в списки опорных картинок данной картинки.

Таким образом, список опорных картинок текущей картинки содержит как временные опорные картинки, так и межвидовые опорные картинки.

Вектор движения, ассоциированный с опорным индексом, соответствующим временной опорной картинке, обозначается как временной вектор движения.

Вектор движения, ассоциированный с опорным индексом, соответствующим межвидовой опорной картинке, обозначается как диспаратный (характеризующий различие) вектор движения.

3D-HEVC поддерживает все функции MV-HEVC. Таким образом, обеспечена возможность межвидового предсказания выборки, как указано выше.

Кроме того, поддерживаются более продвинутые инструменты только текстурного кодирования и инструменты связанного/зависимого от глубины кодирования.

Инструменты только текстурного кодирования часто требуют идентификации соответствующих блоков (между видами), которые могут принадлежать одному и тому же объекту. Следовательно, вывод вектора различия является базовой технологией в 3D-HEVC.

Фиг. 7 представляет собой концептуальную диаграмму, иллюстрирующую межвидовое предсказание движения на основе под-PU в 3D-HEVC. На фиг. 7 показана текущая картинка 160 текущего вида (V1) и соотнесенная картинка 162 в опорном виде (V0). Текущая картинка 160 включает в себя текущую PU 164, включающую в себя четыре под-PU 166A-166D (под-PU 166). Соответствующие векторы 174A-174D различия (векторы 174 различия) идентифицируют соответствующие под-PU 168A-168D для под-PU 166 в соотнесенной картинке 162. В 3D-HEVC способ межвидового предсказания движения на уровне под-PU выполняется для кандидата межвидового слияния, т.е. кандидата, выведенного из опорного блока в опорном виде.

Когда такой режим разрешен, текущая PU 164 может соответствовать опорной области (с тем же

размером, что и текущая PU, идентифицированная вектором различия) в опорном виде, и опорная область может иметь более богатую информацию о движении, чем требуется для генерации одного набора информации о движении обычно для PU. Поэтому может быть использован способ межвидового предсказания движения на уровне под-PU (SPIVMP), как показано на фиг. 7.

Этот режим также может сигнализироваться как специальный кандидат слияния. Каждая из под-PU содержит полный набор информации о движении. Следовательно, PU может содержать несколько наборов информации о движении.

Наследование параметра движения (MPI) на основе под-PU в 3D-HEVC

Аналогично, в 3D-HEVC кандидат MPI также может быть расширен таким же образом, как межвидовое предсказание движения на уровне под-PU. Например, если PU текущей глубины имеет рядом расположенную область, которая содержит несколько PU, то PU текущей глубины может быть разделена на под-PU, каждая из которых может иметь различный набор информации о движении. Этот метод называется MPI на основе под-PU. То есть векторы 172A-172D движения соответствующих под-PU 168A-168D могут наследоваться под-PU 166A-166D в качестве векторов 170A-170D движения, как показано на фиг. 7.

Информация, относящаяся к под-PU, для 2D кодирования видео: В патентной заявке США № 61/883,111, которая включена в настоящий документ посредством ссылки во всей полноте, описана схема расширенного TMVP на основе PU. В одноуровневом кодировании предлагается схема двухэтапного расширенного предсказания временного вектора движения.

Первый этап состоит в том, чтобы вывести вектор, идентифицирующий соответствующий блок текущей единицы предсказания (PU) в опорной картинке, и второй этап состоит в том, чтобы извлечь несколько наборов информации движения из соответствующего блока и назначить их для под-PU данной PU. Поэтому каждая под-PU в PU компенсируется по движению отдельно.

Концепция ATMVP суммируется следующим образом.

1. Вектор на первом этапе может быть выведен из пространственных и временных соседних блоков текущей PU.

2. Этот процесс может быть реализован как активация кандидата слияния среди всех других кандидатов слияния.

Применительно к одноуровневому кодированию и предсказанию временного вектора движения на основе под-PU, PU или CU могут иметь данные для уточнения движения, которые должны переноситься поверх предсказателей.

Несколько аспектов схемы согласно заявке 61/883,111 выделяются следующим образом.

1. Первый этап вывода вектора также может быть упрощен до только нулевого вектора.

2. Первый этап вывода вектора может включать идентификацию совместно вектора движения и ассоциированной с ним картинки. Предложены различные способы выбора ассоциированной картинки и дальнейшего определения вектора движения как вектора первого этапа.

3. Если информация о движении во время вышеуказанного процесса недоступна, для подстановки используется "вектор первого этапа".

4. Вектор движения, идентифицированный из временного соседа, должен быть масштабирован для использования для текущей под-PU, аналогично масштабированию вектора движения в TMVP. Однако то, к какой опорной картинке может масштабироваться такой вектор движения, может быть спроектировано одним из следующих способов:

a. Картинка идентифицируется фиксированным опорным индексом текущей картинки.

b. Картинка идентифицируется как опорная картинка соответствующего временного соседа, если также доступна в списке опорных картинок текущей картинки.

c. Картинка устанавливается как рядом расположенная картинка, идентифицированная на первом этапе, и из которой захватываются векторы движения.

Фиг. 8 представляет собой концептуальную диаграмму, иллюстрирующую предсказание движения на основе под-PU из опорной картинки. В этом примере текущая картинка 180 включает в себя текущую PU 184 (например, PU). В этом примере вектор 192 движения идентифицирует PU 186 опорной картинки 182 относительно PU 184. PU 186 разбивается на под-PU 188A-188D, каждая из которых имеет соответствующие векторы 190A-190D движения. Таким образом, хотя текущая PU 184 фактически не разбита на отдельные под-PU, в этом примере текущая PU 184 может быть предсказана с использованием информации о движении из под-PU 188A-188D. В частности, устройство кодирования видео может кодировать под-PU текущей PU 184, используя соответствующие векторы 190A-190D движения. Однако устройству кодирования видео не требуется кодировать синтаксические элементы, указывающие, что текущая PU 184 разбита на под-PU. Таким образом, текущая PU 184 может быть эффективно предсказана с использованием множества векторов 190A-190D движения, унаследованных из соответствующих под-PU 188A-188D, без служебной нагрузки сигнализации синтаксических элементов, используемой для разбиения текущей PU 184 на несколько под-PU.

