РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 571 410**(13) **C2**

(51) МПК
H04N 19/105 (2014.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2014116246/08, 20.09.2012**

(24) Дата начала отсчета срока действия патента:
20.09.2012

Приоритет(ы):
(30) Конвенционный приоритет:
23.09.2011 US 61/538,787;
26.09.2011 US 61/539,433;
30.09.2011 US 61/542,034;
19.09.2012 US 13/622,944

(43) Дата публикации заявки: **27.10.2015** Бюл. № **30**

(45) Опубликовано: **20.12.2015** Бюл. № **35**

(56) Список документов, цитированных в отчете о поиске: **RU 2406253 C2, 27.08.2009. RU 2414092 C2, 10.03.2011. US 20090257497 A1, 15.10.2009. KR 20080041972 A, 14.05.2008. JP 2009060402 A, 19.03.2009.**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **23.04.2014**

(86) Заявка PCT:
US 2012/056368 (20.09.2012)

(87) Публикация заявки PCT:
WO 2013/043892 (28.03.2013)

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

(72) Автор(ы):
ВАН Е-Куй (US),
ЧЭНЬ Ин (US)

(73) Патентообладатель(и):
КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

RU 2571410 C2

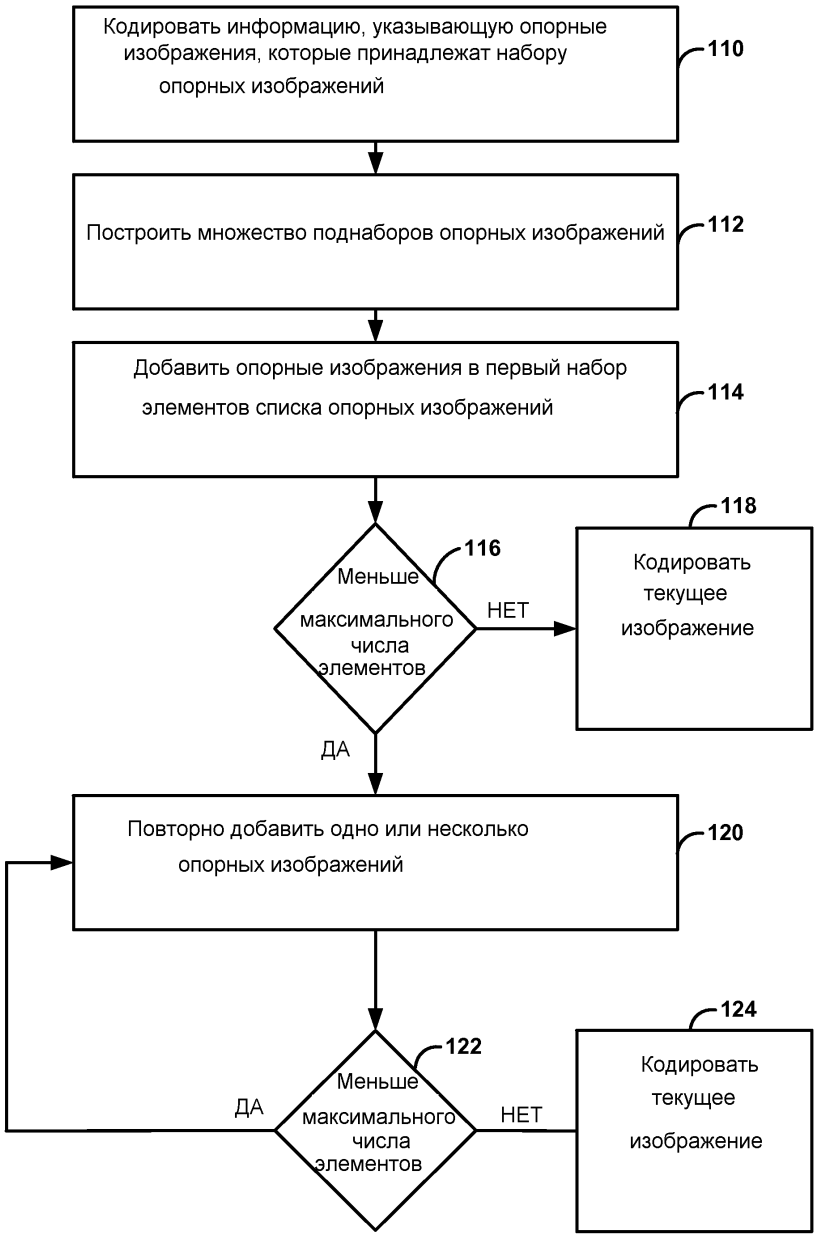
RU 2571410 C2

(54) ПОСТРОЕНИЕ СПИСКА ОПОРНЫХ ИЗОБРАЖЕНИЙ ДЛЯ ВИДЕОКОДИРОВАНИЯ

(57) Реферат:

Изобретение относится к средствам кодирования данных видео. Техническим результатом является повышение эффективности кодирования. Способ содержит кодирование информации, указывающей опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и изображений, следующих после текущего изображения в очередности декодирования, построение множества поднаборов опорных изображений, где каждое идентифицирует нуль или более опорных изображений; добавление опорных изображений в первый набор элементов в списке опорных изображений; определение, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, если нет, то многократное повторное добавление опорных изображений из поднаборов опорных

изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, кодирование текущего изображения на основании списка опорных изображений. 4 н. и 25 з.п. ф-лы, 10 ил.



ФИГ.7

RU 2571410 C2

RU 2571410 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 571 410**(13) **C2**

(51) Int. Cl.
H04N 19/105 (2014.01)

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2014116246/08, 20.09.2012**

(24) Effective date for property rights:
20.09.2012

Priority:
(30) Convention priority:
23.09.2011 US 61/538,787;
26.09.2011 US 61/539,433;
30.09.2011 US 61/542,034;
19.09.2012 US 13/622,944

(43) Application published: **27.10.2015** Bull. № **30**

(45) Date of publication: **20.12.2015** Bull. № **35**

(85) Commencement of national phase: **23.04.2014**

(86) PCT application:
US 2012/056368 (20.09.2012)

(87) PCT publication:
WO 2013/043892 (28.03.2013)

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

(72) Inventor(s):
VAN E-Kuj (US),
ChEhN' In (US)

(73) Proprietor(s):
KVEhLKOMM INKORPOREJTED (US)

RU 2571410 C2

RU 2571410 C2

(54) CREATION OF LIST OF REFERENCE IMAGES FOR VIDEO ENCODING

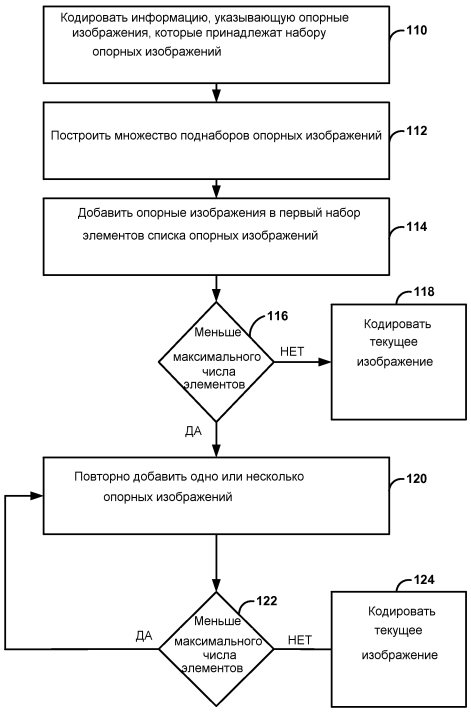
(57) Abstract:

FIELD: information technology.

SUBSTANCE: method comprises encoding information indicating reference images that can potentially be used for outer prediction of the current image and images following the current image in decoding sequence, the construction of the plurality of subsets of the reference images, where each one identifies zero or more reference images; adding the reference images to the first set of elements in the list of the reference images; determining whether the number of elements in the list of the reference images is equal to the maximum number of allowable elements in the list of the reference images, if not, multiple repeated adding of the reference images from the reference image subsets to the elements in the list of the reference images, that are after the first set of elements, the encoding of the current image based on the list of the reference images.

EFFECT: increased efficiency of encoding.

29 cl, 10 dwg



ФИГ.7

RU 2571410 C2

RU 2571410 C2

Данная заявка испрашивает приоритет по:
предварительной заявке на патент США №61/538787, поданной 23 сентября 2011;
предварительной заявке на патент США №61/539433, поданной 26 сентября 2011; и
предварительной заявке на патент США №61/542034, поданной 30 сентября 2011,
все содержание каждой из которых полностью включено в настоящий документ путем ссылки.

ОБЛАСТЬ ТЕХНИКИ

Данное раскрытие относится к видеокодированию и более конкретно - к способам кодирования видеоданных.

УРОВЕНЬ ТЕХНИКИ

Возможности цифрового видео могут быть включены в широкий спектр устройств, включая цифровые телевизоры, цифровые системы прямого вещания, устройства беспроводной связи, персональные цифровые ассистенты (PDA), ноутбуки или настольные компьютеры, планшетные компьютеры, устройства чтения электронных книг, цифровые фотоаппараты, устройства цифровой записи, проигрыватели цифровых данных, устройства видеоигр, игровые приставки, сотовые или спутниковые радио-телефоны, так называемые “умные телефоны,” видео устройства видеоконференц-связи, устройства потокового видео, и т.п.. Устройства цифрового видео реализуют способы сжатия видеоинформации, такие как описанные в стандартах, определенных стандартами Экспертной группой по вопросам движущегося изображения MPEG 2, MPEG 4, стандартами Международного союза электросвязи - сектора телекоммуникаций (ITU-T) H.263, ITU-T H.264/MPEG-4, Часть 10, Усовершенствованное кодирование видеоизображения (AVC), разрабатываемого в настоящее время стандарта высокоэффективного видеокодирования (HEVC), и расширения таких стандартов. Видео устройства могут передавать, принимать, кодировать, декодировать, и/или хранить информацию цифрового видео более эффективно путем реализации таких способов сжатия видео.

Способы сжатия видео выполняют пространственное (внутри картинки) предсказание и/или временное (внешнее для картинки) предсказание, чтобы уменьшить или удалить избыточность, присущую видео последовательностям. Для основанного на блоках кодирования, видео «слайс» (slice) (то есть видео изображение или часть видео изображения) может быть разделен на видеоблоки, которые могут также именоваться древовидными блоками, древовидными блоками кодирования (CTB), древовидные модули (блоки) кодирования (CTU), модули кодирования (CU) и/или узлы кодирования. Видеоблоки в слайсе с внутренним кодированием (I) изображения кодируются (сжимаются) с использованием пространственного предсказания по отношению к опорным выборкам в соседних блоках в том же изображении. Видео блоки в внешне-кодированном (P или B) слайсе изображения могут использовать пространственное предсказание относительно опорных выборок в соседних блоках в том же изображении или временное предсказание относительно опорных выборок в других опорных изображениях. Изображения могут именоваться кадрами, и опорные изображения могут именоваться опорными кадрами.

Пространственное или временное предсказание имеет следствием предсказанный блок для блока, подлежащего кодированию. Остаточные данные представляют пиксельные разности между исходным блоком, который подлежит кодированию, и предсказанным блок. Внешне кодированный блок кодируется согласно вектору движения, который указывает на блок опорных выборок, формирующих предсказанный блок, и остаточные данные, указывающие разность между кодированным блоком и

предсказанным блоком. Внутри кодированный блок кодируется согласно режиму внутрикадрового кодирования и остаточным данным. Для дополнительного сжатия остаточные данные могут быть преобразованы из пиксельной области в область преобразования, имея следствием остаточные коэффициенты преобразования, которые затем могут квантоваться. Квантованные коэффициенты преобразования, первоначально организованные в виде в двумерного массива, могут сканироваться, чтобы создать одномерный вектор коэффициентов преобразования, и применено энтропийное кодирование может применяться, чтобы добиться еще большего сжатия.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

В общем, данное раскрытие описывает способы, относящиеся к получению набора опорных изображений для использования в видеокодировании. Например, набор опорных изображений может составлять комбинацию из множества поднаборов опорных изображения. Каждое из поднаборов опорных изображений может идентифицировать некоторое количество потенциальных опорных изображений, но меньшее, чем все потенциальные опорные изображения. В примерных способах, описанных в этом раскрытии, кодер видео (кодер или декодер) может строить множественные списки, так что каждый включает в себя идентификаторы поднабора потенциальных опорных изображений. На основе этих множественных списков кодер видео может построить множество поднаборов опорных изображений, что имеет результатом получение кодером видео набора опорных изображений.

В дополнение к способам, относящимся к получению набора опорных изображений, это раскрытие описывает упрощенные способы инициализации списка опорных изображений. Такая инициализация списка опорных изображений может устранить необходимость в переупорядочении опорных изображений. Например, если модификация списка опорных изображений не требуется, то начальные списки опорных изображений могут образовывать окончательные списки опорных изображений, и могут не требовать последующего переупорядочения. Способы также могут быть направлены на построение списка опорных изображений некоторым образом, где кодер видео многократно добавляет опорные изображения к списку опорных изображений до тех пор, пока число элементов списка опорных изображений не будет равно максимальному допустимому числу элементов.

В некоторых примерах способы направлены на модификацию списка опорных изображений. Например, кодер видео может модифицировать начальный список опорных изображений путем обращения к одному или более поднаборам опорных изображений и включения одного или более изображений в поднаборах опорных изображений в список опорных изображений после построения начального списка опорных изображений.

В некоторых примерах кодер видео может выполнять управление буфером декодированных изображений (DPB). В этих примерах кодер видео может удалять декодированные изображения из DPB, если декодированное изображение не относится к набору опорных изображений. В некоторых случаях кодер видео может удалять декодированное изображение до кодирования текущего изображения.

В одном примере раскрытие описывает способ кодирования видеоданных, который включает в себя кодирование информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений,

следующих после текущего изображения в очередности декодирования. Способ также включает в себя построение множества поднаборов опорных изображений, что каждое идентифицирует нуль или более опорных изображений набора опорных изображений, и кодирование текущего изображения на основании множества поднаборов опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя кодер видео, который сконфигурирован для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Кодер видео также сконфигурирован, для построения множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений из набора опорных изображений, и для кодирования текущего изображения на основании множества поднаборов опорных изображений.

В одном примере раскрытие описывает считываемый компьютером носитель с наличием хранимых на нем инструкций(и), которые при исполнении предписывают процессору устройства кодирования видеоданных кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Инструкции также предписывают процессору строить множество поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, и кодировать текущее изображение на основании множества поднаборов опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя средство для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Устройство также включает в себя средство для построения множества поднаборов опорных изображений, идентифицирующих каждое нуль или более опорных изображений набора опорных изображений, и средство для кодирования текущего изображения на основании множества поднаборов опорных изображений.

В одном примере раскрытие описывает способ кодирования видеоданных, способ включает в себя кодирование информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Способ также

содержит построение множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, добавление опорных изображений из первого поднабора множества поднаборов опорных изображений, за которыми следуют опорные изображения из второго поднабора множества поднаборов опорных изображений, и за которыми следуют опорные изображения из третьего поднабора множества поднаборов опорных изображений, в список опорных изображений при условии, что число элементов списка опорных изображений не больше максимального числа допустимых элементов опорного списка, и кодирование текущего изображения на основании списка опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя кодер видео, сконфигурированный для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Кодер видео также сконфигурирован для построения множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, добавления опорных изображений из первого поднабора множества поднаборов опорных изображений, за которыми следуют опорные изображения из второго поднабора множества поднаборов опорных изображений, и за которыми следуют опорные изображения из третьего поднабора множества поднаборов опорных изображений, в список опорных изображений при условии, что число элементов списка опорных изображений не больше максимального числа допустимых элементов опорного списка, и кодирования текущего изображения на основании списка опорных изображений.

В одном примере раскрытие описывает считываемый компьютером носитель с наличием хранимых на нем инструкций, которые при исполнении предписывают процессору устройства кодирования видеоданных кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Инструкции также предписывают процессору строить множество поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, добавлять опорные изображения из первого поднабора множества поднаборов опорных изображений, за которыми следуют опорные изображения из второго поднабора множества поднаборов опорных изображений, и за которыми следуют опорные изображения из третьего поднабора множества поднаборов опорных изображений, в список опорных изображений при условии, что число элементов списка опорных изображений не больше максимального числа допустимых элементов опорного списка, и кодировать текущее изображение на основании списка опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя средство для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом

примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Устройство также включает в себя средство для построения множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, средство для добавления опорных изображений из первого поднабора множества поднаборов опорных изображений, за которыми следуют опорные изображения из второго поднабора множества поднаборов опорных изображений, и за которыми следуют опорные изображения из третьего поднабора множества поднаборов опорных изображений, в список опорных изображений при условии, что число элементов списка опорных изображений не больше максимального числа допустимых элементов опорного списка, и средство для кодирования текущего изображения на основании списка опорных изображений.

В одном примере раскрытие описывает способ кодирования видеоданных, способ включает в себя кодирование информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Способ также содержит построение множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, добавление опорных изображений из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений, определение, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, то многократное повторное добавления одного или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений, и кодирование текущего изображения на основании списка опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя кодер видео, сконфигурированный для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Кодер видео также сконфигурирован для построения множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, добавления опорных изображений из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений, определения, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов

в списке опорных изображений, если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, многократного повторного добавления одного или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений, и кодирования текущего изображения на основании списка опорных изображений.

В одном примере раскрытие описывает считываемый компьютером носитель с наличием хранимых на нем инструкций, которые при исполнении предписывают процессору устройства кодирования видеоданных кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Инструкции также предписывают процессору построить множество поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, добавлять опорные изображения из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений, определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, то многократно повторно добавлять одно или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений, и кодировать текущее изображение на основании списка опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя средство для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Устройство также включает в себя средство для построения множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, средство для добавления опорных изображений из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений, средство для определения, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, то средство для многократного повторного добавления одного или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора

элементов, до тех пор, пока число элементы в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений, и средство для кодирования текущего изображения на основании списка опорных изображений.