Фиг. 9 представляет собой концептуальную диаграмму, иллюстрирующую релевантные картинки в ATMVP (аналогично TMVP). В частности, фиг. 9 иллюстрирует текущую картинку 204, исходную кар-

тинку 206 движения и опорные картинку 200, 202. В частности, текущая картинка 204 включает в себя текущий блок 208. Временной вектор 212 движения идентифицирует соответствующий блок 210 исходной картинку 206 движения относительно текущего блока 208. Соответствующий блок 210, в свою очередь, включает в себя вектор 214 движения, который ссылается на опорную картинку 202 и действует как расширенный предсказатель временного вектора движения для по меньшей мере части текущего блока 208, например под-PU текущего блока 208. То есть вектор 214 движения может быть добавлен как предсказатель вектора движения кандидата для текущего блока 208. Если выбрано, то по меньшей мере часть текущего блока 208 может быть предсказана с использованием соответствующего вектора движения, а именно вектора движения 216, который ссылается на опорную картинку 200.

На фиг. 10 представлена блок-схема последовательности операций, иллюстрирующая примерный способ добавления кандидата ATMVP к списку кандидатов во время процесса кодирования в соответствии с методами настоящего раскрытия.

Способ согласно фиг. 10 описывается как выполняемый видеокодером 20 (фиг. 1 и 2). Однако следует понимать, что другие устройства кодирования могут быть сконфигурированы для выполнения этого или аналогичного способа.

Первоначально видеокодер 20 получает блок видеоданных, подлежащих кодированию (не показано на фиг. 10). Блок может включать в себя набор пространственных соседей, таких как показанные на фиг. 4(a) и фиг. 4(b). Модуль 44 компенсации движения может создавать список кандидатов, сначала добавляя левого пространственного кандидата в список кандидатов (250). То есть в отношении фиг. 4(a), предполагая, что PU0 104A является кодируемым блоком, видеокодер 20 может сначала вставить пространственного соседа 108A в список кандидатов.

Затем модуль 44 компенсации движения может добавить вышеуказанного пространственного кандидата в список кандидатов (252). В отношении фиг. 4(a) видеокодер 20 может вставить пространственного соседа 108B в список кандидатов.

Затем модуль 44 компенсации движения может добавить верхнего правого пространственного кандидата в список кандидатов (254). В отношении фиг. 4(a) видеокодер 20 может вставить пространственного соседа 108C в список кандидатов.

Затем модуль 44 компенсации движения может добавить нижнего левого пространственного кандидата в список кандидатов (256). В отношении фиг. 4(a) видеокодер 20 может вставить пространственного соседа 108D в список кандидатов.

Затем модуль 44 компенсации движения может добавить кандидата расширенного предсказания временного вектора движения (ATMVP) в список кандидатов (258). Как обсуждалось выше, кандидат ATMVP может представлять собой соответствующий блок, идентифицированный временным вектором, как показано и обсуждено в отношении, например, фиг. 8 и 9. Кроме того, в некоторых примерах модуль 44 компенсации движения может сначала определить, доступен ли кандидат ATMVP. Например, блок компенсации движения может определять соответствующий блок для текущего блока в опорной картинке и определять, доступна ли информация движения для соответствующего блока. Модуль 44 компенсации движения может затем определить, что кандидат ATMVP (то есть соответствующий блок) доступен, когда информация о движении доступна для соответствующего блока. В некоторых примерах модуль 44 компенсации движения может определять, что информация о движении доступна для соответствующего блока, когда весь соответствующий блок предсказан без использования интра-предсказания, но недоступна, если по меньшей мере часть соответствующего блока предсказана с использованием интра-предсказания.

Аналогичным образом в некоторых примерах модуль 44 компенсации движения может определять, какой из двух потенциальных кандидатов ATMVP должен использоваться в качестве кандидата ATMVP, в конечном счете добавляемого в список кандидатов. Например, модуль 44 компенсации движения может формировать первый временной вектор движения относительно текущего блока, который идентифицирует первого кандидата ATMVP в первой исходной картинке движения, то есть первой опорной картинке. Если информация о движении недоступна для первого кандидата ATMVP, модуль 44 компенсации движения может определять, доступна ли информация о движении для второго, другого кандидата ATMVP. Второй кандидат ATMVP может быть идентифицирован с использованием того же самого временного вектора движения, ссылающегося на вторую, другую опорную картинку, другого временного вектора движения, ссылающегося на ту же самую (то есть первую) опорную картинку, или другого временного вектора движения, ссылающегося на вторую, другую опорную картинку. Опорные картинки, которые должны быть проверены, как обсуждалось выше, могут находиться в порядке возрастания опорных индексов в списке опорных картинок. Аналогично, если используются различные временные векторы движения, то временные векторы движения могут быть выбраны в предопределенном порядке из временных векторов соседних блоков для текущего блока.

Кроме того, модуль 44 компенсации движения может определять, доступен ли вектор движения для под-PU в кандидате ATMVP для конкретного списка опорных картинок. Если это так, то вектор движения считается доступным для этого списка опорных картинок. В противном случае вектор движения считается недоступным для этого списка опорных картинок. Альтернативно, если вектор движения доступен

для другого списка опорных картинок, модуль 44 компенсации движения может изменять информацию о движении путем масштабирования вектора движения, чтобы указывать на целевую опорную картинку в первом списке опорных картинок, как обсуждалось выше.

Видеокодер 20 может затем выбрать одного из кандидатов из списка кандидатов (260). Например, видеокодер 20 может тестировать кодирование блока с использованием любого или всех из кандидатов в списке кандидатов. Дополнительно или альтернативно, модуль 42 оценки движения видеокодера 20 может выполнять поиск движения и определять вектор движения для блока и определять, следует ли кодировать вектор движения с использованием расширенного предсказания вектора движения (AMVP) или режима слияния. В примере согласно фиг. 10 предполагается, что видеокодер 20 сделал выбор кодировать информацию о движении с использованием режима слияния. В общем, видеокодер 20 (в частности, модуль 40 выбора режима) может определять, какой из кандидатов в списке кандидатов дает наилучшие характеристики скорости-искажения, и выбирать этого кандидата, чтобы использовать для предсказания блока.

Соответственно, видеокодер 20 может предсказать текущий блок с использованием выбранного кандидата (262). Т.е. модуль 44 компенсации движения может извлекать один или несколько опорных блоков, идентифицированных информацией о движении выбранного кандидата, и в некоторых примерах может интерполировать значения для дробных пикселей, если информация о движении имеет субпиксельную точность.

Видеокодер 20 может затем сформировать остаточный блок для текущего блока (264). Как обсуждалось выше, сумматор 50 может вычислять пиксельные разности между текущим блоком и предсказанным блоком, формируя остаточный блок. Видеокодер 20 может затем кодировать остаточную информацию остаточного блока и кодировать индекс слияния (266). Т.е. модуль 52 обработки преобразования может преобразовывать остаточный блок, чтобы получить коэффициенты преобразования, представляющие остаточную информацию. Затем модуль 54 квантования может квантовать коэффициенты преобразования. Модуль 56 энтропийного кодирования может затем энтропийно кодировать квантованные коэффициенты преобразования, а также синтаксические элементы, представляющие режим кодирования информации движения (режим слияния, в этом примере), и индекс слияния, представляющий выбранного кандидата из списка кандидатов.

Таким образом, способ, показанный на фиг. 10, представляет собой пример способа, включающего в себя формирование для текущего блока видеоданных списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков к текущему блоку и, после четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), кодирование индекса в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и кодирование текущего блока видеоданных с использованием информации о движении идентифицированного кандидата слияния.

На фиг. 11 представлена блок-схема последовательности операций, иллюстрирующая примерный способ добавления кандидата ATMVP к списку кандидатов во время процесса декодирования в соответствии с методами настоящего раскрытия. Способ согласно фиг. 11 описывается как выполняемый видеодекодером 30 (фиг. 1 и 3). Однако следует понимать, что другие устройства декодирования могут быть сконфигурированы для выполнения этого или аналогичного способа.

Сначала, видеодекодер 30 получает блок видеоданных, подлежащих кодированию (не показано на фиг. 11). Блок может включать в себя набор пространственных соседей, таких как показанные на фиг. 4(a) и фиг. 4(b). Модуль 72 компенсации движения может создавать список кандидатов, сначала добавляя левого пространственного кандидата в список кандидатов (270). Т.е. в отношении фиг. 4 (a), предполагая, что PU0 104A является блоком, подлежащим кодированию, видеодекодер 30 может сначала вставить пространственного соседа 108A в список кандидатов.

Затем модуль 72 компенсации движения может добавить вышеуказанного пространственного кандидата в список кандидатов (272). В отношении фиг. 4(a) видеодекодер 30 может вставить пространственного соседа 108B в список кандидатов.

Затем модуль 72 компенсации движения может добавить верхнего правого пространственного кандидата в список кандидатов (274). В отношении фиг. 4(a), видеодекодер 30 может вставить пространственного соседа 108C в список кандидатов.

Затем модуль 72 компенсации движения может добавить нижнего левого пространственного кандидата в список кандидатов (276). В отношении фиг. 4(a), видеодекодер 30 может вставить пространственного соседа 108D в список кандидатов.

Затем модуль 72 компенсации движения может добавить кандидата расширенного предсказания временного вектора движения (ATMVP) в список кандидатов (278). Как обсуждалось выше, кандидат ATMVP может представлять собой соответствующий блок, идентифицированный временным вектором, как показано и обсуждено, например, со ссылкой на фиг. 8 и 9. Кроме того, в некоторых примерах модуль 72 компенсации движения может сначала определить, доступен ли кандидат ATMVP. Например, блок компенсации движения может определять соответствующий блок для текущего блока в опорной

картинке и определять, доступна ли информация о движении для соответствующего блока. Модуль 72 компенсации движения может затем определить, что кандидат ATMVP (то есть соответствующий блок) доступен, когда информация о движении доступна для соответствующего блока. В некоторых примерах модуль 72 компенсации движения может определять, что информация о движении доступна для соответствующего блока, когда весь соответствующий блок предсказан без использования интра-предсказания, но недоступна, когда по меньшей мере часть соответствующего блока предсказана с использованием интра-предсказания.