В одном примере раскрытие описывает способ кодирования видеоданных, способ включает в себя кодирование информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Способ также включает в себя построение множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, построение начального списка опорных изображений на основании построенных поднаборов опорных изображений, и если требуется модификация опорного изображения, идентификацию опорного изображения в, по меньшей мере, одном из построенных поднаборов опорных изображений, и добавление идентифицированного опорного изображения в текущий элемент в начальном (наборе) опорном изображения, чтобы построить модифицированный список опорных изображений. Способ дополнительно включает в себя кодирование текущего изображения на основании модифицированного списка опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя кодер видео, сконфигурированный, чтобы кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Видеокодер также сконфигурирован для построения множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, построения начального списка опорных изображений на основании построенных поднаборов опорных изображений, и если требуется модификация опорного изображения, то идентификации опорного изображения в, по меньшей мере, одном из построенных поднаборов опорных изображений, и добавления идентифицированного опорного изображения в текущий элемент начального опорного изображения, чтобы построить модифицированный список опорных изображений. Видеокодер также сконфигурирован для кодирования текущего изображения на основании модифицированного списка опорных изображений.

В одном примере раскрытие описывает считываемый компьютером носитель с наличием хранимых на нем инструкций, которые при исполнении предписывают процессору устройства кодирования видеоданных кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Инструкции также предписывают

процессору построить множество поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, построить начальный список опорных изображений на основании построенных поднаборов опорных изображений, и если требуется модификация опорного изображения, то идентифицировать опорное изображение в, по меньшей мере, одном из созданных поднаборов опорных изображений, и добавить идентифицированное опорное изображение в текущий элемент начального опорного изображения, чтобы построить модифицированный список опорных изображений. Инструкции также предписывают процессору кодировать текущее изображение на основании модифицированного списка опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя средство для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Устройство также включает в себя средство для построения множества поднаборов опорных изображений, каждый идентифицирует нуль или более опорных изображений набора опорных изображений, средство для построения начального списка опорных изображений на основании построенных поднаборов опорных изображений, и если требуется модификация опорного изображения то средство для идентификации опорного изображения в, по меньшей мере, одном из построенных поднаборов опорных изображений, и средство для добавления идентифицированного опорного изображения в текущий элемент начального опорного изображения, чтобы строить модифицированный список опорных изображений. Устройство также включает в себя средство для кодирования текущего изображения на основании модифицированного списка опорных изображений.

В одном примере раскрытие описывает способ кодирования видеоданных, способ включает в себя кодирование информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Способ включает в себя получение набора опорных изображений на основании кодированной информации, определение, является ли декодированное изображение, сохраненное в буфере декодированных изображений (DPB), не требуемым для вывода и не идентифицированным в наборе опорных изображений, если декодированное изображение не является требуемым для вывода и не является идентифицированным в наборе опорных изображений, то удаление декодированного изображения из DPB, и после удаления декодированного изображения, кодирование текущего изображения.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя кодер видео, сконфигурированный для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для

внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Кодер видео также сконфигурирован, чтобы получать набор опорных изображений на основании кодированной информации, определять, является ли декодированное изображение, сохраненное в буфере декодированных изображений (DPB), не требуемым для вывода и не идентифицированным в наборе опорных изображений, когда декодированное изображение не является требуемым для вывода и не идентифицировано в наборе опорных изображений, удалять декодированное изображение из DPB, и после удаления декодированного изображения кодировать текущее изображение.

В одном примере раскрытие описывает считываемый компьютером носитель с наличием хранимых на нем инструкций, которые при исполнении предписывают процессору устройства кодирования видеоданных кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Инструкции также предписывают процессору получать набор опорных изображений на основании кодированной информации, определять, является ли декодированное изображение, сохраненное в буфере декодированных изображений (DPB), не требуемым для вывода и не идентифицированным в наборе опорных изображений, если декодированное изображение не является требуемым для вывода и не идентифицировано в наборе опорных изображений, то удалять декодированное изображение из DPB, и после удаления декодированного изображения кодировать текущее изображение.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя средство для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений. В этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Устройство также включает в себя средство для получения набора опорных изображений на основании кодированной информации, средство для определения, является ли декодированное изображение, сохраненное в буфере декодированных изображений (DPB), не требуемым для вывода и не идентифицированным в наборе опорных изображений, если декодированное изображение не является требуемым для вывода и не идентифицировано в наборе опорных изображений, средство для удаления декодированного изображения из DPB и, после удаления декодированного изображения, средство для кодирования текущего изображения.

В одном примере раскрытие описывает способ кодирования видеоданных, способ включает в себя кодирование синтаксических элементов, указывающих долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров. В этом примере, одно или более долгосрочных опорных изображений-кандидатов находятся в наборе опорных изображений для текущего изображения. Кроме того, в этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего

изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Способ также включает в себя кодирование синтаксических элементов, которые указывают, какие долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров, находятся в наборе опорных изображений для текущего изображения, и построение, по меньшей мере, одного из множества поднаборов опорных изображений на основании указания, какие долгосрочные опорные изображения-кандидаты находятся в наборе опорных изображений для текущего изображения. В этом примере множество поднаборов опорных изображений образует набор опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных. Устройство включает в себя кодер видео, сконфигурированный для кодирования синтаксических элементов, указывающих долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров. В этом примере одно или более долгосрочных опорных изображений-кандидатов находятся в наборе опорных изображений для текущего изображения. Кроме того, в этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Кодер видео также сконфигурирован для кодирования синтаксических элементов, которые указывают, какие долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров, находятся в наборе опорных изображений для текущего изображения, и строить по меньшей мере один поднабор из множества поднаборов опорных изображений на основании указания, какие долгосрочные опорные изображения-кандидаты находятся в наборе опорных изображений для текущего изображения. В этом примере множество поднаборов опорных изображений образует набор опорных изображений.

В одном примере раскрытие описывает считываемый компьютером носитель с наличием хранимых на нем инструкций, которые при исполнении предписывают процессору устройства кодирования видеоданных кодировать синтаксические элементы, указывающие долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров. В этом примере, одно или несколько долгосрочных опорных изображений-кандидатов находятся в наборе опорных изображений для текущего изображения. Кроме того, в этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Инструкции также предписывают процессору кодировать синтаксические элементы, которые указывают, какие долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров, находятся в наборе опорных изображений для текущего изображения, и строить по меньшей мере один поднабор из множества поднаборов опорных изображений на основании указания, какие долгосрочные опорные изображения-кандидаты находятся в наборе опорных изображений для текущего изображения. В этом примере множество поднаборов опорных изображений образует набор опорных изображений.

В одном примере раскрытие описывает устройство для кодирования видеоданных.

Устройство включает в себя средство для кодирования синтаксических элементов, указывающих долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров. В этом примере, одно или несколько долгосрочных опорных изображений-кандидатов находятся в наборе опорных изображений для текущего изображения. Кроме того, в этом примере набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. Устройство также включает в себя средство для кодирования синтаксических элементов, которые указывают, какие долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров, находятся в наборе опорных изображений для текущего изображения, и средство для построения по меньшей мере одного поднабора из множества поднаборов опорных изображений на основании указания, какие долгосрочные опорные изображения-кандидаты находятся в наборе опорных изображений для текущего изображения. В этом примере множество поднаборов опорных изображений образует набор опорных изображений.

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

КРАТКОЕ ОПИСАНИЕ ФИГУР ЧЕРТЕЖЕЙ

Фиг. 1 - блок-схема, иллюстрирующая примерную систему кодирования и декодирования видео, которая может использовать способы, описанные в этом раскрытии.

Фиг. 2 - концептуальная схема, иллюстрирующая примерную видео последовательность, которая включает в себя множество изображений, которые кодируются и передаются.

Фиг. 3 - блок-схема, иллюстрирующая примерный видеокодер, который может осуществлять способы, описанные в этом раскрытии.

Фиг. 4 - блок-схема, иллюстрирующая примерный видеодекодер, который может осуществлять способы, описанные в этом раскрытии.

Фиг. 5 - блок-схема, иллюстрирующая примерную операцию получения набора опорных изображений.

Фиг. 6 - блок-схема, иллюстрирующая примерную операцию построения списка опорных изображений.

Фиг. 7 - блок-схема, иллюстрирующая другую примерную операцию построения списка опорных изображений.

Фиг. 8 - блок-схема, иллюстрирующая примерную операцию модифицирования начального списка опорных изображений.

Фиг. 9 - блок-схема, иллюстрирующая примерную операцию удаления декодированного изображения.

Фиг. 10 - блок-схема, иллюстрирующая примерную операцию определения, какие долгосрочные опорные изображения принадлежат набору опорных изображений для текущего изображения.

ПОДРОБНОЕ ОПИСАНИЕ

Способы данного раскрытия в общем направлены на управление опорными изображениями, которые используются для внешнего предсказания. Например, кодер видео (например, видеокодер или видеодекодер) включает в себя буфер декодированных изображений (DPB). DPB хранит декодированные изображения, включая опорные

изображения. Опорные изображения являются изображениями, которые потенциально могут быть использованы для внешнего предсказания изображения. Другими словами, кодер видео может предсказывать изображение, в течение кодирования (кодирования или декодирования) для этого изображения на основании одного или более опорных изображений, сохраненных в DPB.

Для эффективного использования DPB может быть точно определен процесс управления DPB, такой как процесс сохранения для декодированных изображений в DPB, процесс маркировки опорных изображений, процесс вывода и удаления декодированных изображений из DPB, и т.д. В общем, в некоторых существующих и разрабатывающихся стандартах кодирования видео управление DPB может включать в себя одно или более из следующих аспектов: идентификацию изображения и идентификацию опорного изображения, построение списка опорных изображений, маркировку опорного изображения, вывод изображения из DPB, вставку изображения в DPB и удаление изображения из DPB.

Для содействия пониманию последующее обеспечивает краткое описание, каким образом маркировка опорного изображения и построение списка опорных изображений могут происходить в соответствии с некоторыми стандартами кодирования видео. Некоторые из способов, описанных в этом раскрытии, обращаются к вопросам, которые могут присутствовать в маркировке опорного изображения, построении списка опорных изображений, и удалении и выводе изображения из DPB, с тем, чтобы повысить эффективность использования DPB.

Для маркировки опорного изображения, максимальное число, именуемое M (num_ref_frames), опорных изображений, используемых для внешнего предсказания, указывается в наборе параметров активной последовательности. Когда опорное изображение декодируется, оно маркируется “используемое для опорного”. Если декодирование опорного изображения обусловило более M изображений, маркированных “используемое для опорного", то по меньшей мере одно изображение должно маркироваться “неиспользуемое для опорного”. DPB процесс удаления затем удалит изображения, маркированные как “неиспользуемое для опорного” из DPB, если они не являются требуемыми для вывода также.

Когда изображение декодируется, оно может быть либо неопорным изображением, либо опорным изображением. Опорное изображение может быть долгосрочным опорным изображением или краткосрочным опорным изображением, и когда оно маркируется как “неиспользуемое для опорного”, оно может становиться более не требуемым для ссылки. В некоторых стандартах кодирования видео могут иметься операции маркировки опорного изображения, которые изменяют состояние опорных изображений.

Могут иметься два типа операций для маркировки опорного изображения: скользящее окно и адаптивное управление памятью. Операционный режим для маркировки опорного изображения может выбираться на основе изображения; тогда как операция скользящего окна может работать в виде очереди "первым пришел - первым обслужен" с постоянным числом краткосрочных опорных изображений. Другими словами, краткосрочные опорные изображения с самым ранним временем декодирования могут быть первыми, подлежащими удалению (маркированными как изображение, не используемое для опорного), неявным образом.

Адаптивное управление памятью однако удаляет краткосрочные или долгосрочные изображения явно. Оно также дает возможность переключения состояния краткосрочных и долгосрочных изображений, и т.д. Например, в адаптивном управлении

памятью, видеокодер может сигнализировать синтаксические элементы, которые указывают, какие изображения должны маркироваться используемыми в качестве опорных. Видеодекодер может принимать синтаксические элементы и маркировать изображения, как указано. В скользящем окне видеокодеру может не требоваться сигнализировать, какие изображения должны маркироваться используемыми в качестве опорных. Предпочтительнее видеодекодер может неявно (то есть без приема синтаксических элементов) определять, какие изображения должны маркироваться используемыми в качестве опорных, на основании того, какие изображения находятся внутри скользящего окна.

Кодеру видео также может быть поставлена задача построения списков опорных изображений, которые указывают, какие опорные изображения могут быть использованы с целями внешнего предсказания. Два из этих списков опорных изображений именуются List 0 и List 1, соответственно. Кодер видео во-первых использует способы построения «по умолчанию», чтобы строить List 0 и List 1 (например, предварительно сконфигурированные схемы построения для построения List 0 и List 1). Необязательно, после построения начальных List 0 и List 1, видеодекодер может декодировать синтаксические элементы, если присутствуют, которые инструктирует видеодекодер модифицировать начальный List 0 и List 1.

Видеокодер может сигнализировать синтаксические элементы, которые указывают идентификатор(ы) опорных изображений в DPB, и видеокодер может также сигнализировать синтаксические элементы, которые включают в себя индексы, в List 0, List 1, или обоих и List 0, и List 1, которые указывают, какое опорное изображение или изображения использовать для декодирования закодированного блока текущего изображения. Видеодекодер, в свою очередь, использует принятый идентификатор, чтобы идентифицировать значение или значения индекса для опорного изображения или опорных изображений, внесенных в List 0, List 1, или и List 0, и List 1. На основе значения(й) индекса, а также идентификатора(ов) опорного изображения или опорных изображений, видеодекодер извлекает опорное изображение или опорные изображения, или часть(и) таковых, из DPB, и декодирует закодированный блок текущего изображения на основании извлеченного опорного изображения или изображений и одного или более векторов движения, которые идентифицируют блоки внутри опорного изображения или изображений, которые используются для декодирования закодированного блока.

Например, построение списка опорных изображений для первого или второго списка опорных изображений для изображения с двунаправленным предсказанием включает в себя два этапа: инициализацию списка опорных изображений и модификацию списка опорных изображений (также именуемую переупорядочением списка опорных изображений). Инициализация списка опорных изображений может быть неявным механизмом, который помещает опорные изображения (находящиеся) в памяти опорных изображений (также известную как буфер декодированных изображений) в список на основании значений счетчика очередности изображения (POC) (Picture Order Count, упорядоченного по очередности отображения изображения). Механизм переупорядочения списка опорных изображений может модифицировать позицию изображения, которое было помещено в список в течение инициализации списка опорных изображений, в какую-либо новую позицию, или помещать какое-либо опорное изображение в память опорных изображений в какой-либо позиции, даже если изображение не принадлежит проинициализированному списку. Некоторые изображения после переупорядочения (модификации) списка опорных изображений, может быть

помещены в весьма далекую позицию в списке. Однако, если позиция изображения превышает число активных опорных изображений в списке, изображение не считается элементом окончательного списка опорных изображений. Число активных опорных изображений может сигнализироваться для каждого списка в заголовке слайса.

Способы, описанные в этом раскрытии, могут быть применимыми к различным стандартам кодирования видео. Примеры стандартов кодирования видео включают в себя ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 или MPEG-2 ISO/IEC 2 Visual, ITU-T H.263, MPEG-4 ISO/IEC Visual и ITU-T H.264 (также известный как MPEG-4 ISO/IEC AVC), включая его расширения Масштабируемое кодирование видео (SVC) и Многовидовое кодирование видео (MVC). Кроме того, имеется новый стандарт кодирования видео, а именно, высокоэффективное кодирование видеоизображений (HEVC), в настоящее время разрабатываемый Объединенной совместной группой по кодированию видео (JCT-VC) в составе Экспертной группы по кодированию видео ITU-T (VCEG) и Экспертной группой по вопросам движущегося изображения (MPEG) Международной комиссии по стандартизации и Международной электротехнической комиссии (ISO/IEC).

С целями лишь иллюстрации, способы описываются в контексте стандарта HEVC. Недавний Рабочий проект (WD) по HEVC, и называемый WD8 HEVC ниже в документе, является с 20 июля 2012 доступным по адресу http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip.

Как описано выше, способы, описанные в этом раскрытии, могут решить вопросы, которые могут присутствовать в существующих решениях для управления буфером декодированных изображений (DPB). В качестве одного примера, в некоторых примерных способах, описанных в этом раскрытии, может не требоваться маркировка опорных изображений как “неиспользуемый для опорного”. Например, описанные в этом раскрытии способы могут решать вопросы, связанные со способами управления DPB, которые возможно не очень подходят для временной масштабируемости, вопросы, связанные с непроизводительными издержками сигнализации долгосрочных опорных изображений, вопросы, связанные с эффективностью и сложностью инициализации и модификации списка опорных изображений. Способы, описанные в этом раскрытии, могут также решать вопросы, связанные с маркировкой “нет опорного изображения” для незаполненных элементов в списке опорных изображений в течение инициализации списка опорных изображений, вопросы, связанные с выводом декодированного изображения, вставкой в DPB и удалением из такового, а также вопросы, связанные с возможными значениями для значения счетчика очередности изображения (POC).

В соответствии со способами, описанными в этом раскрытии, списки опорных изображений строятся из набора опорных изображений. Набор опорных изображений определяется как набор опорных изображений, связанных с изображением, состоящее из всех опорных изображений, которые находятся до связанного изображения в очередности декодирования, которое может использоваться для внешнего предсказания блоков в связанном изображении или любом изображении, следующем после связанного изображения в очередности декодирования, например, до следующего изображения с мгновенным обновлением декодирования (IDR), или изображения с доступом с разорванной связью (BLA). Другими словами, опорные изображения в наборе опорных изображений могут требовать следующие характеристики: (1) они все находятся до текущего изображения в очередности декодирования, и (2) они могут быть использованы для внешнего предсказания текущего изображения и/или внешнего предсказания любого изображения, следующего за текущим изображением в очередности декодирования, и

в некоторых примерах, до следующего IDR изображения или BLA изображения. Могут быть другие альтернативные определения набора опорных изображений, которые обеспечиваются ниже.

В примерных способах, описанных в этом раскрытии, кодер видео может получить набор опорных изображений, и после такого выведения, кодер видео может строить списки опорных изображений. Например, только опорные изображения в наборе опорных изображений могут быть опорными изображениями-кандидатами, которые используются для построения списков опорных изображений.

Для построения набора опорных изображений, кодер видео может построить множество поднаборов опорных изображений. Объединение поднаборов опорных изображений может совместно образовывать набор опорных изображений. Например, видеокодер может явно сигнализировать, в кодированном битовом потоке, значения, которые дают возможность декодеру видео определять идентификаторы для опорных изображений, которые включены в набор опорных изображений. Например, идентификаторы опорных изображений могут быть счетчиками очередности изображения. Каждое изображение связано с одним счетчиком очередности изображения, называемым PicOrderCnt. PicOrderCnt указывает очередность вывода или очередность отображения соответствующего изображения относительно предыдущего IDR изображения в очередности декодирования, и, в некоторых других альтернативах, указывает позицию связанного изображения в очередности вывода относительно позиций очередности вывода для других изображений в той же кодированной видео последовательности.