Аналогичным образом, в некоторых примерах модуль 72 компенсации движения может определять, какого из двух потенциальных кандидатов ATMVP следует использовать в качестве кандидата ATMVP, который в конечном счете добавляется в список кандидатов. Например, модуль 72 компенсации движения может формировать первый временной вектор движения относительно текущего блока, который идентифицирует первого кандидата ATMVP в первой исходной картинке движения, то есть первой опорной картинке. Если информация о движении недоступна для первого кандидата ATMVP, модуль 72 компенсации движения может определить, доступна ли информация движения для второго, другого кандидата ATMVP. Второй кандидат ATMVP может быть идентифицирован с использованием того же самого временного вектора движения, ссылающегося на вторую, другую опорную картинку, другого временного вектора движения, ссылающегося на ту же самую (то есть первую) опорную картинку, или другого временного вектора движения, ссылающегося на вторую, другую опорную картинку. Опорные картинки, которые должны быть проверены, как обсуждалось выше, могут находиться в порядке возрастания опорных индексов в списке опорных картинок. Аналогично, если используются различные временные векторы движения, то временные векторы движения могут быть выбраны в predetermined порядке из временных векторов соседних блоков к текущему блоку.

Кроме того, модуль 72 компенсации движения может определить, доступен ли вектор движения для под-PU в кандидате ATMVP для конкретного списка опорных картинок. Если это так, то вектор движения считается доступным для этого списка опорных картинок. В противном случае вектор движения считается недоступным для этого списка опорных картинок. Альтернативно, если вектор движения доступен для другого списка опорных картинок, модуль 72 компенсации движения может изменять информацию о движении путем масштабирования вектора движения, чтобы указывать на целевую опорную картинку в первом списке опорных картинок, как обсуждалось выше.

Видеодекoder 30 может затем декодировать индекс слияния, чтобы выбрать одного из кандидатов из списка кандидатов (280). Более конкретно, модуль 70 энтропийного декодирования видеодекодера 30 может энтропийно декодировать один или несколько синтаксических элементов, представляющих, закодирована ли информация о движении текущего блока с использованием режима слияния, а также индекс слияния, представляющий выбранного кандидата из списка кандидатов.

Соответственно, видеодекoder 30 может предсказывать текущий блок с использованием выбранного кандидата (282). То есть модуль 72 компенсации движения может извлекать один или несколько опорных блоков, идентифицированных информацией о движении выбранного кандидата, и в некоторых примерах может интерполировать значения для мелких пикселей, если информация о движении имеет суб-пиксельную точность.

Видеодекoder 30 может также декодировать остаточный блок для текущего блока (284). В частности, модуль 70 энтропийного декодирования может декодировать квантованные коэффициенты преобразования, которые модуль 76 обратного квантования может обратно квантовать для формирования блока преобразования. Модуль 78 обратного преобразования может затем обратно преобразовывать блок преобразования для воспроизведения остаточного блока. Сумматор 80 может затем объединить предсказанный блок с остаточным блоком для декодирования текущего блока (286), в частности, путем восстановления текущего блока.

Таким образом, способ, показанный на фиг. 11, представляет собой пример способа, включающего в себя формирование, для текущего блока видеоданных, списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата из четырех соседних блоков к текущему блоку и, после четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), кодирование индекса в список кандидатов слияния, который идентифицирует кандидата слияния из множества кандидатов слияния в списке кандидатов слияния, и кодирование текущего блока видеоданных с использованием информации о движении идентифицированного кандидата слияния.

Должно быть понятно, что, в зависимости от примера некоторые действия или события любого из описанных здесь методов могут быть выполнены в другой последовательности, могут быть добавлены, объединены или вообще опущены (например, не все описанные действия или события необходимы для практической реализации этих методов). Более того, в некоторых примерах действия или события могут выполняться одновременно, например, посредством многопоточковой обработки, обработки прерываний или нескольких процессоров, а не последовательно.

В одном или нескольких примерах описанные функции могут быть реализованы в аппаратных

средствах, программном обеспечении, встроенном программном обеспечении или любой их комбинации. При реализации в программном обеспечении функции могут храниться или передаваться в виде одной или нескольких инструкций или кода на считываемом компьютером носителе и выполняться аппаратным процессором. Считываемые компьютером носители могут включать в себя считываемые компьютером носители хранения данных, которые соответствуют материальному носителю, такому как носители хранения данных, или среды связи, включающие в себя любой носитель, который облегчает перенос компьютерной программы из одного места в другое, например, согласно протоколу связи. Таким образом, считываемые компьютером носители, в общем, могут соответствовать (1) материальному считываемому компьютером носителю хранения данных, который является невременным, или (2) среде связи, такой как сигнал или несущая волна. Носителями хранения данных могут быть любые доступные носители, к которым может получить доступ один или несколько компьютеров или один или несколько процессоров для извлечения инструкций, кода и/или структур данных для реализации методов, описанных в настоящем раскрытии. Компьютерный программный продукт может включать в себя считываемый компьютером носитель.

В качестве примера, а не ограничения, такие считываемые компьютером носители хранения данных могут содержать RAM, ROM, EEPROM, CD-ROM или другое хранилище на оптических дисках, хранилище на магнитных дисках или другие магнитные запоминающие устройства, флэш-память или любой другой носитель, который можно использовать для хранения желательного программного кода в виде инструкций или структур данных, и к которому может получать доступ компьютер. Кроме того, любое соединение может надлежащим образом определяться как считываемый компьютером носитель. Например, если инструкции передаются с веб-сайта, сервера или другого удаленного источника с использованием коаксиального кабеля, оптоволоконного кабеля, витой пары, цифровой абонентской линии (DSL) или беспроводных технологий, таких как инфракрасная, радио и микроволновая, то коаксиальный кабель, волоконно-оптический кабель, витая пара, DSL или беспроводные технологии, такие как инфракрасная, радио и микроволновая, включаются в определение носителя. Однако следует понимать, что считываемые компьютером носители хранения и носители хранения данных не включают в себя соединения, несущие волны, сигналы или другие временные (переходные) носители, но вместо этого ориентированы на невременные материальные носители хранения. Магнитный диск (disk) и оптический диск (disc), как используется здесь, включают в себя компакт-диск (CD), лазерный диск, оптический диск, цифровой многофункциональный диск (DVD), гибкий диск и Blu-ray диск, где магнитные диски (disks) обычно воспроизводят данные магнитным способом, в то время как оптические диски (discs) оптически воспроизводят данные с помощью лазеров. Комбинации вышеизложенного также должны быть включены в объем считываемых компьютером носителей.

Инструкции могут выполняться одним или несколькими процессорами, такими как один или несколько процессоров цифровых сигналов (DSP), микропроцессоры общего назначения, специализированные интегральные схемы (ASIC), программируемые пользователем вентильные матрицы (FPGA) или другие эквивалентные интегральные или дискретные логические схемы. Соответственно, термин "процессор", используемый здесь, может относиться к любой из вышеперечисленных структур или любой другой структуре, подходящей для реализации описанных здесь методов. Кроме того, в некоторых аспектах, функциональность, описанная здесь, может быть предоставлена в рамках специализированных аппаратных средств и/или программных модулей, сконфигурированных для кодирования и декодирования или интегрированных в объединенный кодек. Кроме того, методы могут быть полностью реализованы в одной или нескольких схемах или логических элементах.

Методы настоящего раскрытия могут быть реализованы в самых разных приборах или устройствах, включая беспроводную трубку, интегральную схему (IC) или набор микросхем (например, чипсет). Различные компоненты, модули или устройства описаны в настоящем раскрытии, чтобы подчеркнуть функциональные аспекты устройств, сконфигурированных для выполнения раскрытых методов, но не обязательно требуют реализации различными аппаратными модулями. Скорее, как описано выше, различные модули могут быть объединены в аппаратный модуль кодека или обеспечены набором взаимодействующих аппаратных модулей, включая один или несколько процессоров, как описано выше, в сочетании с подходящим программным обеспечением и/или встроенным программным обеспечением.

Выше были описаны различные примеры. Эти и другие примеры находятся в пределах объема следующей формулы изобретения.

#### ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ кодирования видеоданных в соответствии с форматом видео, в котором картинку делят на множество блоков видеоданных, причем способ содержит

формирование для текущего блока видеоданных списка кандидатов слияния, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата, извлекаемых в предопределенном порядке из четырех соседних к текущему блоку блоков и, непосредственно после упомянутых четырех пространственных соседних кан-

дидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), при этом кандидат ATMVP указывает, что текущий блок должен быть предсказан с использованием блока, идентифицированного кандидатом ATMVP, который разделен на множество подблоков, причем каждый из множества подблоков имеет соответствующие наборы информации о движении;

кодирование индекса, который идентифицирует кандидата ATMVP из множества кандидатов слияния в списке кандидатов слияния; и

на основе индекса, идентифицирующего кандидата ATMVP, кодирование текущего блока видеоданных, причем кодирование текущего блока содержит кодирование подблоков текущего блока с использованием соответствующей информации о движении подблоков блока, идентифицированного кандидатом ATMVP.

2. Способ по п.1, в котором формирование списка кандидатов слияния содержит: определение для текущего блока соответствующего блока в опорной картинке; определение, доступна ли информация о движении для соответствующего блока; и формирование списка кандидатов слияния для включения кандидата ATMVP после определения того, что информация о движении доступна для соответствующего блока.

3. Способ по п.2, в котором определение, доступна ли информация о движении для соответствующего блока, содержит определение, является ли часть соответствующего блока интра-предсказанной.

4. Способ по п.1, в котором формирование списка кандидатов слияния содержит формирование кандидата ATMVP из репрезентативного набора информации о движении для соответствующего текущему блоку блока в опорной картинке.

5. Способ по п.4, в котором формирование кандидата ATMVP из репрезентативного набора информации о движении содержит формирование кандидата ATMVP из информации о движении для predetermined позиции соответствующего блока или из информации о движении для predetermined под-единицы предсказания (под-PU) соответствующего блока.

6. Способ по п.1, в котором формирование списка кандидатов слияния содержит: использование первого временного вектора движения относительно текущего блока для идентификации первого кандидата расширенного предсказания временного вектора движения (ATMVP) в первой исходной картинке движения;

когда первый кандидат ATMVP доступен, добавление первого кандидата ATMVP в список кандидатов слияния в качестве кандидата ATMVP;

когда первый кандидат ATMVP недоступен:

использование второго временного вектора движения относительно текущего блока для идентификации второго кандидата ATMVP во второй исходной картинке движения и

добавление второго кандидата ATMVP в список кандидатов слияния в качестве кандидата ATMVP.

7. Способ по п.6, в котором первый временной вектор движения и второй временной вектор движения содержат тот же самый временной вектор движения и в котором первая исходная картинка движения и вторая исходная картинка движения содержат различные исходные картинки движения.

8. Способ по п.6, в котором первый временной вектор движения и второй временной вектор движения содержат различные временные векторы движения.

9. Способ по п.6, дополнительно содержащий выбор первого временного вектора движения и второго временного вектора движения в соответствии с predetermined порядком из временных векторов соседних блоков.

10. Способ по п.1, в котором формирование списка кандидатов слияния содержит:

определение, доступен ли вектор движения для подблока кандидата ATMVP для списка X опорных картинок и

добавление кандидата ATMVP в список кандидатов после определения, что вектор движения доступен.

11. Способ по п.10, в котором, когда вектор движения недоступен для списка X опорных картинок, но доступен для списка Y опорных картинок, причем Y содержит список опорных картинок, отличный от списка X опорных картинок, устанавливают этот вектор движения доступным для списка X опорных картинок и масштабируют этот вектор движения к опорной картинке в списке X опорных картинок.