PicOrderCnt может именоваться значением счетчика очередности изображения (POC). Значение POC может указывать очередность вывода или отображения изображения, и может использоваться для идентификации изображения. Например, в рамках кодированной видео последовательности, изображение с меньшим значением POC выводится или отображается ранее, чем изображение с более большим значением POC.

Видео декодер может определять идентификаторы для опорных изображений, и исходя из этих идентификаторов создавать множества поднаборов опорных изображений. Исходя из этих поднаборов опорных изображений видеодекодер может получить набор опорных изображений, как описано более подробно ниже. В некоторых примерах каждое из поднаборов опорных изображений включает в себя различные опорные изображения, в которых нет перекрытия опорных изображений в поднаборах опорных изображений. Таким образом, каждое из опорных изображений может находиться только в одном из поднаборов опорных изображений, и никаком другом поднаборе опорных изображений. Однако, аспекты данного раскрытия не следует считать таким образом ограниченными.

После определения идентификаторов (например, значения POC) опорных изображений набора опорных изображений или его поднаборах, видеодекодер может строить поднабора опорных изображений. Как описано более подробно ниже, видеодекодер может построить шесть поднаборов опорных изображений, хотя для видеодекодера может быть возможным построить большее или меньшее количество поднаборов опорных изображений.

Эти шесть поднаборов опорных изображений именуются: RefPicSetStCurr0 (текущее), RefPicSetStCurr1, RefPicSetStFoll0 (последующее), RefPicSetStFoll1, RefPicSetLtCurr и RefPicSetLtFoll. Поднабор опорных изображений RefPicSetStCurr0 может называться поднабором опорных изображений RefPicSetStCurrBefore (до), и поднабор опорных изображений RefPicSetStCurr1 может называться поднабором опорных изображений

RefPicSetStCurrAfter (после).

Поднаборы опорных изображений RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0 и RefPicSetStFoll1 могут идентифицировать краткосрочные опорные изображения. В некоторых примерах эти поднаборов опорных изображений могут идентифицировать краткосрочные опорные изображения на основании того, являются ли краткосрочные опорные изображения более ранними в очередности отображения или более поздними в очередности отображения, чем текущее кодируемое изображение, а также могут ли краткосрочные опорные изображения потенциально использоваться для внешнего предсказания текущего изображения и изображений, следующих после текущего изображения в очередности декодирования, или потенциально могут быть использованы для внешнего предсказания только изображений, следующих после текущего изображения в очередности декодирования.

Например, поднабор опорных изображений RefPicSetStCurr0 может включать в себя, и может включать в себя только идентификационную информацию, такую как значения POC, для всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода или отображения, чем текущее изображение, и которые потенциально могут быть использованы для опорного в внешнем предсказании для текущего изображения, и потенциально могут быть использованы для опорного в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования. Поднабор RefPicSetStCurr1 опорных изображений может включать в себя, и может включать в себя только идентификационную информацию всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода или отображения, чем текущее изображение, и которые потенциально могут быть использованы для ссылки в внешнем предсказании для текущего изображения, и может потенциально использоваться для ссылки в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования.

Поднабор опорных изображений RefPicSetStFoll0 может включать в себя, и может включать в себя только идентификационную информацию всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода или отображения, чем текущее изображение, которые потенциально могут быть использованы для ссылки в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования, и которые не могут быть использованы для ссылки в внешнем предсказании для текущего изображения. Поднабор опорных изображений RefPicSetStFoll1 может включать в себя, и может включать в себя только идентификационную информацию всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода или отображения, чем текущее изображение, которые потенциально могут быть использованы для ссылки в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования, и которые не могут быть использованы для ссылки в внешнем предсказании для текущего изображения.

Поднаборы RefPicSetLtCurr и RefPicSetLtFoll опорных изображений могут идентифицировать долгосрочные опорные изображения. В некоторых примерах эти поднаборы опорных изображений могут идентифицировать долгосрочные опорные изображения на основании того, являются ли долгосрочные опорные изображения в очередности отображения более ранними или поздними в очередности отображения, чем текущее кодируемое изображение.

Например, поднабор RefPicSetLtCurr опорных изображений может включать в себя,

и может включать в себя только идентификационную информацию всех долгосрочных опорных изображений, которые потенциально могут быть использованы для ссылки в внешнем предсказании для текущего изображения, и которые потенциально могут быть использованы для ссылки в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования. Поднабор опорных изображений RefPicSetLtFoll может включать в себя, и может включать в себя только идентификационную информацию всех долгосрочных опорных изображений, которые потенциально могут быть использованы для ссылки в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования, и которые не могут быть использованы для ссылки в внешнем предсказании для текущего изображения.

После построения поднаборов опорных изображений видеодекодер может упорядочивать поднабора опорных изображений в другом порядке для получения набора опорных изображений. В качестве одного примера, очередностью в наборе опорных изображений может быть RefPicSetStCurr0, RefPicSetSetCurr1, RefPicSetFoll0, RefPicSetFoll1, RefPicSetLtCurr и RefPicSetLtFoll. Однако, другое упорядочение поднаборов может быть возможным для получения набора опорных изображений. Например, в качестве другого примера, порядком в наборе опорных изображений может быть поднабор опорных изображений RefPicSetStCurr0, за которым следует (под) набор опорных изображений RefPicSetStCurr1, за которым следует поднабор опорных изображений RefPicSetLtCurr, за которым следует поднабор опорных изображений RefPicSetStFoll0, за которым следует поднабор опорных изображений RefPicSetFoll1, и за которым следует поднабор опорных изображений RefPicSetLtFoll.

В соответствии со способами, описанными в этом раскрытии, поднаборы RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr включают в себя все опорные изображения, которые могут быть использованы в внешнем предсказании блока в текущем изображении, и которые могут быть использованы в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования. Поднаборы RefPicSetStFoll0, RefPicSetStFoll1 и RefPicSetLtFoll включают в себя все опорные изображения, которые не используются в внешнем предсказании блока в текущем изображении, но могут быть использованы в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования.

Следует понимать, что шесть поднаборов опорных изображений описываются лишь с целью иллюстрации, и не должны считаться ограничивающими. В альтернативных примерах, может быть больше или меньше поднаборов опорных изображений. Такие поднабора опорных изображений, в этих альтернативных примерах, описываются более подробно ниже.

В некоторых способах, описанных в этих раскрытиях, видеодекодеру может не требоваться маркировать декодированные изображения являющимися "используемое для опорного", "неиспользуемое для опорного", "используемое для краткосрочного опорного", или "используемое для долгосрочного опорного". Предпочтительнее является ли декодированное изображение, сохраненное в DPB, требуемым для внешнего предсказания, указывается тем, включается ли оно в набор опорных изображений для текущего изображения. В альтернативных примерах может быть возможным, что видеодекодер маркирует декодированные изображения как "используемое для опорного", "неиспользуемое для опорного", "используемое для краткосрочного опорного", или "используемое для долговременного опорного". В этих примерах, после

декодирования изображения видеодекодером, оно является опорным изображением и маркированным как “используемое для опорного”. Затем, после вызова процесса для получения набора опорных изображений, все опорные изображения, сохраненные в DPB, но не включенные в набор опорных изображений для текущего изображения, маркируются “неиспользуемое для опорного”, перед возможным удалением декодированных изображений из DPB. Таким образом, является ли декодированное изображение, сохраненное в DPB, требуемым для внешнего предсказания, может указываться маркировкой его как “используемое для опорного”.

Как только видеодекодер получает набор опорных изображений из множества поднаборов опорных изображений, видеодекодер может строить списки опорных изображений (например, List 0 и List 1) на основе набора опорных изображений. Например, построение списков опорных изображений может включать в себя этап инициализации и возможно этап модификации. Путем получения набора опорных изображений описанным выше образом, видеодекодер способен повысить эффективность и уменьшить сложность для инициализации списка опорных изображений и модификации списка опорных изображений.

Могут иметься различные пути, которыми видеодекодер может строить списки опорных изображений. Способы, описанные в этом раскрытии, обеспечивают механизм, посредством которого видеодекодер может строить списки опорных изображений без необходимости переупорядочивать опорные изображения, подлежащие включению в (начальный) список опорных изображений. Например, видеодекодер может быть сконфигурирован для осуществления способа построения опорного списка «по умолчанию», в котором видеодекодер использует поднабор опорных изображений для построения начального списка опорных изображений. Затем, если модификация списка опорных изображений не требуется, окончательные списки опорных изображений могут быть такими же, как начальные списки опорных изображений, без необходимости какого-либо дополнительного переупорядочения списка опорных изображений.

В некоторых примерах, описанные в этом раскрытии способы могут относиться к построению списка опорных изображений таким образом, что не имеется незаполненных элементов. Например, способы могут многократно добавлять опорные изображения к списку опорных изображений из одного или более поднаборов опорных изображений. Например, после того, как видеодекодер добавляет опорные изображения из одного или более поднаборов опорных изображений для построения начального списка опорных изображений, видеодекодер может определять, является ли число элементов в списке опорных изображений меньше, чем максимальное допустимое число элементов. Если число элементов в списке опорных изображений меньше максимального числа для допустимого числа элементов, видеодекодер может повторно добавлять, по меньшей мере, одно из опорных изображений из одного из поднаборов опорных изображений, используемых для построения списка опорных изображений, в список опорных изображений. Это повторное добавление (также называемое переупорядочением) опорного изображения может происходить в другой позиции внутри списка опорных изображений, по сравнению с позицией, где опорное изображение было сначала добавлено видеодекодером.

Как используется в этом раскрытии, повторное построения списка или повторное добавление относится к добавлению снова (например, идентификацией снова) опорного изображения, которое было ранее добавлено (например, идентифицировано) в начальный список опорных изображений. Однако, при повторном добавлении опорного изображения, опорное изображение может находиться на двух различных элементах

в начальном списке опорных изображений. Другими словами, при повторном добавлении опорного изображения, могут быть два значения индекса в начальном списке опорных изображений, которые идентифицируют одно и то же опорное изображение.

В некоторых примерах способы, описанные в этом раскрытии, могут относиться к модифицированию начального списка опорных изображений. Например, видеодекодер может построить начальный список опорных изображений. Видеодекодер может определить, что модификация списка опорных изображений является необходимой, на основании синтаксических элементов, сигнализированных видеокодером в кодированном битовом потоке. Когда требуется модификация списка опорных изображений, видеодекодер может идентифицировать опорное изображение в, по меньшей мере, одном из построенных поднаборов опорных изображений. Видеодекодер может вносить в список (например, добавлять), идентифицированное опорное изображение в текущий элемент начального списка опорных изображений, чтобы построить модифицированный список опорных изображений. Видеодекодер может затем декодировать текущее изображение на основании модифицированного списка опорных изображений.

В некоторых примерах описанные в этом раскрытии способы могут относиться к выводу и удалению декодированных изображений из буфера декодированных изображений (DPB). примерные способы могут удалять декодированное изображение из DPB до кодирования текущего изображения. Например, примерные способы могут удалять декодированное изображение, если это декодированное изображение не идентифицировано в наборе опорных изображений для текущего изображения, и если это декодированное изображение не требуется для вывода (то есть оно не предназначалось для вывода, или оно предназначалось для вывода, но уже было выведено).

На Фиг. 1 показана блок-схема, иллюстрирующая примерную систему 10 кодирования и декодирования видео, которая может использовать способы, описанные в этом раскрытии. В общем, набор опорных изображений определяется как набор опорных изображений, связанных с изображением, состоящее из всех опорных изображений, находящихся до связанного изображения в очередности декодирования, которые могут быть использованы для внешнего предсказания связанного изображения или любого изображения, следующего после связанного изображения в очередности декодирования. В некоторых примерах опорные изображения, которые находятся до связанного изображения, могут быть опорными изображениями до следующего изображения с мгновенным обновлением декодирования (IDR), или изображением с доступом с разорванной связью (BLA). Другими словами, опорные изображения в наборе опорных изображений могут все находится до текущего изображения в очередности декодирования. Кроме того, опорные изображения в наборе опорных изображений могут быть использованы для внешнего предсказания текущего изображения и/или внешнего предсказания любого изображения, следующего за текущим изображением в очередности декодирования, до следующего IDR изображения или BLA изображения.

Могут быть другие альтернативные определения набора опорных изображений. Например, набор опорных изображений может быть набором опорных изображений, связанных с изображением, состоящим из всех опорных изображений, исключая само связанное изображение, которое может использоваться для внешнего предсказания связанного изображения или любого изображения, следующего после связанного изображения в очередности декодирования, и которые имеют значение temporal_id (временный идентификатор), меньшее или равное таковому для связанного изображения.

temporal_id может быть временным идентификационным значением. Временное идентификационное значение может быть иерархическим значением, которое указывает, какие изображения могут быть использованы для кодирования текущего изображения. В общем, изображение с конкретным значением temporal_id может возможно являться опорным изображением для изображений с равными или более большими значениями temporal_id, но не наоборот. Например, изображение со значением temporal_id 1 может возможно являться опорным изображением для изображений со значениями 1, 2, 3, …, temporal_id но не для изображения со значением 0 для temporal_id.

Нижнее значение temporal_id может также указывать низшую скорость отображения. Например, если видеодекодер декодировал только изображения со значением 0 для temporal_id, частотой воспроизведения может быть 7,5 кадров в секунду. Если видеодекодер декодировал только изображения со значениями 0 и 1 для temporal_id, частотой воспроизведения может быть 15 кадров в секунду, и т.д.

В качестве другого примера, набор опорных изображений может быть набором опорных изображений, связанных с изображением, состоящим из всех опорных изображений, исключая само связанное изображение, которые могут использоваться для внешнего предсказания связанного изображения или любого изображения, следующего после связанного изображения в очередности декодирования. В качестве еще одного примера, набор опорных изображений может быть определено в виде набора опорных изображений, связанных с изображением, состоящим из всех опорных изображений, возможно включающим в себя само связанное изображение, которые могут быть использованы для внешнего предсказания связанного изображения или любого изображения, следующего после связанного изображения в очередности декодирования. В качестве другого примера, набор опорных изображений может быть определено в виде набора опорных изображений, связанных с изображением, состоящим из всех опорных изображений, возможно включающим в себя само связанное изображение, которые могут быть использованы для внешнего предсказания связанного изображения или любого изображения, следующего после связанного изображения в очередности декодирования, и которые имеют значение temporal_id, меньшее или равное таковому для связанного изображения.

В качестве еще одного примера, в вышеупомянутых определениях набора опорных изображений, фраза “может использоваться для внешнего предсказания” заменяется на "используются для внешнего предсказания”. Хотя могут быть альтернативные определения набора опорных изображений, в этом раскрытии, примеры описываются с определением набора опорных изображений в виде являющегося набором опорных изображений, связанных с изображением, состоящим из всех опорных изображений, находящимися до связанного изображения в очередности декодирования, которые могут быть использованы для внешнего предсказания связанного изображения или любого изображения, следующего после связанного изображения в очередности декодирования.

Например, некоторые из опорных изображений набора опорных изображений являются опорными изображениями, которые потенциально могут быть использованы для внешнего предсказания блока для текущего изображения, а не изображений, следующих после текущего изображения в очередности декодирования. Некоторые из опорных изображений набора опорных изображений являются опорными изображениями, которые потенциально могут быть использованы для внешнего предсказания блока для текущего изображения, и блоков в одном или нескольких изображениях, следующих после текущего изображения в очередности декодирования.

Некоторые из опорных изображений набора опорных изображений являются опорными изображениями, которые потенциально могут быть использованы для внешнего предсказания блоков в одном или нескольких изображениях, следующих после текущего изображения в очередности декодирования, и не могут быть использованы для внешнего предсказания блока в текущем изображении.

Как используется в этом раскрытии, опорные изображения, которые потенциально могут быть использованы для внешнего предсказания, относятся к опорным изображениям, которые могут быть использованы для внешнего предсказания, но не обязательно должны использоваться для внешнего предсказания. Например, набор опорных изображений может идентифицировать опорные изображения, которые потенциально могут быть использованы для внешнего предсказания. Однако, это не означает, что все идентифицированные опорные изображения должны использоваться для внешнего предсказания. Предпочтительнее одно или несколько этих идентифицированных опорных изображений могут быть использованы для внешнего предсказания, но все не обязательно должны использоваться для внешнего предсказания.

Как показано на Фиг. 1, система 10 включает в себя исходное устройство (источник) 12, которое формирует кодированное видео для декодирования целевым устройством (получателем) 14. Исходное устройство 12 и целевое устройство 14 могут быть каждое примером устройства кодирования видео. Исходное устройство 12 может передавать кодированное видео на целевое устройство 14 через канал 16 связи или может сохранять кодированное видео на носителе 17 данных или файловом сервере 19, так что к кодированному видео может осуществлять доступ целевое устройство 14, если требуется.

Исходное устройство 12 и целевое устройство 14 могут содержать любое из широкого спектра устройств, включая беспроводной телефон, такой как так называемые "интеллектуальные" телефоны, так называемые "интеллектуальные" клавиатуры, или другие такие беспроводные устройства, оснащенные для беспроводной связи. Дополнительные примеры исходного устройства 12 и целевого устройства 14 включают в себя, но без ограничения указанным, цифровой телевизор, устройство в системе цифрового прямого вещания, устройство в системе беспроводного вещания, персональные цифровые ассистенты (PDA), ноутбук, настольный компьютер, планшетный компьютер, устройство чтения электронной книги, цифровой фотоаппарат, устройство цифровой записи, проигрыватель цифровых данных, устройство для видеоигр, игровую приставку, телефон сотовой радиосвязи, спутниковый радио-телефон, устройство видеоконференц-связи, и устройство потокового видео, устройство беспроводной связи или подобное.

Как указано выше, во многих случаях, исходное устройство 12 и/или целевое устройство 14 могут быть оборудованы для беспроводной связи. Следовательно, канал 16 связи может содержать канал беспроводной связи, канал проводной связи или комбинацию беспроводного и проводного каналов, подходящую для передачи кодированных видеоданных. Подобным образом к файловому серверу 19 может осуществлять доступ целевое устройство 14 через любое стандартное информационное соединение, включая соединение сети Интернет. Это может включать в себя беспроводной канал (например, соединение Wi-Fi), проводное соединение (например, цифровую абонентскую линию (DSL), кабельный модем, и т.д.), или комбинацию обоих, которая является подходящей для осуществления доступа к кодированным видеоданным, сохраненным на файловом сервере.