12. Способ по п.1, в котором кодирование индекса содержит декодирование индекса и в котором кодирование текущего блока содержит:

предсказание текущего блока с использованием информации о движении подблоков блока, идентифицированного кандидатом ATMVP, для формирования предсказанного блока;

декодирование остаточной информации для текущего блока и

декодирование текущего блока с использованием декодированной остаточной информации и предсказанного блока.

13. Способ по п.1, в котором кодирование индекса содержит кодировку индекса и в котором кодирование текущего блока содержит:

предсказание текущего блока с использованием информации о движении подблоков блока, идентифицированного кандидатом ATMVP, для формирования предсказанного блока;

формирование остаточного блока, представляющего разности между текущим блоком и предсказанным блоком; и

кодировку остаточной информации.

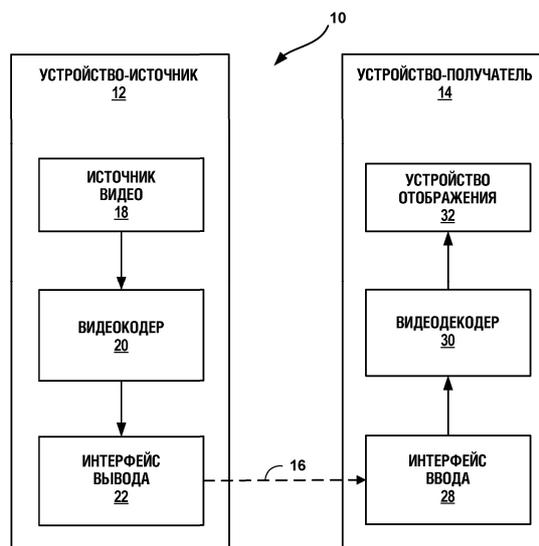
14. Устройство для кодирования видеоданных в соответствии с форматом видео, в котором картинку делят на множество блоков видеоданных, причем устройство содержит:

средство для формирования для текущего блока видеоданных списка кандидатов слиянием, включающего в себя множество кандидатов слияния, причем множество кандидатов слияния включают в себя четыре пространственных соседних кандидата, извлекаемых в predetermined порядке из четырех соседних к текущему блоку блоков и, непосредственно после упомянутых четырех пространственных соседних кандидатов, кандидата расширенного предсказания временного вектора движения (ATMVP), при этом кандидат ATMVP указывает, что текущий блок должен быть предсказан с использованием блока, идентифицированного кандидатом ATMVP, который разделен на множество подблоков, причем каждый из множества подблоков имеет соответствующие наборы информации о движении;

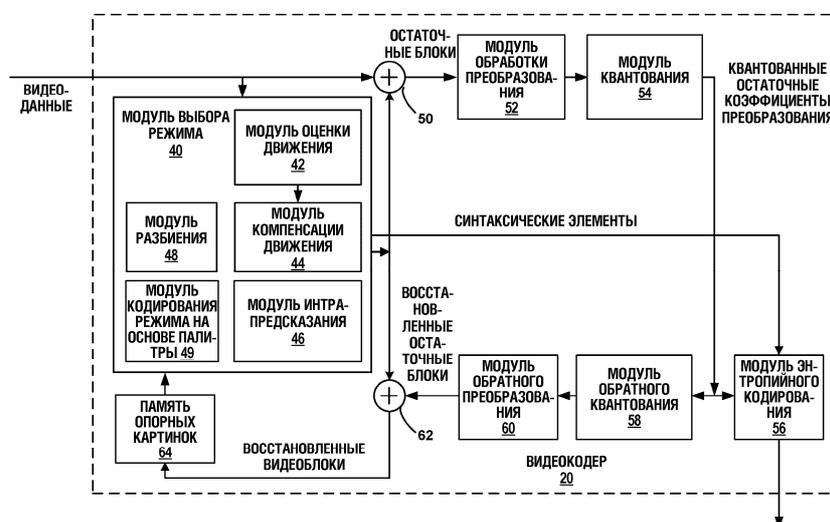
средство для кодирования индекса, который идентифицирует кандидата ATMVP из множества кандидатов слияния в списке кандидатов слияния; и

средство для кодирования текущего блока видеоданных, причем средство для кодирования текущего блока видеоданных содержит средство для кодирования подблоков текущего блока с использованием соответствующей информации о движении подблоков блока, идентифицированного кандидатом ATMVP, на основе индекса, идентифицирующего кандидата ATMVP.

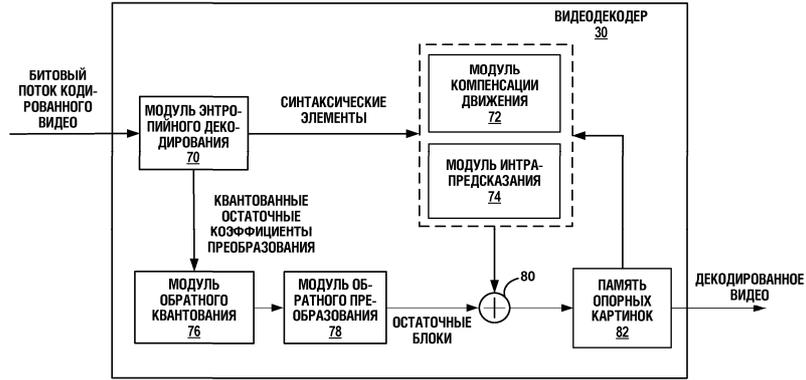
15. Считываемый компьютером носитель хранения данных, содержащий сохраненные на нем инструкции, которые при исполнении побуждают процессор выполнять способ по любому из пунктов с 1 по 13.



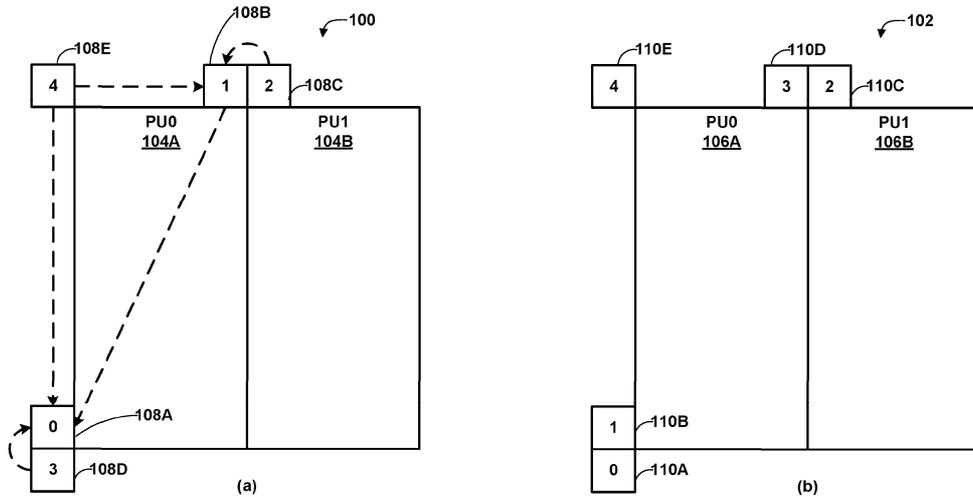
Фиг. 1



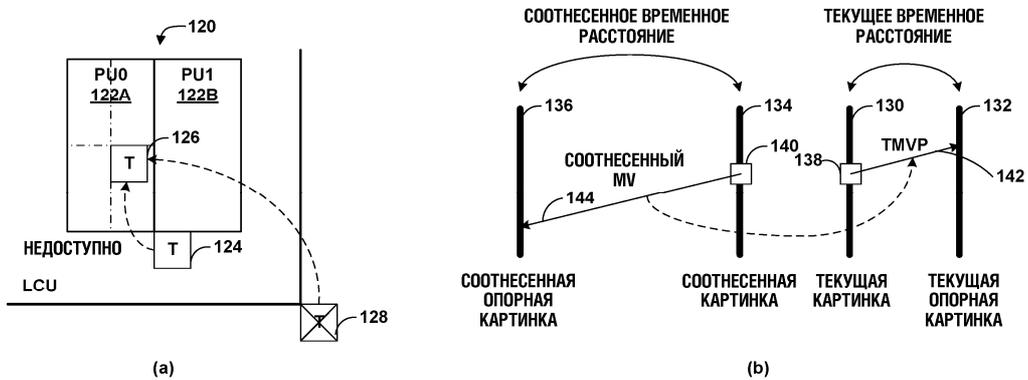
Фиг. 2



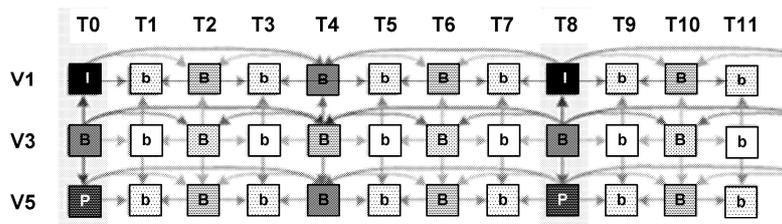
Фиг. 3



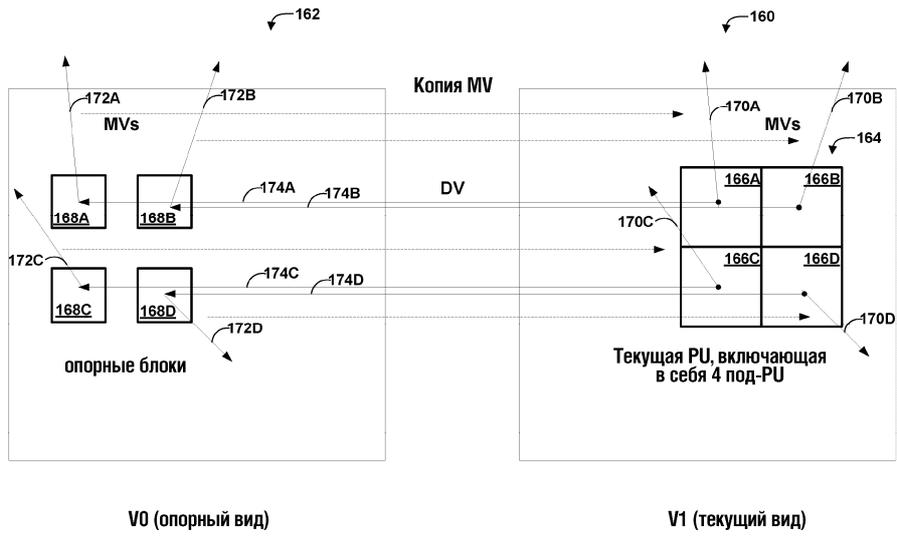
Фиг. 4