Способы по данному раскрытию, однако, могут применяться к кодированию видео в поддержке любого из множества мультимедийных приложений, таких как эфирное

телевизионное вещание, передачи кабельного телевидения, передачи спутникового телевидения, передачи потокового видео, например, через Интернет, кодирование (сжатие) цифрового видео для хранения на носителе данных, декодирование цифрового видео, сохраненного на носителе данных, или других приложений. В некоторых примерах система 10 может быть сконфигурирована для поддержки односторонней или двухсторонней передачи видео, чтобы поддерживать приложения, такие как потоковая передача видео, воспроизведение видео, видеовещание и/или видео телефония.

В примере по Фиг. 1 исходное устройство 12 включает в себя видео источник 18, видеокодер 20, модулятор/демодулятор (МОДЕМ) 22 и выходной интерфейс 24. В исходном устройстве 12 видео источник 18 может включать в себя источник, такой как устройство захвата/оцифровки видеоизображений, такое как видеокамера, архив видео, содержащий ранее записанное видео, интерфейс внешнего видеосигнала для приема видео от поставщика видеоконтента, и/или систему компьютерной графики для формирования данных компьютерной графики в качестве источника видеоизображения, или комбинацию таких источников. В качестве одного примера, если видео источник 18 является видеокамерой, исходное устройство 12 и целевое устройство 14 могут составлять так называемые телефоны с камерой или видеотелефоны. Однако, способы, описанные в этом раскрытии, могут быть применимыми к видеокодированию в целом, и могут применяться к приложениям беспроводной и/или проводной связи.

Захваченное, предварительно захваченное, или машинно-формируемое видеоизображение может кодироваться видеокодером 20. Кодированная видеоинформация может модулироваться модемом 22 согласно стандарту связи, такому как протокол беспроводной связи, и передаваться на целевое устройство 14 через выходной интерфейс 24. Модем 22 может включать в себя различные микшеры, фильтры, усилители или другие компоненты, предназначенные для модуляции сигнала. Выходной интерфейс 24 может включать в себя схемы, предназначенные для передачи данных, включая усилители, фильтры, и одну или несколько антенн.

Захваченное, предварительно захваченное или машинно-формируемое видеоизображение, которое кодируется видеокодером 20, может также сохраняться на носителе 17 данных или в файловом сервере 19 для более позднего использования. Носитель 17 данных может включать в себя диски по технологии Blu-ray, цифровые многофункциональные диски (DVD), ПЗУ на компакт-дисках (CD-ROM), флэш-память или любые другие подходящие цифровые носители для сохранения кодированного видеоизображения. К кодированному видеоизображению, сохраненному на носителе 17 данных, может затем осуществлять доступ целевое устройство 14 для декодирования и воспроизведения.

Файловый сервер 19 может быть любым типом сервера, способным сохранять кодированное видеоизображение и передавать это кодированное видеоизображение на целевое устройство 14. Примерные файловые серверы включают в себя веб-сервер (например, для веб-сайта), сервер с поддержкой протокола передачи файлов (FTP), устройства подключаемых к сети хранилищ данных (NAS), локальный накопитель на дисках, или любой другой тип устройства, способного сохранять кодированные видеоданные и передавать их на целевое устройство. Передача кодированных видеоданных из файлового сервера 19 может быть потоковой передачей, передачей загрузки из сети или их комбинацией. К файловому серверу 19 может осуществлять доступ целевое устройство 14 через посредство любого стандартного информационного соединения, включая Интернет-соединение. Это могут включать в себя беспроводной канал (например, соединение Wi-Fi), проводное соединение (например, DSL, кабельный

модем, подключение Ethernet, универсальную шину последовательной передачи данных (USB), и т.д.), или их комбинацию, которая является подходящей для осуществления доступа к кодированным видеоданным, сохраненным на файловом сервере.

Целевое устройство 14, в примере по Фиг. 1, включает в себя входной интерфейс 26, модем 28, видеодекодер 30 и устройство отображения 32. Входной интерфейс 26 целевого устройства 14 принимает информацию по каналу 16 в качестве одного примера, или от носителя 17 данных или файлового сервера 17 в качестве альтернативного примера, и модем 28, демодулирует информацию, чтобы создавать демодулированный битовый поток для видеодекодера 30. Демодулированный битовый поток может включать в себя различную синтаксическую информацию, сформированную видеокодером 20 для использования видеодекодером 30 в ходе декодирования видеоданных. Такой синтаксис может также быть включен кодированными видеоданными, сохраненными на носителе 17 данных или файловом сервере 19. В качестве одного примера, синтаксис может быть вложен в кодированные видеоданные, хотя аспекты данного раскрытия не следует считать ограниченными таким требованием. Синтаксическая информация, заданная видеокодером 20, которая также используется видеодекодером 30, может включать в себя синтаксические элементы, которые описывают характеристики и/или обработку для видеоблоков, таких как древовидные модули кодирования (CTU), древовидные блоки кодирования (CTB), модули предсказания (PU), модули кодирования (CU) или другие структурные единицы кодированного видео, например, видео слайсы, видеокадры, и видео последовательности или группы кадров (GOP). Каждый из видеокодера 20 и видеодекодера 30 может составлять часть соответственного кодера-декодера (КОДЕК), который способен кодировать (сжимать) или декодировать (восстанавливать) видеоданные.

Устройство 32 отображения может быть объединенным с целевым устройством 14 или внешним по отношению к нему. В некоторых примерах целевое устройство 14 может включать в себя интегрированный дисплей и также быть сконфигурировано, чтобы взаимодействовать через интерфейс с внешним устройством отображения. В других примерах целевое устройство 14 может быть устройством отображения. В общем, устройство 32 отображения отображает декодированные видеоданные пользователю и может содержать любое устройство из множества устройств отображения, такое как жидкокристаллическое устройство отображения (LCD), плазменное устройство отображения, устройство отображения на органических светодиодах (OLED), или другой тип устройства отображения.

В примере по Фиг. 1 канал 16 связи может содержать любую беспроводную или проводную среду передачи, такую как радиочастотный (RF) спектр или одна или несколько физических линий передачи, или любую комбинацию беспроводных и проводных сред. Канал 16 связи может составлять часть сети с пакетной передачей, такой как локальная сеть, региональная сеть или глобальная сеть, такая как сеть Интернет. Канал 16 связи обычно представляет любую подходящую среду передачи, или набор различных сред передачи, чтобы передавать видеоданные от исходного устройства 12 на целевое устройство 14, включающую в себя любую подходящую комбинацию проводных или беспроводных сред передачи. Канал 16 связи может включать в себя маршрутизаторы, коммутаторы, базовые станции или любое другое оборудование, которое может быть полезным, чтобы содействовать передаче от исходного устройства 12 на целевое устройство 14.

Видеокодер 20 и видеодекодер 30 могут работать в соответствии со стандартом сжатия видеоизображения, таким как включая ITU-T H.261, MPEG-1 ISO/IEC Visual,

ITU-T H.262 или MPEG-2 ISO/IEC Visual, ITU-T H.263, MPEG-4 ISO/IEC Visual и ITU-T H.264 (также известный как ISO/IEC MPEG-4 AVC), включая его расширения Масштабируемое кодирование видео (SVC) и Многовидовое кодирование видео (MVC). Кроме того, имеется новый стандарт кодирования видеоизображения, а именно, стандарт Высокоэффективного видеокодирования (HEVC), в настоящее время разрабатываемый Объединенной группой Сотрудничества по Видеокодированию (JCT-VC) ITU-T Video Coding Experts Group (VCEG) и Экспертной группы по цифровой записи видео и звука ISO/IEC (MPEG). Недавний Рабочий Проект (WD) HEVC, и называемый HEVC WD8 ниже в документе, является доступным с 20 июля 2012 по адресу http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip.

Способы по данному раскрытию, однако, не ограничиваются каким-либо конкретным стандартом кодирования. С целями лишь иллюстрации, способы описываются в соответствии со стандартом HEVC.

Хотя не показано на Фиг. 1, в некоторых аспектах, видеокодер 20 и видеодекодер 30 каждый могут быть интегрированными с кодером и декодером аудио, и могут включать в себя соответственные блоки мультиплексирования/демультиплексирования (MUX-DEMUX), или другое аппаратное и программное обеспечение, чтобы обрабатывать кодирование и аудио, и видео в общем потоке данных или раздельных потоках данных. Если применимо, блоки MUX-DEMUX могут соответствовать протоколу ITU H.223 для мультиплексоров(ания), или другим протоколам, таким как протокол передачи пользовательских дейтаграмм (UDP).

Видеокодер 20 и видеодекодер 30 каждый может быть реализован в виде любой схемы из множества подходящих схемных решений кодера, такой как один или большее число процессоров, включая микропроцессоры, цифровые процессоры сигналов (DSP), специализированные интегральные схемы (ASIC), программируемые вентильные матрицы (FPGA), дискретную логику, программное обеспечение, аппаратные средства, микропрограммное обеспечение или любые комбинации таковых. Когда способы реализуются частично программно, устройство может сохранять инструкции для программного обеспечения в подходящем, долговременном считываемом компьютером носителе и исполнять инструкции аппаратными средствами, использующими один или большее число процессоров, чтобы выполнять способ по данному раскрытию.

Каждый видеокодер 20 и видеодекодер 30 может быть включен в один или большее число кодеров или декодеров, любой из которых может быть интегрирован в виде составной части объединенного кодера/декодера (КОДЕКа) в соответственном устройстве. В некоторых случаях, видеокодер 20 и видеодекодер 30 могут обобщенно именоваться кодером видео, который кодирует информацию (например, изображения и синтаксические элементы). Кодирование информации может относиться к кодированию, когда кодер видео соответствует видеокодеру 20. Кодирование информации может относиться к декодированию, когда кодер видео соответствует видеодекодеру 30.

Кроме того, способы, описанные в этом раскрытии, могут относиться к видеокодеру 20 информации сигнализации. Когда видеокодер 20 сигнализирует информацию, способы по данному раскрытию обычно относятся к любому способу, которым видеокодер 20 предоставляет информацию. Например, когда видеокодер 20 сигнализирует синтаксические элементы на видеодекодер 30, это может означать, что видеокодер 20 передал синтаксические элементы на видеодекодер 30 через выходной интерфейс 24 и канал 16 связи, или что видеокодер 20 сохранил синтаксические элементы через выходной интерфейс 24 на носителе 17 данных и/или файловом сервере 19 для

возможного приема видеодекодером 30. Таким образом, сигнализацию от видеокодера 20 на видеодекодер 30 не следует интерпретировать в качестве требующей передачи от видеокодера 20, которая немедленно принимается видеодекодером 30, хотя это может быть возможным. Предпочтительнее сигнализацию от видеокодера 20 на видеодекодер 30 следует интерпретировать в качестве какого-либо способа, с помощью которого видеокодер 20 предоставляет информацию для возможного приема видеодекодером 30, либо непосредственно, либо через промежуточное хранилище (например, на носителе 17 данных и/или файловом сервере 19).

Видеокодер 20 и видеодекодер 30 могут быть сконфигурированы для осуществления примерных способов, описанных в этом раскрытии, предназначенных для получения набора опорных изображений. Например, видеодекодер 30 может активизировать процесс для получения набора опорных изображений один раз на одно изображение. Видеодекодер 30 может активизировать процесс, чтобы получить набор опорных изображений, после декодирования заголовка слайса, но до декодирования какого-либо модуля кодирования и до процесса декодирования для построения списка опорных изображений для слайса.

Как описано выше, набором опорных изображений является абсолютное описание опорных изображений, используемых в процессе декодирования текущего изображения, и будущих кодированных изображений в очередности декодирования до следующего изображения с мгновенным обновлением декодирования (IDR), или изображения с доступом с разорванной связью (BLA). В примерах, описанных в этом раскрытии, видеокодер 20 может явно сигнализировать значения, исходя из которых видеодекодер 30 может определять идентификаторы для опорных изображений, которые принадлежат набору опорных изображений. Сигнализация набора опорных изображений является явной в том смысле, что все опорные изображения, включенные в набор опорных изображений, приводятся в списке явно, кроме некоторых изображений, например, IDR изображений, синтаксические элементы набора опорных изображений не включаются в заголовок слайса, и набор опорных изображений получают являющимся пустым.

Могут иметься различные пути, которыми видеокодер 20 может сигнализировать синтаксические элементы в кодированном битовом потоке, который видеодекодер 30 может использовать для получения набора опорных изображений. Например, видеокодер 20 может сигнализировать синтаксические элементы в наборе параметров изображения (PPS), наборе параметров последовательности (SPS), заголовке изображения (если имеется), заголовке слайса, или любой комбинации таковых. С целью лишь иллюстрации, видеокодер 20 может сигнализировать синтаксические элементы, используя SPS, PPS и заголовок слайса, как описано более подробно.

Для получения набора опорных изображений, видеодекодер 30 может осуществлять процесс декодирования, чтобы определять идентификаторы для изображений, которые принадлежат набору опорных изображений. Видеодекодер 30 может затем построить множество поднаборов опорных изображений, где каждый из поднаборов идентифицирует нуль или более опорных изображений, которые принадлежат набору опорных изображений. Видеодекодер 30 может получить набор опорных изображений из синтаксически разобранных поднаборов опорных изображений. Например, видеодекодер 30 может вносить в список множество поднаборов опорных изображений в конкретном порядке, чтобы получать набор опорных изображений.

Могут иметься различные пути, которыми видеодекодер 30 может определять идентификаторы для изображений, принадлежащих набору опорных изображений. В общем, видеокодер 20 может сигнализировать значения, на основе которых

видеодекодер 30 может определять идентификаторы для изображений, включая изображения, которые принадлежат набору опорных изображений. Идентификаторами изображений могут быть PicOrderCnt (то есть значения счетчика очередности изображения (POC)). Как описано выше, значение POC может указывать очередность отображения или вывода изображения, где изображения с меньшими значениями POC отображаются ранее, чем изображения с более большими значениями POC. Значение POC для данного изображения может быть относительным по отношению к предшествующему изображению с мгновенным обновлением декодирования (IDR). Например, PicOrderCnt (то есть значение POC) для IDR изображения может быть 0, значением POC для изображения после IDR изображения в очередности отображения или вывода может быть 1, значением POC для изображения после изображения со значением 1 для POC в очередности отображения или вывода может быть 2, и т.д.

В соответствии со способами, описанными в этом раскрытии, когда текущее изображение не является IDR изображением, последующее может применять для получения POC для текущего изображения. Последующее предназначается для содействия пониманию, и не должно рассматриваться ограничивающим.

Например, можно рассмотреть переменную списка listD, которая включает в себя в качестве элементов значения PicOrderCnt (значения POC), связанные со списком изображений, включающим все из следующего: (1) первым изображением в списке является предшествующее IDR изображение в очередности декодирования, и (2), все другие изображения следуют в очередности декодирования после первого изображения в списке и либо предшествуют текущему изображению в очередности декодирования, либо являются текущим изображением. В этом примере текущее изображение включается в listD до активизации процесса получения набора опорных изображений. Кроме того, можно рассмотреть переменную списка listO, которая включает в себя элементы listD, отсортированные в порядке возрастания значений POC. В этом примере listO, может не содержать значение POC, у которого имеется значение, равное значению POC другого изображения.

В некоторых примерах значения POC могут быть ограничены диапазоном от $-2^{pocLen-1}$ до $2^{pocLen-1} - 1$ включительно. В этом примере pocLen может быть равным значению long_term_ref_pic_id_len_delta + long_term_ref_pic_id_delta_len_minus4 + 4. long_term_ref_pic_id_len_delta, и long_term_ref_pic_id_delta_len_minus4 могут быть синтаксическими элементами, которые видеодекодер 30 принимает в кодированном битовом потоке в виде части синтаксиса набора параметров изображения, как описывается более подробно ниже. В качестве другого примера, значения POC могут быть ограничены диапазоном от -2^{31} до $2^{31}-1$, включительно.

В качестве одного примера, видеодекодер 30 может принимать в кодированном битовом потоке (то есть битовом потоке, сигнализированном видеокодером 20) синтаксический элемент pic_order_cnt_lsb. Синтаксический элемент pic_order_cnt_lsb может указывать счетчик очередности изображения по модулю MaxPicOrderCntLsb для кодированного изображения. Длиной синтаксического элемента pic_order_cnt_lsb может быть log2_max_pic_order_cnt_lsb_minus4+4 битов. Значение pic_order_cnt_lsb может находиться в диапазоне от 0 до MaxPicOrderCntLsb-1, включительно. Видеодекодер 30 может принимать синтаксический элемент pic_order_cnt_lsb в синтаксисе заголовка слайса для текущего изображения, подлежащего декодированию.

Видеодекодер 30 может также принимать синтаксический элемент log2_max_pic_order_cnt_lsb_minus4 в кодированном битовом потоке, сигнализированном видеокодером

20. Видеодекодер 30 может принимать синтаксический элемент log2_max_pic_order_cnt _lsb_minus4 в наборе параметров последовательности. Значение log2_max_pic_order_cnt_lsb_ minus4 может находиться в диапазоне от 0 до 12 включительно. Синтаксический элемент log2_max_pic_order_cnt_lsb_minus4 может указывать значение переменной MaxPicOrderCntLsb, которую видеодекодер 30 использует в процессе декодирования для определения значения POC. Например:

$$MaxPicOrderCntLsb=2^{(log2\_max\_pic\_order\_cnt\_lsb\_minus4+4)}.$$

Исходя из этих принятых синтаксических элементов видеодекодер 30 может определять значение POC текущего изображения, как изложено ниже. Например, видеодекодер 30 может определить PicOrderCntMsb для текущего изображения. Значением POC для текущего изображения может определяться PicOrderCntMsb для текущего изображения плюс принятый pic_order_cnt_lsb для текущего изображения.

В последующем функция PicOrderCnt(picX) соответствует значению POC для изображения X. Функция DiffPicOrderCnt(picA,picB) соответствует PicOrderCnt(picA) минус PicOrderCnt(picB). В некоторых примерах кодированный битовый поток может не включать в себя данные, которые имеют результатом значения DiffPicOrderCnt (picA,picB), используемые в процессе декодирования, которые превышают диапазон от -2^{15} до $2^{15}-1$, включительно. Кроме того, пусть X будет текущим изображением, а Y и Z будут двумя другими изображениями в той же последовательности, где Y и Z считаются одинаковыми направлением очередности вывода от X, когда обе функции DiffPicOrderCnt(X,Y) и DiffPicOrderCnt(X,Z) являются положительными, или обе являются отрицательными. Кроме того, в некоторых примерах видеокодер 20 может назначать PicOrderCnt пропорциональным времени выборки соответствующего изображения относительно времени выборки предыдущего IDR изображения.