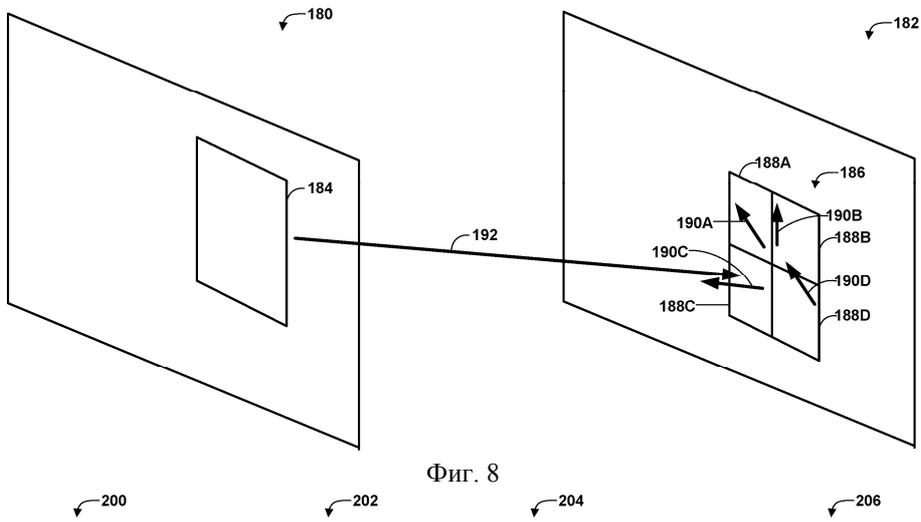
Фиг. 5



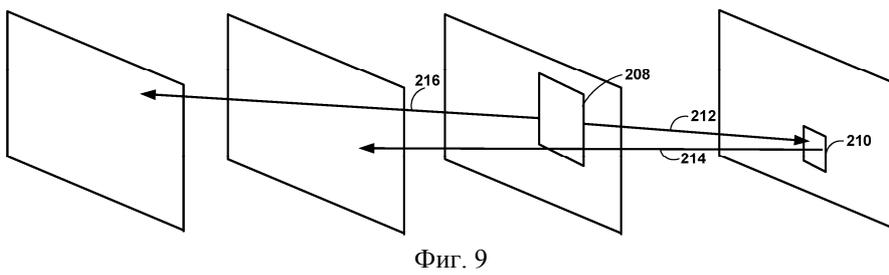
Фиг. 6



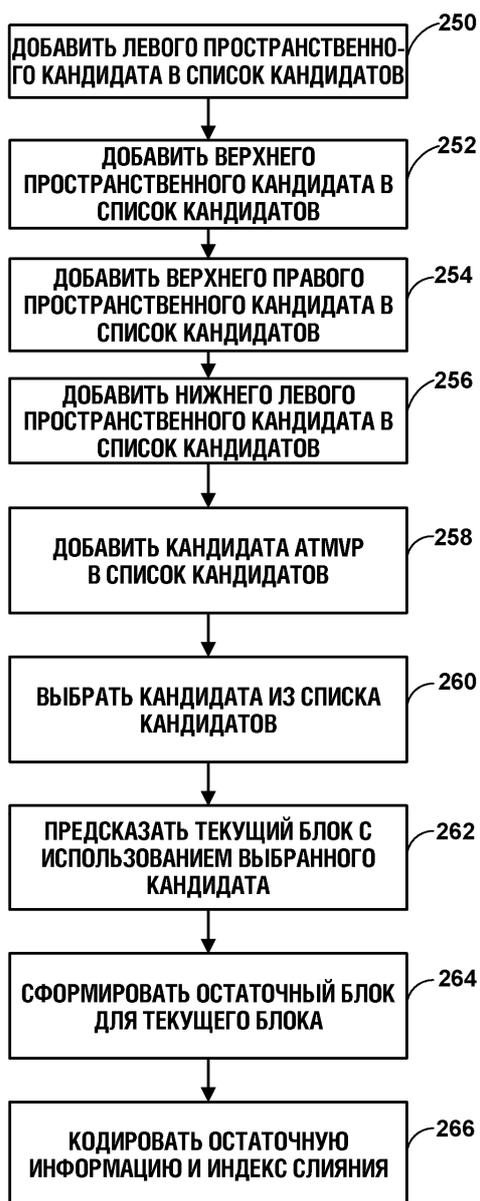
Фиг. 7



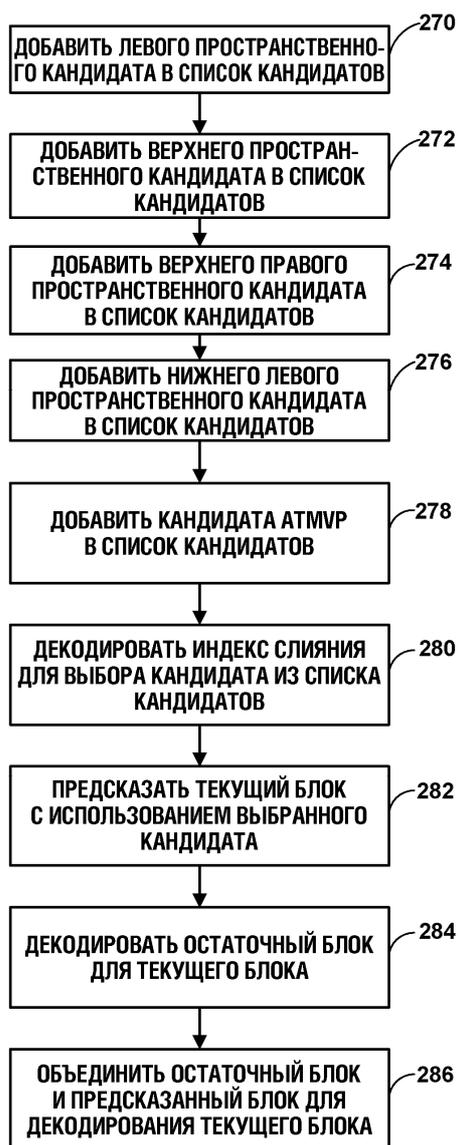
Фиг. 8



Фиг. 9



Фиг. 10



Фиг. 11