В качестве части процесса определения значения POC для текущего изображения видеодекодер 30 может определять переменные prevPicOrderCntMsb и prevPicOrderCntLsb. Например, если текущим изображением является IDR изображение, видеодекодер 30 может установить prevPicOrderCntMsb равным 0, и установить prevPicOrderCntLsb равным 0. Иначе (то есть когда текущее изображение не является IDR изображением), видеодекодер 30 может установить prevPicOrderCntMsb равным PicOrderCntMsb предыдущего опорного изображения в очередности декодирования с temporal_id, меньшим или равным текущему изображению, и установить prevPicOrderCntLsb равным значению pic_order_cnt_lsb предыдущего опорного изображения в очередности декодирования с temporal_id, меньшим или равным таковому для текущего изображения.

С помощью этих значений переменных и значений синтаксических элементов (например, значений prevPicOrderCntMsb, prevPicOrderCntLsb, pic_order_cnt_lsb и MaxPicOrderCntLsb) видеодекодер 30 может определять значение PicOrderCntMsb на основании этапов, изложенных в последующем псевдокоде. Следует понимать, что видеодекодер 30 может осуществлять этапы, изложенные в следующем псевдокоде, чтобы определять PicOrderCntMsb для каждого текущего изображения, которое используется для получения POC текущего изображения.

```
if((pic_order_cnt_lsb<prevPicOrderCntLsb) && ((prevPicOrderCntLsb-pic_order_cnt_lsb)>=
(MaxPicOrderCntLsb/2)))
    PicOrderCntMsb=prevPicOrderCntMsb+MaxPicOrderCntLsb
    else if((pic_order_cnt_lsb>prevPicOrderCntLsb) && ((pic_order_cnt_lsb−prevPicOrderCntLsb)
>(MaxPicOrderCntLsb/2)))
    PicOrderCntMsb=prevPicOrderCntMsb-MaxPicOrderCntLsb
```

Else

PicOrderCntMsb=prevPicOrderCntMsb

После определения PicOrderCntMsb для текущего изображения видеодекодер 30 может определять значение POC для текущего изображения на основании PicOrderCntMsb для текущего изображения и pic_order_cnt_lsb для текущего изображения. Видеодекодер 30 может определять значение POC для текущего изображения, как изложено ниже:

PicOrderCnt=PicOrderCntMsb+pic_order_cnt_lsb.

После декодирования изображения видеодекодер 30 может сохранять значение PicOrderCntMsb, значение pic_order_cnt_lsb и значение POC для этого изображения, включая каждое из опорных изображений, относящихся к набору опорных изображений, в буфере декодированных изображений (DPB) в видеодекодере 30. Таким образом, каждое изображение в DPB связано со значением POC, значением PicOrderCntMsb и значением pic_order_cnt_lsb.

Способы для определения значений POC для опорных изображений, включенных в набор опорных изображений для текущего изображения, описываются более подробно ниже. На основе определенных значений POC видеодекодер 30 может осуществлять процесс получения набора опорных изображений. Однако, до описания способа, которым видеодекодер 30 осуществляет процесс получения для набора опорных изображений, последующее представляет таблицы синтаксических элементов, которые видеодекодер 30 может принимать в кодированном битовом потоке, сигнализированном видеокодером 20. Например, видеокодер 20 может сигнализировать синтаксические элементы в следующих таблицах в кодированном битовом потоке, который принимает видеодекодер 30. Некоторые из этих синтаксических элементов были описаны выше. На основе синтаксических элементов видеодекодер 30 может определять значения POC для опорных изображений, включенных в набор опорных изображений, и кроме того, получать набор опорных изображений.

Например, в способах, описанных в этом раскрытии, последующие синтаксические структуры модифицированы относительно предшествующих стандартов кодирования видео: синтаксис набора параметров последовательности (SPS) в полезной нагрузке необработанной последовательности байтов (RBSP), seq_paramater_set_rbsq(), синтаксис RBSP для набора параметров изображения (PPS), pic_parameter_set_rbsp(), синтаксис заголовка слайса, slice_header(), и синтаксис модификации списка опорных изображений, ref_pic_list_modification(). Модификация списка опорных изображений описывается более подробно после описания получения набора опорных изображений и инициализации одного или более списков опорных изображений.

Кроме того, в соответствии со способами, описанными в этом раскрытии, следующие синтаксические структуры добавляются к кодированному битовому потоку: синтаксис списка краткосрочных опорных изображений, short_term_ref_pic_set(), и синтаксис списка долгосрочных опорных изображений, long_term_ref_pic_set(). Видеодекодер 30 может использовать синтаксис списка краткосрочных опорных изображений и синтаксис списка долгосрочных опорных изображений с целями построения поднаборов опорных изображений, на основе которых видеодекодер 30 получает набор опорных изображений.

Например, для определения видеодекодером 30 значений POC для опорных изображений, которые принадлежат набору опорных изображений, видеокодер 20 может сигнализировать идентификационную информацию опорного изображения, какой видеодекодер 30 используется для определения значений POC, в наборе параметров изображения и индекс в списке могут указываться в заголовке слайса. Однако, это представляет один примерный способ, которым видеокодер 20 может

сигнализировать такую идентификационную информацию опорного изображения.

В одном альтернативном примере видеокодер 20 может сигнализировать информацию опорного изображения в наборе параметров последовательности, и индекс к списку может указываться в идентификационной информации опорного изображения, каковое может снизить издержки сигнализации. В другом альтернативном примере кодер видео может сигнализировать информацию опорного изображения в новом типе набора параметров (например, набор параметров набора опорных изображений (RPSPS)), и идентификатор RPSPS, а также индекс к списку идентификационной информации опорного изображения могут указываться в идентификационной информации опорного изображения. Это может снизить издержки сигнализации, а также не увеличивать необходимость ряда наборов параметров изображения или наборов параметров последовательности. В других примерах видеокодер 20 может использовать любую комбинацию этих примерных способов, чтобы сигнализировать идентификационную информацию опорного изображения.

Таблица 1
Синтаксис RBSP для набора параметров последовательности

seq_parameter_set_rbsp( ) {	Дескриптор
profile_idc	u(8)
reserved_zero_8bits/* equal to 0 */	u(8)
level_idc	u(8)
seq_parameter_set_id	ue(v)
max_temporal_layers_minus1	u(3)
pic_width_in_luma_samples	u(16)
pic_height_in_luma_samples	u(16)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
pcm_bit_depth_luma_minus1	u(4)
pcm_bit_depth_chroma_minus1	u(4)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
max_num_ref_frames	ue(v)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
log2_min_pcm_coding_block_size_minus3	ue(v)
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
chroma_pred_from_luma_enabled_flag	u(1)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
temporal_id_nesting_flag	u(1)
inter_4x4_enabled_flag	u(1)
rbsp_trailing_bits( )	
}	

Элемент pic_width_in_luma_samples может указывать ширину каждого декодированного изображения в выборках сигнала яркости. Значение pic_width_in_luma_samples может находиться в диапазоне от 0 до 2^{16}-1, включительно.

Элемент pic_height_in_luma_samples может указывать высоту каждого декодированного изображения в выборках сигнала яркости. Значение

pic_height_in_luma_samples может находиться в диапазоне от 0 до 2^{16}-1, включительно.

Как указано в Таблице 1, видеодекодер 30 может принимать в наборе параметров последовательности (SPS) синтаксический элемент log2_max_pic_order_cnt_lsb_minus4. Как описано выше, значение log2_max_pic_order_cnt_lsb_minu4 может указывать значение переменной MaxPicOrderCntLsb, которую видеодекодер 30 использует в процессе декодирования для определения значения POC, где MaxPicOrderCntLsb = 2 (log2_max_pic_order_cnt_lsb_minus4+4)

.

Таблица 2
Синтаксис RBSP для набора параметров изображения

pic_parameter_set_rbsp( ) {	Дескриптор
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
entropy_coding_mode_flag	u(1)
num_short_term_ref_pic_sets_pps	ue(v)
for(i=0; i<num_short_term_ref_pic_sets_pps; i++)	
short_term_ref_pic_set( )	
long_term_ref_pics_present_flag	u(1)
if(long_term_ref_pics_present_flag) {	
long_term_ref_pic_id_delta_len_minus4	ue(v)
long_term_ref_pic_id_len_delta	ue(v)
num_long_term_ref_pics_pps	ue(v)
for(i=0; i< num_long_term_ref_pics_pps; i++)	
long_term_ref_pic_id_pps[i]	i(v)
}	
num_temporal_layer_switching_point_flags	ue(v)
for(i=0;i<num_temporal_layer_switching_point_flags; i++)	
temporal_layer_switching_point_flag[i]	u(1)
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
pic_init_qp_minus26/* relative to 26*/	se(v)
constrained_intra_pred_flag	u(1)
slice_granularity	u(2)
shared_pps_info_enabled_flag	u(1)
if(shared_pps_info_enabled_flag)	
if(adaptive_loop_filter_enabled_flag)	
alf_param( )	
if(cu_qp_delta_enabled_flag)	
max_cu_qp_delta_depth	u(4)
rbsp_trailing_bits( )	
}	

Элемент num_short_term_ref_pic_sets_pps указывает число синтаксических структур short_term_ref_pic_set(), включенных в набор параметров изображения. Значение num_short_term_ref_pic_sets_pps должно находиться в диапазоне от 0 до 32, включительно.

Элемент long_term_ref_pics_present_flag, равный 0, указывает, что долгосрочное опорное изображение не используется для внешнего предсказания какого-либо кодированного изображения, обращающегося к набору параметров изображения, и синтаксические элементы long_term_ref_pic_id_delta_len_minus4, long_term_ref_pic_id_len_delta и num_long_term_ref_pics_pps не присутствуют. Элемент long_term_ref_pics_present_flag, равный 1, указывает, что долгосрочные опорные изображения могут быть использованы для внешнего предсказания одного или более кодированных изображений, обращающегося к набору параметров изображения, и

синтаксические элементы long_term_ref_pic_id_delta_len_minus4, long_term_ref_pic_id_len_delta и num_long_term_ref_pics_pps присутствуют.

Элемент long_term_ref_pic_id_delta_len_minus4 плюс 4 указывает длину в битах синтаксических элементов long_term_ref_pic_id_delta_add_foll[i]. Значение long_term_ref_pic_id_delta_len_minus4 должно находиться в диапазоне от 0 до 12, включительно.

Элемент long_term_ref_pic_id_len_delta плюс long_term_ref_pic_id_delta_len_minus4 плюс 4 указывает длину в битах синтаксического элемента long_term_ref_pic_id_pps[i]. Значение long_term_ref_pic_id_len_delta может находиться в диапазоне от 0 до 28 - long_term_ref_pic_id_delta_len_minus4, включительно. Значение long_term_ref_pic_id_len_delta + long_term_ref_pic_id_delta_len_minus4 + 4 во всех наборах параметров изображения, обращающихся к одному конкретному набору параметров последовательности, может быть идентичным.

Элемент num_long_term_ref_pics_pps указывает число идентификаций долгосрочных опорных изображений, включенных в набор параметров изображения. Значение num_long_term_ref_pics_pps может находиться в диапазоне от 0 до 32, включительно.

Элемент long_term_ref_pic_id_pps[i] указывает i-ую идентификационную информацию долгосрочного опорного изображения, включенную в набор параметров изображения. Число битов, используемых для представления long_term_ref_pic_id_pps[i], может быть равным long_term_ref_pic_id_len_delta+long_term_pic_id_len_minus4+4.

Таблица 3
Синтаксис списка краткосрочных опорных изображений

short_term_ref_pic_set() {	Дескриптор
num_short_term_curr0	ue(v)
num_short_term_curr1	ue(v)
num_short_term_foll0	ue(v)
num_short_term_foll1	ue(v)
NumShortTerm=num_short_term_curr0+num_short_term_curr1+num_short_term_foll0 + num_short_term_foll1	
for(i=0; i<NumShortTerm; i++)	
short_term_ref_pic_id_delta_minus1[i]	ue(v)
}	

Синтаксис списка краткосрочных опорных изображений может предназначаться для краткосрочных изображений. Краткосрочное изображение может быть определено как опорное изображение, для которого идентификационная информация является включенной в синтаксическую структуру short_term_ref_pic_set() для кодированного изображения, либо включенной в заголовок(и) слайса, либо включенной в указываемое набор параметров изображения и ссылкой синтаксическим элементом short_term_ref_pic_set_idx в заголовке(ах) слайса. Синтаксические элементы заголовка слайса приведены в Таблице 4 ниже.

Элемент num_short_term_curr0 указывает число краткосрочных опорных изображений в RefPicSetStCurr0, когда синтаксическая структура short_term_ref_pic_set() используется для получения набора опорных изображений кодированного изображения, как описано ниже. Значение num_short_term_curr0 может находиться в диапазоне от 0 до max_num_ref_frames, включительно.

Элемент num_short_term_curr1 указывает число краткосрочных опорных изображений в RefPicSetStCurr1, когда синтаксическая структура short_term_ref_pic_set() используется для получения набора опорных изображений кодированного изображения, как описано ниже. Значение num_short_term_curr1 может находиться в диапазоне от 0 до max_num_ref_frames - num_short_term_curr0, включительно.

Элемент num_short_term_foll0 указывает число краткосрочных опорных изображений

в RefPicSetStFoll0, когда синтаксическая структура short_term_ref_pic_set() используется для получения набора опорных изображений кодированного изображения, как описано ниже. Значение num_short_term_foll0 может находиться в диапазоне от 0 до max_num_ref_frames - num_short_term_curr0 - num_short_term_curr1, включительно.

Элемент num_short_term_foll1 указывает число краткосрочных опорных изображений в RefPicSetStFoll1, когда синтаксическая структура short_term_ref_pic_set() используется для получения набора опорных изображений кодированного изображения, как описано ниже. Значение num_short_term_foll1 должно находиться в диапазоне от 0 до max_num_ref_frames-num_short_term_curr0-num_short_term_curr1-num_short_term_foll0, включительно.

Элемент short_term_ref_pic_id_delta_minus1[i] указывает идентификационную информацию i-го краткосрочного опорного изображения, включенного в синтаксическую структуру short_term_ref_pic_set().

Таблица 4
Синтаксис заголовка слайса

slice_header( ) {	Дескриптор
lightweight_slice_flag	u(1)
if(!lightweight_slice_flag) {	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
if(IdrPicFlag) {	
idr_pic_id	ue(v)
no_output_of_prior_pics_flag	u(1)
}	
pic_order_cnt_lsb	u(v)
if(!IdrPicFlag) {	
short_term_ref_pic_set_pps_flag	u(1)
if(short_term_ref_pic_set_pps_flag)	
short_term_ref_pic_set_idx	ue(v)
Else	
short_term_ref_pic_set( )	
if(long_term_ref_pics_present_flag)	
long_term_ref_pic_set( )	
}	
if(slice_type = = P || slice_type = = B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type = = B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
ref_pic_list_modification( )	
ref_pic_list_combination( )	
}	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
first_slice_in_pic_flag	u(1)
if(first_slice_in_pic_flag == 0)	
slice_address	u(v)
if(!lightweight_slice_flag) {	
slice_qp_delta	se(v)
if(sample_adaptive_offset_enabled_flag)	
sao_param()	

if(deblocking_filter_control_present_flag) {	
disable_deblocking_filter_idc	
if(disable_deblocking_filter_idc != 1) {	
slice_alpha_c0_offset_div2	
slice_beta_offset_div2	
}	
}	
if(slice_type = = B)	
collocated_from_l0_flag	u(1)
if(adaptive_loop_filter_enabled_flag) {	
if(!shared_pps_info_enabled_flag)	
alf_param( )	
alf_cu_control_param( )	
}	
}	
}	

Элемент no_output_of_prior_pics_flag указывает, каким образом предварительно декодированные изображения в буфере декодированных изображений обрабатываются после декодирования IDR изображения. Когда IDR изображение является первым IDR изображением в битовом потоке, значение no_output_of_prior_pics_flag может не влиять на процесс декодирования. Когда IDR изображение является не первым IDR изображением в битовом потоке, и значения pic_width_in_luma_samples или pic_height_in_luma_samples, или max_dec_frame_buffering, полученные из активного набора параметров последовательности, могут быть отличными от значения pic_width_in_luma_samples или pic_height_in_luma_samples, или max_dec_frame_buffering, полученные из набора параметров последовательности, активного для предшествующего изображения, равный 1 флаг no_output_of_prior_pics_flag может, но не обязательно, логически выводиться декодером независимо от фактического значения флага no_output_of_prior_pics_flag.

Элемент short_term_ref_pic_set_pps_flag, равный 1, указывает, что идентификационная информация набора краткосрочных опорных изображений, включенных в набор опорных изображений для текущего изображения, присутствует в указываемом наборе параметров изображения. Флаг short_term_ref_pic_set_pps_flag, равный 0, указывает, что идентификационная информация набора краткосрочных опорных изображений, включенных в набор опорных изображений для текущего изображения, не присутствует в указываемом наборе параметров изображения.

Элемент short_term_ref_pic_set_idx указывает индекс синтаксической структуры short_term_ref_pic_set(), включенной в указываемое набор параметров изображения, которое включает в себя идентификационную информацию набора краткосрочных опорных изображений набора опорных изображений для текущего изображения.

Переменные NumShortTermCurr0 и NumShortTermCurr1 указываются в виде:

NumShortTermCurr0=num_short_term_curr0

NumShortTermCurr1=num_short_term_curr1

Когда элементы num_short_term_curr0 и num_short_term_curr0 являются синтаксическими элементами с таким же именем, соответственно, в синтаксической структуре short_term_ref_pic_set(), или существуют в указываемом наборе параметров изображения и указываемый посредством short_term_ref_pic_set_idx, или непосредственно присутствует в заголовке слайса.

Элемент num_ref_idx_l0_active_minus1 указывает максимальный ссылочный индекс для списка 0 опорных изображений, который должен использоваться для декодирования

слайса.

Когда текущий слайс является слайсом P или B, и num_ref_idx_l0_active_minus1 не присутствует, num_ref_idx_l0_active_minus1 можно принять равным num_ref_idx_l0_default_active_minus1.

Значение элемента num_ref_idx_l0_active_minus1 может находиться в диапазоне от 0 до 15, включительно.

Элемент num_ref_idx_l1_active_minus1 указывает максимальный ссылочный индекс для списка опорных изображений 1, который должен использоваться для декодирования слайса.

Когда текущий слайс является слайсом P или B, и num_ref_idx_l1_active_minus1 не присутствует, num_ref_idx_l1_active_minus1 можно принять являющимся равным num_ref_idx_l1_default_active_minus1.

Значение num_ref_idx_l1_active_minus1 может находиться в диапазоне от 0 до 15, включительно.

Таблица 5
Синтаксис набора долгосрочных опорных изображений

long_term_ref_pic_set( ) {	Дескриптор
num_long_term_pps_curr	ue(v)
num_long_term_add_curr	ue(v)
num_long_term_pps_foll	ue(v)
num_long_term_add_foll	ue(v)
for(i=0; i<num_long_term_pps_curr+num_long_term_ pps_foll; i++)	
long_term_ref_pic_set_idx_pps[i]	ue(v)
for(i=0; i<num_long_term_add_curr+num_long_term _add_foll; i++)	
long_term_ref_pic_id_delta_add[i]	i(v)
}	

Синтаксис набора долгосрочных опорных изображений может предназначаться для долгосрочных изображений. Долгосрочное изображение может быть определено как опорное изображение, для которого идентификационная информация включается в синтаксическую структуру long_term_ref_pic_set() для кодированного изображения.

Элемент num_long_term_pps_curr указывает число всех долгосрочных опорных изображений, идентификационная информация которых включена в указываемое набор параметров изображения, и которые могут быть использованы для внешнего предсказания текущего изображения. Если num_long_term_pps_curr не присутствует, значение может приниматься равным 0. Значение num_long_term_pps_curr может находиться в диапазоне от 0 до max_num_ref_frames, включительно.

Элемент num_long_term_add_curr указывает число всех долгосрочных опорных изображений, идентификационная информация которых не включена в указываемое набор параметров изображения, и которые могут быть использованы для внешнего предсказания текущего изображения. Если num_long_term_add_curr не присутствует, значение может приниматься равным 0. Значение num_long_term_add_curr может находиться в диапазоне от 0 до max_num_ref_frames - num_long_term_pps_curr, включительно.

Переменная NumLongTermCurr задается как:

NumLongTermCurr=num_long_term_pps_curr + num_long_term_add_curr

Элемент num_long_term_pps_foll указывает число всех долгосрочных опорных изображений, идентификационная информация которых включена в указываемое набор параметров изображения, которые не используются для внешнего предсказания

текущего изображения, и которые могут быть использованы для внешнего предсказания какого-либо из изображений, следующих после текущего изображения в очередности декодирования. Если num_long_term_pps_foll не присутствует, значение может приниматься равным 0. Значение num_long_term_pps_foll может находиться в диапазоне от 0 до max_num_ref_frames, включительно.

Элемент num_long_term_add_foll указывает число всех долгосрочных опорных изображений, идентификационная информация которых не включается в указываемое набор параметров изображения, которые не используются для внешнего предсказания текущего изображения, и которые могут быть использованы для внешнего предсказания любого из следующих изображений в очередности декодирования. Если num_long_term_add_foll не присутствует, значение может приниматься равным 0. Значение num_long_term_add_foll может находиться в диапазоне от 0 до max_num_ref_frames - num_long_term_pps_foll, включительно.

Элемент long_term_ref_pic_set_idx_pps[i] указывает индекс к списку для идентификационной информации долгосрочного опорного изображения, включенной в указываемое набор параметров изображения, для i-го долгосрочного опорного изображения, наследуемый от указываемого набора параметров изображения к набору опорных изображений текущего изображения. Значение long_term_ref_pic_set_idx_pps [i] может находиться в диапазоне от 0 до 31, включительно.

Элемент long_term_ref_pic_id_delta_add[i] указывает идентификационную информацию долгосрочного опорного изображения для i-го долгосрочного опорного изображения, которая не наследуется от указываемого набора параметров изображения, но включается в набор опорных изображений текущего изображения. Число битов, используемой для представления long_term_ref_pic_id_add_curr[i], может быть равным long_term_pic_id_len_minus4+4.

С помощью вышеупомянутых сигнализированных или полученных значений (то есть значений из Таблиц 1-5), видеодекодер 30 может получить набор опорных изображений. Как описано выше, полученное набор опорных изображений может идентифицировать опорные изображения, которые потенциально могут быть использованы для кодирования/предсказания текущего изображения (то есть изображения, являющегося в текущий момент декодируемым), и изображений, которые следуют после текущего изображения в очередности декодирования. В соответствии со способами, описанными в этом раскрытии, порядок декодирования всех опорных изображений в полученном наборе опорных изображений является более ранним, чем порядок декодирования текущего изображения.

Процесс получения может включать в себя построение набора опорных изображений из множества поднаборов опорных изображений. Этот процесс может активизироваться один раз на одно изображение после декодирования заголовка слайса, но до декодирования любой единицы кодирования и до процесса декодирования для построения списка опорных изображений для слайса. Например, на основе полученных значений и сигнализированных синтаксических элементов видеодекодер 30 может определять значения POC для опорных изображений, которые принадлежат набору опорных изображений. На основе определенных значений POC видеодекодер 30 может построить поднабора опорных изображений, которые вместе образуют набор опорных изображений. Таким образом, путем построения поднаборов опорных изображений видеодекодер 30 может построить набор опорных изображений. Например, видеодекодер 30 может упорядочивать поднабора опорных изображений конкретным образом, чтобы получить набор опорных изображений. Упорядочение может относиться к способу,

которым видеодекодер 30 составляет список поднаборов опорных изображений, чтобы получить набор опорных изображений.

Как описано выше, для получения набора опорных изображений видеодекодер 30 может построить множество поднаборов опорных изображений. В некоторых примерах видеодекодер 30 может построить шесть поднаборов опорных изображений. Шесть поднаборов опорных изображений могут быть именованы: RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr и RefPicSetLtFoll. RefPicSetStCurr0 может именоваться RefPicSetStCurrBefore, и RefPicSetStCurr1 может именоваться RefPicSetStCurrAfter.

Следует понимать, что шесть поднаборов опорных изображений описываются с целью иллюстрации и не должны рассматриваться ограничительными. В качестве одного примера, видеодекодер 30 может строить меньшее количество поднаборов опорных изображений, чем шесть поднаборов опорных изображений, например, объединяя некоторые из поднаборов. Некоторые из этих примеров, где видеодекодер 30 строит менее шести поднаборов опорных изображений, описываются ниже. Однако, с целью иллюстрации, способы описываются с помощью примеров, где видеодекодер 30 строит шесть поднаборов опорных изображений.

Поднабора опорных изображений RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0 и RefPicSetStFoll1 могут идентифицировать краткосрочные опорные изображения. В некоторых примерах эти поднабора опорных изображений могут идентифицировать краткосрочные опорные изображения на основании того, являются ли краткосрочные опорные изображения более ранними в очередности отображения или более поздними в очередности отображения, чем текущее кодируемое изображение, а также могут ли краткосрочные опорные изображения потенциально использоваться для внешнего предсказания текущего изображения и изображения, следующего за текущим изображением в очередности декодирования, или могут ли потенциально использоваться для внешнего предсказания только изображений, следующих после текущего изображения в очередности декодирования.

Например, поднабор опорных изображений RefPicSetStCurr0 может включать в себя, и может включать единственно, идентификационную информацию, такую как значения POC, всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода или отображения, чем текущее изображение, и которые потенциально могут быть использованы для опорных (ссылки) в внешнем предсказании для текущего изображения, и потенциально могут быть использованы для опорных в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования. Поднабор опорных изображений RefPicSetStCurr1 может включать в себя, и может включать только, идентификационную информацию всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода или отображения, чем текущее изображение, и которые может потенциально использоваться для опорных в внешнем предсказании для текущего изображения, и потенциально могут быть использованы для опорных в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования.

Поднабор опорных изображений RefPicSetStFoll0 может включать в себя, и может включать только, идентификационную информацию всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода или отображения, чем текущее изображение, которые потенциально могут быть использованы для опорных в внешнем предсказании одного или более изображений, следующих после текущего

изображения в очередности декодирования, и которые не могут быть использованы для опорных в внешнем предсказании для текущего изображения. Поднабор опорных изображений RefPicSetStFoll1 может включать в себя, и может включать только, идентификационную информацию всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода или отображения, чем текущее изображение, которые потенциально могут быть использованы для опорных в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования, и которые не могут быть использованы для опорных в внешнем предсказании для текущего изображения.

Поднабора опорных изображений RefPicSetLtCurr и RefPicSetLtFoll могут идентифицировать долгосрочные опорные изображения. В некоторых примерах эти поднабора опорных изображений могут идентифицировать долгосрочные опорные изображения на основании того, являются ли долгосрочные опорные изображения более ранними в очередности отображения или более поздними в очередности отображения, чем текущее кодируемое изображение.

Например, поднабор опорных изображений RefPicSetLtCurr может включать в себя, и может включать только, идентификационную информацию всех долгосрочных опорных изображений, которые потенциально могут быть использованы для опорных в внешнем предсказании для текущего изображения, и которые потенциально могут быть использованы для опорных в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования. Поднабор опорных изображений RefPicSetLtFoll может включать в себя, и может включать только, идентификационную информацию всех долгосрочных опорных изображений, которые потенциально могут быть использованы для опорных в внешнем предсказании одного или более изображений, следующих после текущего изображения в очередности декодирования, и которые не могут быть использованы для опорных в внешнем предсказании для текущего изображения.

Если текущее изображение, подлежащее декодированию, является IDR изображением, видеодекодер 30 может установить поднабора опорных изображений RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr и RefPicSetLtFoll в «пустые». Это может быть, поскольку IDR изображение может не быть внешне предсказанным, и что изображение после IDR изображения в очередности декодирования не может использовать какое-либо изображение до IDR изображения в декодировании для опорного. Иначе (например, когда текущее изображение является не-IDR изображением), видеодекодер 30 может строить поднаборы краткосрочных опорных изображений и поднаборы долгосрочных опорных изображений путем осуществления следующего псевдокода.

Например, когда видеодекодер 30 декодирует экземпляр синтаксической структуры short_term_ref_pic_set() либо в заголовке слайса, либо путем обращения к указываемому набору параметров изображения, видеодекодер 30 может осуществлять следующий псевдокод, чтобы построить поднаборов опорных изображений RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0 и RefPicSetStFoll1.

```
cIdx=0
for(i=0, pocPred=PicOrderCnt(CurrPic); i<num_short_term_curr0; pocPred=RefPicSetStCurr0[i], i++, cIdx++)
RefPicSetStCurr0[i]=pocPred-short_term_ref_pic_id_delta_minus1[cIdx]-1
for(i=0, pocPred=PicOrderCnt(CurrPic); i<num_short_term_curr1; pocPred=RefPicSetStCurr1[i], i++, cIdx++)
```

```
RefPicSetStCurr1[i]=short_term_ref_pic_id_delta_minus1[cIdx]+1-pocPred
for(i=0, pocPred=PicOrderCnt(CurrPic); i<num_short_term_foll0; pocPred=RefPicSetStFoll
[i], i++, cIdx++)
RefPicSetStFoll0[i]=pocPred-short_term_ref_pic_id_delta_minus1[cIdx]-1
for(i=0, pocPred=PicOrderCnt(CurrPic); i<num_short_term_foll1; pocPred=RefPicSetStFoll
[i], i++, cIdx++)
RefPicSetStFoll1[i]=short_term_ref_pic_id_delta_minus1[cIdx]+1-pocPred
```

Если видеодекодер 30 определяет, что long_term_ref_pics_present_flag равен 0, видеодекодер 30 может установить RefPicSetLtCurr и RefPicSetLtFoll в «пустой», поскольку для этого случая не имеется долгосрочных опорных изображений. Иначе, если видеодекодер 30 декодирует экземпляр синтаксической структуры long_term_ref_pic_set() в заголовке слайса, видеодекодер 30 может осуществлять следующий псевдокод для построения поднаборов опорных изображений RefPicSetLtCurr и RefPicSetLtFoll.

```
cIdx=0
for(i=0; i<num_long_term_pps_curr; i++, cIdx++)
{
pIdx=long_term_ref_pic_idx_pps[i]
RefPicSetLtCurr[cIdx]=long_term_ref_pic_id_pps[pIdx]
}
for(i=0; i<num_long_term_add_curr; i++, cIdx++)
{
picIdDelta=long_term_ref_pic_id_delta_add[i]
RefPicSetLtCurr[cIdx]=PicOrderCnt(CurrPic)-picIdDelta
}
cIdx=0
for(i=0; i< num_long_term_pps_foll; i++, cIdx++)
{
pIdx=long_term_ref_pic_idx_pps[i+num_long_term_pps_curr]
RefPicSetLtFoll[cIdx]=long_term_ref_pic_id_pps[pIdx]
}
for( i=0; i<num_long_term_add_foll; i++, cIdx++)
{
picIdDelta=long_term_ref_pic_id_delta_add[i+num_long_term_add_curr]
RefPicSetLtFoll[cIdx]=PicOrderCnt(CurrPic)-picIdDelta
}
```

В соответствии со способами, описанными в этом раскрытии, опорное изображение с конкретным значением PicOrderCnt (значением POC) может именоваться включенным в набор опорных изображений для кодированного изображения, если опорное изображение включено в какой-либо из шести поднаборов в наборе опорных изображений для этого кодированного изображения. Опорное изображение с конкретным значением PicOrderCnt именуется включенным в конкретный поднабор набора опорных изображений, если конкретное значение PicOrderCnt (значение POC) равно одному из значений PicOrderCnt, включенных в этот поднабор.

После построения поднаборов опорных изображений видеодекодер 30 может получать набор опорных изображений. Например, видеодекодер 30 может упорядочивать поднабора опорных изображений для получения набора опорных изображений. В качестве одного примера, видеодекодер 30 может вносить в список

поднабор RefPicSetStCurr0 опорных изображений, за которым следует поднабор RefPicSetStCurr1 опорных изображений, за которым следует поднабор RefPicSetStFoll0 опорных изображений, за которым следует поднабор RefPicSetStFoll1 опорных изображений, за которым следует поднабор RefPicSetLtCurr опорных изображений, и затем поднабор RefPicSetLtFoll опорных изображений. В качестве другого примера, видеодекодер 30 может вносить в список поднабор RefPicSetStCurr0 опорных изображений, за которым следует поднабор RefPicSetStCurr1 опорных изображений, за которым следует поднабор RefPicSetLtCurr опорных изображений, за которым следует поднабор RefPicSetStFoll0 опорных изображений, за которым следует поднабор RefPicSetStFoll1, и затем поднабор RefPicSetLtFoll опорных изображений.

Другие изменения образа действий, которым видеодекодер 30 упорядочивает поднабора опорных изображений, могут быть возможными для получения набора опорных изображений. В некоторых примерах построение поднаборов опорных изображений и получение набора опорных изображений могут объединяться вместе. Например, построение поднаборов опорных изображений может автоматически иметь следствием получение видеодекодером 30 набора опорных изображений. Другими словами, видеодекодер 30 может не нуждаться в выполнении других этапов для построения поднаборов опорных изображений и получения набора опорных изображений, хотя может быть возможным, что видеодекодер 30 сначала строит поднабора опорных изображений и затем получает набор опорных изображений.

Кроме того, в соответствии со способами, описанными в этом раскрытии, построение набора опорных изображений описанным выше образом, может иметь следствием удовлетворение видеодекодером 30 следующим ограничениям. Например, конкретное опорное изображение с конкретным значением PicOrderCnt может не являться включенным в более, чем одно из поднаборов опорных изображений набора опорных изображений для текущего изображения. Другими словами, опорное изображение, идентифицированное в одном из поднаборов опорных изображений, не может быть идентифицировано ни в одном из других поднаборов опорных изображений. В качестве другого примера, в полученном наборе опорных изображений, не может иметься опорное изображение с temporal_id, более большим, чем для текущего изображение, которое включено в какое-либо из поднаборов опорных изображений, которые образуют набор опорных изображений.

Как описано выше, временное идентификационное значение (temporal_id) может быть иерархическим значением, которое указывает, какие изображения могут быть использованы для кодирования/предсказания текущего изображения. В общем, изображение с конкретным значением temporal_id возможно может являться опорным изображением для изображений с равными или большими значениями temporal_id, но не наоборот. Например, изображение со значением 1 для temporal_id возможно может являться опорным изображением для изображений со значениями temporal_id равными 1, 2, 3, …, но не для изображения со значением temporal_ID равным 0.

Нижнее значение temporal_id может также указывать нижнюю скорость отображения. Например, если видеодекодер 30 только декодировал изображения со значениями temporal_id, равными 0, частотой воспроизведения может быть 7,5 кадров в секунду. Если видеодекодер 30 только декодировал изображения со значениями temporal_id, равными 0 и 1, то частотой воспроизведения может быть 15 кадров в секунду, и т.д.

В некоторых примерах только изображения со значениями temporal_id, меньшими или равными temporal_id для текущего изображения, могут включаться в набор опорных изображений для текущего изображения. Как описано выше, только изображения со

значениями temporal_id, меньшими или равными temporal_id для текущего изображения, могут быть использованы в качестве опорных изображений. Таким образом, все опорные изображения с более низкими или равными значениями temporal_id могут быть использованы текущим изображением для внешнего предсказания и могут включаться в набор опорных изображений. Кроме того, некоторые опорные изображения, имеющие более высокие значения temporal_id, чем у текущего изображения, и которые подлежат использованию изображениями, следующими после текущего изображения в очередности декодирования, и имеющие более высокие значения temporal_id, чем у текущего изображения, исключаются из набора опорных изображений.

При наличии этих способов не требуется сигнализация temporal_id в дополнение к идентификации изображения для получения набора опорных изображений; следовательно, сигнализация набора опорных изображений становится более эффективной. Например, видеокодер 20 может не сигнализировать значения temporal_id для опорных изображений, которые принадлежат набору опорных изображений, и видеодекодер 30, может не принимать значения temporal_id для опорных изображений, которые принадлежат набору опорных изображений, с целью получения набора опорных изображений.

Кроме того, таким образом построенные поднаборов опорных изображений могут не идентифицировать опорные изображения значениями temporal_id, большими чем у текущего изображения. Например, видеодекодер 30 может быть способным строить поднабор опорных изображений, и обеспечивать, что нет опорного изображения, идентифицированного в каком-либо из поднаборов опорных изображений, имеющего значение temporal_id, большее чем у текущего изображения, поскольку согласование битового потока может потребовать, чтобы значения temporal_id не включались в битовый поток, сигнализированный видеокодером 20 и принимаемый видеодекодером 30. Таким образом видеодекодер 30 может получить набор опорных изображений без приема временных идентификационных значений для опорных изображений, которые принадлежат набору опорных изображений.

В вышеупомянутых примерах видеодекодер 30 может строить шесть поднаборов опорных изображений, четыре для краткосрочных опорных изображений (то есть RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, и RefPicSetStFoll1) и два для долгосрочных опорных изображений (то есть RefPicSetLtCurr и RefPicSetLtFoll). Однако, аспекты данного раскрытия не являются этим самым ограниченными. В других примерах два или большее число из этих поднаборов опорных изображений могут объединяться в один поднабор опорных изображений, имея следствием меньшее количество поднаборов опорных изображений, которое видеодекодер 30 строит. Последующее описывает некоторые неограничительные примеры, в которых видеодекодер 30 может строить меньше поднаборов опорных изображений. Могут иметься другие пути, которыми видеодекодер 30 может строить меньше поднаборов опорных изображений.

Например, в некоторых примерах, может не иметься разделения поднабора для текущего изображения и поднабора для последующих изображений в очередности декодирования. Таким образом, может быть два поднабора для краткосрочных опорных изображений, называемых RefPicSetSt0 и RefPicSetSt1, и может быть только один поднабор для долгосрочных опорных изображений, называемое RefPicSetLt. В этом примере поднабор RefPicSetSt0 опорных изображений может быть конкатенацией RefPicSetStCurr0 и RefPicSetStFoll0 с RefPicSetStCurr0 в начале результата конкатенации. В этом примере поднабор RefPicSetSt1 опорных изображений может быть конкатенацией RefPicSetStCurr1 и RefPicSetStFoll1, с RefPicSetStCurr1 в начале результата конкатенации.

Поднабор RefPicSetLt опорных изображений может быть конкатенацией RefPicSetLtCurr и RefPicSetLtFoll с RefPicSetLtCurr в начале результата конкатенации.

В качестве другого примера, может не иметься разделения поднабора с более ранней или более поздней очередностью вывода, чем у текущего изображения. Это может применяться только к краткосрочным опорным изображениям. Таким образом, могут иметься два поднабора для краткосрочных опорных изображений, называемые RefPicSetStCurr и RefPicSetStFoll. Поднабор RefPicSetStCurr опорных изображений может быть конкатенацией RefPicSetStCurr0 и RefPicSetStCurr1, с RefPicSetStCurr0 в начале результата конкатенации. Поднабор RefPicSetStFoll опорных изображений может быть конкатенацией RefPicSetStFoll0 и RefPicSetStFoll1, с RefPicSetStFoll0 в начале результата конкатенации.

В качестве другого примера, оба типа упомянутых выше разделений могут не применяться. Таким образом, может быть только один поднабор для краткосрочных опорных изображений, называемое RefPicSetSt, и только один поднабор для долгосрочных опорных изображений, называемое RefPicSetLt. Поднабор RefPicSetSt опорных изображений может быть конкатенацией RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0 и RefPicSetStFoll1 в приведенном порядке (или любом другом порядке), и RefPicSetLt может быть таким же, как выше.

Вышеупомянутые способы описывают примерный образ действий, которым видеодекодер 30 может получить набор опорных изображений. В течение процесса кодирования видеокодеру 20 может также требоваться декодировать кодированные изображения с целью кодирования последующих изображений, в том, что называется процессом восстановления. Соответственно, в некоторых примерах, видеокодер 20 также может быть сконфигурирован с возможностью получать набор опорных изображений. В некоторых примерах видеокодер 20 может осуществлять те же самые способы, которые видеодекодер 30 осуществлял для получения набора опорных изображений. Однако, получение набора опорных изображений видеокодером 20 может не требоваться в каждом примере, и видеодекодер 30 может быть только кодером, который получает набор опорных изображений.

Соответственно, в некоторых примерах, кодер видео (например, видеокодер 20 или видеодекодер 30) может кодировать (например, закодировать или декодировать, соответственно), информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. Например, видеокодер 20 может сигнализировать кодированный битовый поток, который включает в себя значения для определения, какие опорные изображения принадлежат набору опорных изображений. Подобным образом видеодекодер 30 может декодировать битовый поток для определения, какие опорные изображения принадлежат набору опорных изображений.

Кодер видео может строить множество поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений. Например, кодер видео может построить шесть поднаборов опорных изображений (то есть поднабора опорных изображений RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr и RefPicSetLtFoll), где каждый из поднаборов идентифицирует нуль или более опорных изображений. В некоторых примерах кодер видео может осуществлять кодирование текущего изображения на основании множества поднаборов опорных изображений.

Например, кодер видео может получать набор опорных изображений из построенной множества поднаборов опорных изображений. Например, кодер видео может

упорядочивать поднабора опорных изображений в любом порядке, чтобы получить набор опорных изображений, или может получить набор опорных изображений в виде части построения поднаборов опорных изображений. В некоторых примерах, на основе полученного набора опорных изображений кодер видео может кодировать текущее изображение. Поскольку набор опорных изображений получают из множества поднаборов опорных изображений, кодер видео можно рассматривать кодирующим текущее изображение на основании множества поднаборов опорных изображений.

В некоторых примерах, для упорядочивания поднаборов опорных изображений, кодер видео может строить поднабора опорных изображений в порядке, в котором поднабора опорных изображений должны входить в список в наборе опорных изображений. Например, кодер видео может сначала построить поднабор RefPicSetLtCurr опорных изображений, затем построить поднабор RefPicSetLtFoll опорных изображений, затем построить поднабор RefPicSetStCurr0 опорных изображений, затем построить поднабор RefPicSetStCurr1 опорных изображений, затем построить поднабор RefPicSetStFoll0 опорных изображений и затем построить поднабор RefPicSetStFoll1 опорных изображений. В этом иллюстративном примере порядком следования поднаборов опорных изображений набора опорных изображений может быть RefPicSetLtCurr, RefPicSetLtFoll, RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0 и RefPicSetStFoll1, в этом порядке, хотя другие порядки следования являются возможными.

В соответствии с примерными способами, описанными в этом раскрытии, после получения набора опорных изображений, видеодекодер 30 может декодировать слайсы в текущем изображения. Часть процесса декодирования касается построения одного или двух списков опорных изображений. Списком опорных изображений является список опорных изображений, который используется для предсказания слайса P или B. Для процесса декодирования P-слайса имеется один список опорных изображений (List 0). Для процесса декодирования B-слайса имеются два списка опорных изображений (List 0 и List 1). List 0, иногда называемый списком 0 опорных изображений или RefPicList0, является списком опорных изображений, используемым для внешнего предсказания P- или B- слайса. Все внешнее предсказания, используемые для P-слайсов, используют List 0. Список List 0 опорных изображений является одним из двух списков опорных изображений, используемых для двунаправленного предсказания для B-слайса, причем другим является списком 1 опорных изображений. List 1, иногда называемый списком 1 опорных изображений или списком RefPicList1 опорных изображений, используемым для предсказания B-слайса. Список 1 опорных изображений является одним из двух списков опорных изображений, используемых для предсказания для B-слайса, причем другим является список 0 опорных изображений. Некоторые блоки в B-слайсе могут быть предсказываемыми двунаправленно с использованием и List 0, и List1, и некоторые блоки в B-слайсе могут быть предсказываемыми однонаправленно с использованием либо List 0, либо List 1.

Для построения списков опорных изображений, видеодекодер 30 может осуществлять технику построения «по умолчанию» для построения начального List 0 и, для B-слайсов, начального List 1. Построение начального List 0 и начального List 1 может именоваться процессом инициализации. В некоторых примерах кодированный битовый поток может указывать, что видеодекодер 30 должен модифицировать начальный List 0 и/или начальный List 1, чтобы сформировать конечный List 0 и конечный List 1. Модификация начального List 0 и/или начального List 1 может именоваться процессом модификации. Процесс модификации может не требоваться в каждом примере, и форма, в которой видеодекодер 30 может осуществлять процесс модификации, описывается более

подробно ниже. В соответствии со способами, описанными в этом раскрытии, когда не требуется модификация начального List 0 или начального List 1, конечный List 0 или конечный List 1 (то есть список 0 или 1 опорных изображений, который используется для декодирования слайса текущего изображения), может соответствовать начальному List 0 или начальному List 1. Таким образом, переупорядочение списка опорных изображений может не требоваться.

В способах, описанных в этом раскрытии, видеодекодер 30 может строить начальный List 0 или начальный List 1 таким образом, что видеодекодеру 30, может не требоваться выполнять переупорядочение опорных изображений, подлежащих включению в начальный List 0 или начальный List 1, независимо от того, необходим ли процесс модификации, поскольку опорные изображения в каждом из поднаборов опорных изображений уже находятся в надлежащем порядке. Например, в некоторых других способах, независимо от того, необходим ли процесс модификации, требуется переупорядочение опорных изображений, подлежащих включению в начальный List 0 или начальный List 1 в соответствии с их значениями POC при добавлении или внесении опорных изображений в начальный List 0 или начальный List 1.

В процессе инициализации видеодекодер 30 может осуществлять способ построения «по умолчанию», чтобы построить начальный List 0 и начальный List 1. Способ построения «по умолчанию» может означать, что видеодекодер 30 строит начальные списки опорных изображений без приема синтаксических элементов от видеокодера 20, относящимся к способу, которым видеодекодер 30 должен строить начальные списки опорных изображений, или опорным изображениям, которые должны быть идентифицированы в начальных списках опорных изображений.

Видеодекодер 30 может активизировать процесс построения списка опорных изображений при декодировании заголовка P или B слайса. Например, при декодировании P слайса видеодекодер 30 может активизировать процесс для построения начального List 0, но может не активизировать процесс для построения начального List 1, поскольку блок в P-слайсе является только однонаправленно предсказанным относительно опорного изображения, идентифицированного в List 0. При декодировании B слайса, видеодекодер 30 может активизировать процесс для построения начального List 0 и построить начальный List 1, поскольку блок в B слайсе может быть двунаправленно предсказанными относительно опорных изображений, идентифицированных в каждом List 0 и List 1.

В соответствии с примерными способами, описанными в этом раскрытии, видеодекодер 30 может использовать поднабор опорных изображений для построения начального List 0 и начального List 1. Например, начальный List 0 и начальный List 1 могут иметь в списке нуль или более опорных изображений, идентифицированных в RefPicSetStCurr0, RefPicSetStCurr1 или RefPicSetLtCurr. В этом примере, когда процесс построения списка опорных изображений активизируется, может иметься, по меньшей мере, одно опорное изображение в RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr. Хотя начальный List 0 и начальный List 1 могут идентифицировать одно или более опорных изображений из одних и тех же поднаборов опорных изображений, порядок, в котором видеодекодер 30 добавляет опорные изображения в начальный List 0, может быть отличным от порядка, в котором видеодекодер 30 добавляет опорные изображения в начальный List 1.

В этом раскрытии, когда видеодекодер 30 добавляет (например, вносит) опорные изображения из одного или более поднаборов опорных изображений в начальный List 0 или начальный List 1, это раскрытие относится к видеодекодеру 30,

идентифицирующему опорные изображения в начальном List 0 или начальном List 1. Например, множество поднаборов опорных изображений может каждая(ое) идентифицировать нуль или более опорных изображений. Для построения начального List 0 и начального List 1 видеодекодер 30 может идентифицировать одно или более опорных изображений, которые идентифицированы в поднаборах опорных изображений в начальном List 0 или начальном List 1.

Во избежание путаницы и содействия ясности это раскрытие может относиться к видеодекодеру 30, вносящему или добавляющему нуль или более опорных изображений, идентифицированных в поднаборах опорных изображений, в начальный List 0 и начальный List 1, чтобы построить начальный List 0 и начальный List 1. Таким образом, добавление или внесение в список опорных изображений видеодекодером 30 означает, что видеодекодер 30 добавляет или вносит в список идентификатор опорного изображения, идентифицированного в поднаборе опорных изображений. Соответственно, результирующий начальный List 0 и начальный List 1 включают в себя множество идентификаторов для опорных изображений, которые потенциально могут быть использованы для кодирования блока или слайса текущего изображения. Эти опорные изображения сохраняются в соответственных буферах декодированных изображений в видеодекодере 30 и видеокодере 20.

Например, для построения начального List 0 видеодекодер 30 может сначала вносить (например, добавлять) опорные изображения, идентифицированные в RefPicSetStCurr0, в начальный List 0, за которыми следуют опорные изображения, идентифицированные в RefPicSetStCurr1, в начальный List 0, и затем опорные изображения, идентифицированные в RefPicSetLtCurr, в начальный List 0. Для построения начального List 1, видеодекодер 30 может сначала вносить (например, добавлять) опорные изображения, идентифицированные в RefPicSetStCurr1, в начальный List 1, за которыми следуют опорные изображения, идентифицированные в RefPicSetStCurr0, в начальный List 1, и затем опорные изображения, идентифицированные в RefPicSetLtCurr, в начальный List 1.

Кроме того, в дополнение к добавлению опорных изображений в поднаборах опорных изображений в различном порядке, видеодекодер 30 может использовать различное число опорных изображений из каждого из поднаборов опорных изображений при построении List 0 и List 1. Например, List 0 и List 1 не должны включать все из опорных изображений из RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr. Предпочтительнее число опорных изображений, которые вносятся в список из этих примерных поднаборов опорных изображений для построения начального List 0 и начального List 1, может основываться на синтаксических элементах, которые указывают максимальное число опорных изображений в рамках каждого из начального List 0 и начального List 1.

Например, для начального List 0 видеокодер 20 может сигнализировать синтаксический элемент num_ref_idx_l0_active_minus1 для P- и B-слайсов в заголовке слайса, и синтаксический элемент num_ref_idx_l1_active_minus1 для B-слайсов, которые являются двунаправленно предсказанными. Как описано выше, num_ref_idx_l0_active_minus1 может указывать максимальное число опорных изображений, которые могут находиться в начальном List 0, и num_ref_idx_l1_active_minus1 может указывать максимальное число опорных изображений, которые могут находиться в начальном List 1. В некоторых примерах может быть возможным, что значение num_ref_idx_l0_active_minus1 является отличным от значения num_ref_idx_l1_active_minus1, хотя это может быть не обязательным в

каждом примере. В некоторых примерах значение num_ref_idx_l0_active_minus1 может быть таким же, как значение num_ref_idx_l1_active_minus1.

Как описано выше, видеодекодер 30 может принимать в кодированном битовом потоке значения для num_short_term_curr0 и num_short_term_curr1. Видеодекодер 30 может задать переменную NumShortTermCurr0, равную num_short_term_curr0, и задать переменную NumShortTermCurr1, равную num_short_term_curr1. NumShortTermCurr0 может указывать количество опорных изображений в подмножестве RefPicSetStCurr0 опорных изображений, и NumShortTermCurr1 может указывать количество опорных изображений в подмножестве RefPicSetStCurr1 опорных изображений.

Видеодекодер 30 может также принимать в кодированном битовом потоке значения для num_long_term_pps_curr и num_long_term_add_curr. Видеодекодер 30 может задавать variableNumLongTermCurr в виде равной num_long_term_pps_curr плюс num_long_term_add_curr. NumLongTermCurr может указывать количество опорных изображений в RefPicSetLtCurr.

Для построения начального List 0 видеодекодер 30 может сначала добавлять опорные изображения (находящиеся) в RefPicSetStCurr0 в начальный List 0 до тех пор, пока видеодекодер 30 не добавит в начальный List 0 все опорные изображения из RefPicSetStCurr0, и пока число элементов в начальном List 0 (например, число опорных изображений, идентифицированных в List 0) меньше или равно num_ref_idx_l0_active_minus1. Например, NumShortTermCurr0 может указывать число опорных изображений в подмножестве RefPicSetStCurr0 опорных изображений. В этом примере видеодекодер 30 может вносить (например, добавлять) опорные изображения из поднабора опорных изображений RefPicSetStCurr0, пока число опорных изображений, внесенных из RefPicSetStCurr0, не будет равным NumShortTermCurr0. Однако, при внесении опорных изображений RefPicSetStCurr0 в начальный List 0, если общее количество элементов в начальном List 0 равняется num_ref_idx_l0_active_minus1, то видеодекодер 30 может остановить добавление опорных изображений из поднабора RefPicSetStCurr0 опорных изображений, даже если имеются дополнительные изображения в RefPicSetStCurr0, которые не были внесены в начальный List 0. В этом случае видеодекодер 30 возможно завершил построение начального List 0.

После того, как видеодекодер 30 внес все опорные изображения в подмножество RefPicSetStCurr0 опорных изображений, и если общее число элементов в начальном List 0 меньше num_ref_idx_l0_active_minus1, видеодекодер 30 может затем добавлять опорные изображения в RefPicSetStCurr1, пока видеодекодер 30 не идентифицирует все опорные изображения в RefPicSetStCurr1, и пока число элементов в начальном List 0 (например, число опорных изображений, идентифицированных в List 0) меньше или равно num_ref_idx_l0_active_minus1. Например, подобно вышеуказанному, NumShortTermCurr1 может указывать число опорных изображений в подмножестве RefPicSetStCurr1 опорных изображений. В этом примере видеодекодер 30 может вносить опорные изображения из поднабора RefPicSetStCurr1 опорных изображений, пока число опорных изображений, внесенных из RefPicSetStCurr1, не будет равно NumShortTermCurr1. Однако, при внесении опорных изображений из RefPicSetStCurr1, если общее число элементов в начальном List 0 равняется num_ref_idx_l0_active_minus1, то видеодекодер 30 может остановить добавление опорных изображений из поднабора RefPicSetStCurr1 опорных изображений, даже если имеются дополнительные изображения в RefPicSetStCurr1, которые не были внесены в начальный List 0. В этом случае видеодекодер 30 возможно завершил построение начального List 0.

После внесения в список видеодекодером 30 всех опорных изображений в поднаборе

опорных изображений RefPicSetStCurr1 и общем числе элементов в начальном List 0 меньшем num_ref_idx_l0_active_minus1, видеодекодер 30 может затем вносить опорные изображения из RefPicSetLtCurr, пока видеодекодер 30 не внесет в список все опорные изображения в RefPicSetLtCurr, и пока число элементов в начальном List 0 (например, число опорных изображений, идентифицированных в List 0) меньше или равно num_ref_idx_l0_active_minus1. Например, подобно вышеуказанному, NumLongTermCurr может указывать число опорных изображений в подмножестве RefPicSetLtCurr опорных изображений. В этом примере видеодекодер 30 может вносить опорные изображения из поднабора опорных изображений RefPicSetLtCurr, пока число опорных изображений, внесенных из RefPicSetLtCurr, не будет равно NumLongTermCurr. Однако, при внесении опорных изображений из RefPicSetLtCurr в начальный List 0, если общее число элементов в начальном List 0 равно num_ref_idx_l0_active_minus1, то видеодекодер 30 может остановить добавление опорных изображений из поднабора RefPicSetLtCurr опорных изображений, даже если имеются дополнительные изображения в RefPicSetLtCurr, которые не были внесены в начальный List 0. В этом случае, видеодекодер 30 возможно завершил построение начального List 0.

Последующий псевдокод иллюстрирует образ действий, которым видеодекодер 30 может построить начальный List 0.

```
cIdx=0
while(cIdx<=num_ref_idx_l0_active_minus1)
{
for(i=0; i< NumShortTermCurr0 && cIdx<= num_ref_idx_l0_active_minus1; cIdx++, i++)
RefPicList0[ cIdx ] = RefPicSetStCurr0[i]
for(i=0; i< NumShortTermCurr1 && cIdx <= num_ref_idx_l0_active_minus1; cIdx++, i++)
RefPicList0[ cIdx ]=RefPicSetStCurr1[i]
for( i=0; i< NumLongTermCurr && cIdx <= num_ref_idx_l0_active_minus1; cIdx++, i++ )
RefPicList0[ cIdx ] = RefPicSetLtCurr[i]
}
```

В вышеуказанном псевдокоде RefPicList0 может быть начальным List 0. В примерах, где модификация List 0 не требуется, конечный List 0 может соответствовать начальному List 0. Следовательно, в примерах, где модификация List 0 не требуется, RefPicList0, в вышеуказанном псевдокоде, может быть конечным List 0.

Видеодекодер 30 может подобным образом построить начальный List 1. Однако, для построения начального List 1 видеодекодер 30 может сначала добавлять опорные изображения из поднабора RefPicSetStCurr1 опорных изображений в начальный List 1, за которым следует поднабор опорных изображений RefPicSetStCurr0, в начальный List 1, и за которым следует поднабор опорных изображений RefPicSetLtCurr, в начальный List 1. Кроме того, подобно вышеуказанному, если при внесении опорных изображений из кого-либо из поднаборов опорных изображений RefPicSetStCurr1, RefPicSetStCurr0 и RefPicSetLtCurr, общее число элементов в начальном List 1 равняется num_ref_idx_l1_active_minus1, видеодекодер 30 может остановить добавление опорных изображений, даже если имеются дополнительные опорные изображения в этих поднаборах опорных изображений.

Например, для построения начального List 1 видеодекодер 30 может сначала вносить опорные изображения из RefPicSetStCurr1 до тех пор, пока видеодекодер 30 не идентифицирует все опорные изображения в RefPicSetStCurr1, и пока число элементов в начальном List 1 (например, число опорных изображений, идентифицированных в List 1) меньше или равно num_ref_idx_l1_active_minus1. Например, значение

NumShortTermCurr1 может указывать, когда видеодекодер 30 завершил внесение всех опорных изображений в поднаборе опорных изображений RefPicSetStCurr1. Однако, при внесении опорных изображений из RefPicSetStCurr1, если общее число элементов в начальном List 1 равняется num_ref_idx_l1_active_minus1, то видеодекодер 30 может остановить добавление опорных изображений в поднаборе опорных изображений RefPicSetStCurr1, даже если имеются дополнительные изображения в RefPicSetStCurr1, которые не были внесены в начальный List 1. В этом случае, видеодекодер 30 возможно завершил построение начального List 1.

После внесения видеодекодером 30 всех опорных изображений в подмножестве RefPicSetStCurr1 опорных изображений, и если общее число элементов в начальном List 1 меньше num_ref_idx_l1_active_minus1, видеодекодер 30 может затем вносить опорные изображения из RefPicSetStCurr0, пока видеодекодер 30 не внесет все опорные изображения в RefPicSetStCurr0, и пока число элементов в начальном List 1 (например, число опорных изображений, идентифицированных в List 1), меньше или равно num_ref_idx_l1_active_minus1. Например, подобно вышеуказанному, значение NumShortTermCurr0 может указывать, когда видеодекодер 30 завершил внесение всех опорных изображений в подмножестве RefPicSetStCurr0 опорных изображений. Однако, при внесении в список опорных изображений из RefPicSetStCurr0 в начальный List 1, если общее число элементов в начальном List 1 равняется num_ref_idx_l1_active_minus1, то видеодекодер 30 может остановить добавление опорных изображений в подмножестве RefPicSetStCurr0 опорных изображений, даже если имеются дополнительные изображения в RefPicSetStCurr0, которые не были внесены в начальный List 1. В этом случае, видеодекодер 30 возможно завершил построение начального List 1.

После внесения видеодекодером 30 в список всех опорных изображений в поднаборе опорных изображений RefPicSetStCurr0 и если общее число элементов в начальном List 1 меньше num_ref_idx_l1_active_minus1, видеодекодер 30 может затем вносить опорные изображения в RefPicSetLtCurr до тех пор, пока видеодекодер 30 не внесет в список все опорные изображения в RefPicSetLtCurr, и пока число элементов в начальном List 1 (например, число опорных изображений, идентифицированных в List 1) меньше или равно num_ref_idx_l1_active_minus1. Например, подобно вышеуказанному, значение NumLongTermCurr может указывать, когда видеодекодер 30 завершил внесение всех опорных изображений в подмножестве RefPicSetLtCurr опорных изображений. Однако, при внесении опорных изображений из RefPicSetLtCurr, если общее число элементов в начальном List 1 равняется num_ref_idx_l1_active_minus1, то видеодекодер 30 может остановить добавление опорных изображений, находящихся в подмножестве RefPicSetLtCurr опорных изображений, даже если имеются дополнительные изображения в RefPicSetLtCurr, которые не были внесены в начальный List 1. В этом случае видеодекодер 30 возможно завершил построение начального List 1.

Последующий псевдокод иллюстрирует образ действий, которым видеодекодер 30 может построить начальный List 1.

```
cIdx=0
while(cIdx<=num_ref_idx_l1_active_minus1)
{
for(i=0; i< NumShortTermCurr1 && cIdx <= num_ref_idx_l1_active_minus1; cIdx++, i++ )
RefPicList1[cIdx] = RefPicSetStCurr1[i]
for(i=0; i< NumShortTermCurr0 && cIdx <= num_ref_idx_l1_active_minus1; cIdx++, i++)
RefPicList1[cIdx]=RefPicSetStCurr0[i]
for(i=0; i< NumLongTermCurr && cIdx <= num_ref_idx_l1_active_minus1; cIdx++, i++)
```

RefPicList1[cIdx] = RefPicSetLtCurr[i]
}

В вышеуказанном псевдокоде RefPicList1 может быть начальным List 1. В примерах, где модификация List 1 не требуется, конечный List 1 может соответствовать начальному List 1. Следовательно, в примерах, где модификация List 1 не требуется, RefPicList1 в вышеуказанном псевдокоде может быть конечным List 1.

Предшествующее является одним примером образа действий, которым видеодекодер 30 может построить конечный список 0 и конечный список 1, когда модификация списка опорных изображений не требуется. В других примерах видеодекодер 30 может добавлять поднабор опорных изображений в другом порядке, чем описанные выше. В некоторых еще других примерах видеодекодер 30 может добавлять поднабора опорных изображений, отличные от описанных выше.

Хотя предшествующие примеры описали способы для построения списка опорных изображений, выполняемого видеодекодером 30, аспекты данного раскрытия не являются этим самым ограниченными, и видеокодер 20 может осуществлять подобные способы для построения списков опорных изображений. Однако, может не требоваться видеокодеру 20 строить списки опорных изображений таким же образом, каким видеодекодер 30 строит списки опорных изображений.

Соответственно, кодер видео (например, видеокодер 20 или видеодекодер 30) может быть сконфигурирован, чтобы кодировать (например, закодировать или декодировать) информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. Как описано выше, набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования.

Кодер видео также может быть сконфигурирован для построения множества поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений. Например, кодер видео может построить, по меньшей мере, поднаборы RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений. Кодер видео может строить дополнительные поднабора опорных изображений, такие как описанные выше.

Кодер видео может затем добавлять опорные изображения из первого поднабора опорных изображений, за которыми следуют опорные изображения из второго поднабора опорных изображений, и за которыми следуют опорные изображения из третьего поднабора опорных изображений в начальный список опорных изображений, пока количество элементов начального списка изображения не больше максимального числа допустимых элементов списка опорных изображений. Например, кодер видео может вносить опорные изображения из поднабора RefPicSetStCurr0 опорных изображений, за которыми следует поднабор RefPicSetStCurr1 опорных изображений, и за которыми следует поднабор RefPicSetLtCurr, в начальный List 0, пока число элементов в начальном List 0 не больше num_ref_idx_l0_active_minus1. Снова значение num_ref_idx_l0_active_minus1 может указывать максимальное число допустимых элементов списка опорных изображений для List 0.

В некоторых примерах кодер видео может добавлять опорные изображения из первого поднабора опорных изображений в начальный список опорных изображений, пока все опорные изображения в первом поднаборе опорных изображений не будут внесены в начальный список опорных изображений, или число элементов начального

списка изображения не будет равно максимальному числу допустимых элементов списка опорных изображений. Если число элементов начального списка изображения меньше максимального числа допустимых элементов списка опорных изображений, и после добавления опорных изображений из первого поднабора опорных изображений, кодер видео может добавлять опорные изображения из второго поднабора опорных изображений в начальный список опорных изображений, пока все опорные изображения во втором поднаборе опорных изображений не будут внесены в начальный список опорных изображений, или число элементов начального списка изображения не будет равно максимальному числу допустимых элементов списка опорных изображений. Если число элементов начального списка изображения меньше максимального числа допустимых элементов списка опорных изображений, и после добавления опорных изображений из второго поднабора опорных изображений, кодер видео может добавлять опорные изображения из третьего поднабора опорных изображений в начальный список опорных изображений, пока все опорные изображения в третьем поднаборе опорных изображений не будут внесены в начальный список опорных изображений, или число элементов начального списка изображения не будет равно максимальному числу допустимых элементов списка опорных изображений.

Кодер видео может подобным образом построить начальный List 1. Например, кодер видео может добавлять опорные изображения из второго поднабора опорных изображений, за которыми следуют опорные изображения из первого поднабора опорных изображений, и за которыми следуют опорные изображения из третьего поднабора опорных изображений, в начальный List 1, пока число элементов начального списка изображения в начальном List 1 не будет больше num_ref_idx_l1_active_minus1. Синтаксический элемент num_ref_idx_l1_active_minus1 может задавать максимальное число допустимых элементов в List 1.

В некоторых примерах, таких как где модификация не требуется, начальный List 0 и начальный List 1 могут соответствовать конечному List 0 и конечному List 1. Другими словами, кодер видео может построить конечный List 0 и конечный List 1 без модификации начального List 0 и начального List 1, когда модификация не требуется. В этих случаях после построения начального List 0 и начального List 1 кодеру видео может не требоваться выполнять дополнительные этапы, чтобы построить конечный List 0 и конечный List 1 (то есть списки опорных изображений, которые кодер видео использовал для кодирования блока текущего изображения).

Как указано в вышеуказанном псевдокоде, видеодекодер 30 может строить начальный List 0, тогда как cIdx меньше или равно num_ref_idx_l0_active_minus1, и может строить начальный List 1, тогда как cIdx меньше или равно num_ref_idx_l1_active_minus1. Это может иметь следствием построение видеодекодером 30 начального List 0 и начального List 1 без какого-либо незаполненного элемента в списках опорных изображений. Например, в некоторых других способах кодирования видео, видеодекодер для этих других способов кодирования видео построит начальный List 0 и List 1, используя способы, отличные от описанных в этом раскрытии. Для этих других способов кодирования видео, если число элементов в начальном List 0 и начальном List 1 было меньше максимального допустимого числа элементов, видеодекодер для этих других способов кодирования видео заполнит оставшиеся элементы в List 0 и List 1 значением “нет опорного изображения” для незаполненных элементов. Незаполненные элементы относятся к элементам в List 0 и List 1 после последнего элемента, который идентифицирует опорное изображение, и вплоть до последнего возможного элемента.

В качестве иллюстративного примера для помощи пониманию видеодекодер для

этих других способов кодирования видео может построить List 0 с пятью элементами, где максимальным числом допустимых элементов является десять элементов. В этом примере видеодекодер для этих других способов кодирования видео заполнит элементы от шестого до десятого как “нет опорного изображения”. В этом примере незаполненные элементы будут шестым элементом (например, элемент после последнего элемента, который идентифицирует опорное изображение) вплоть до десятого элемента (например, последний возможный элемент как задано максимальным числом допустимых элементов).

В соответствии со способами данного раскрытия, видеодекодер 30 может построить начальный List 0 и начальный List 1 так, что отсутствуют незаполненные элементы. Кроме того, в примерах, где не требуется модификация списка опорных изображений, конечный List 0 и конечный List 1 могут соответствовать начальному List 0 и начальному List 1. Следовательно, в примерах, где не требуется модификация списка опорных изображений, видеодекодер 30 может построить конечный List 0 и конечный List 1 так, что отсутствуют незаполненные элементы. Даже там, где требуется модификация списка опорных изображений, модификация может не приводить к каким-либо незаполненным элементам. Следовательно, даже в примерах, где модификация списка опорных изображений требуется, видеодекодер 30 может построить конечный List 0 и конечный List 1 так, что отсутствуют незаполненные элементы.

Например, List 0 и List 1 можно рассматривать списками с элементами, и каждый элемент может идентифицировать опорное изображение (например, посредством его значения POC). Другими словами, видеодекодер 30 может идентифицировать опорное изображение по его значению POC в каждом элементе List 0 и List 1. Число элементов в List 0 и List 1 может быть задано синтаксическими элементами num_ref_idx_l0_active_minus1 и num_ref_idx_l1_active_minus1, соответственно.

Чтобы обеспечить, что отсутствуют незаполненные элементы, видеодекодер 30 может многократно вносить (например, добавлять или идентифицировать) опорные изображения из поднаборов опорных изображений в начальный List 0 и начальный List 1, пока видеодекодер 30 не определит, какое опорное изображение должно быть идентифицировано в каждом возможном элементе начального List 0 и начального List 1. Например, как описано выше, для построения начального List 0, после добавления опорных изображений из поднаборов RefPicSetStCurr0 и RefPicSetStCurr1 опорных изображений в начальный List 0, видеодекодер 30 может добавлять опорные изображения из поднабора RefPicSetLtCurr опорных изображений в начальный List 0.

В некоторых примерах может быть возможным, что общее число элементов в начальном List 0 меньше максимального числа допустимых элементов в List 0 после добавления видеодекодером 30 опорных изображений из поднабора опорных изображений RefPicSetLtCurr в начальный List 0. Например, в псевдокоде для построения начального List 0, cIdx может указывать число элементов в List 0. В некоторых примерах, после идентификации видеодекодером 30 опорных изображений в RefPicSetLtCurr в начальном List 0, значение cIdx может быть меньше чем num_ref_idx_l0_active_minus1, где num_ref_idx_l0_active_minus1 указывает максимальное число допустимых элементов в List 0.

В соответствии со способами, описанными в этом раскрытии, после внесения опорных изображений из трех подмножеств в множества поднаборов опорных изображений, если число элементов в начальном List 0 меньше максимального числа допустимых элементов, видеодекодер 30 может многократно добавлять опорные изображения из трех поднаборов опорных изображений, пока не будут полными все элементы в List 0.

Например, после добавления видеодекодером 30 опорных изображений из множества RefPicSetLtCurr опорных изображений, и числе элементов в начальном List 0 менее максимального числа допустимых элементов, видеодекодер 30 может затем повторно составить список (например, повторно добавлять или повторно идентифицировать) опорных изображений из поднабора опорных изображений RefPicSetStCurr0.

В аспектах, описанных в этом раскрытии, когда видеодекодер 30 вносит опорные изображения из поднаборов RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений, видеодекодер 30 можно рассматривать добавляющим опорные изображения из этой множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений (например, List 0). Например, первое множество элементов может быть элементами в списке опорных изображений, в котором видеодекодер 30 идентифицировал опорные изображения из поднаборов RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений. Затем, если число элементов в списке опорных изображений меньше максимального допустимого числа элементов, видеодекодер 30 может повторно внести в список (например, повторно добавлять или повторно идентифицировать) одно или более опорных изображений из по меньшей мере одного из поднаборов RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов. Элементы, следующие после первого набора элементов, могут быть элементами, следующими после первого набора элементов, в которые видеодекодер 30 добавляет уже внесенные опорные изображения из одного или более подмножеств RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений, как описано ниже.

Если, при повторном добавлении опорных изображений из поднабора опорных изображений RefPicSetStCurr0, общее число элементов в начальном List 0 равняется num_ref_idx_l0_active_minus1, видеодекодер 30 может прекратить повторно добавлять опорные изображения в начальный List 0. В этом случае видеодекодер 30, возможно завершил построение начального List 0, и может не иметься незаполненных элементов. Иначе, видеодекодер 30 может повторно добавлять опорные изображения из поднабора RefPicSetStCurr0 опорных изображений, пока все опорные изображения из поднабора опорных изображений RefPicSetStCurr0 не будут переупорядочены.

Если после повторного добавления всех опорных изображений в подмножестве RefPicSetStCurr0 опорных изображений, число элементов в начальном List 0 меньше num_ref_idx_l0_active_minus1, видеодекодер 30 может затем повторно добавлять опорные изображения из поднабора RefPicSetStCurr1 опорных изображений образом, аналогичным тому, которым, в котором видеодекодер 30 переупорядочивает опорные изображения из поднабора опорных изображений RefPicSetStCurr0. Если после повторного добавления всех опорных изображений в поднаборе опорных изображений RefPicSetStCurr1, число элементов в начальном List 0 меньше num_ref_idx_l0_active_minus1, видеодекодер 30 может затем повторно добавлять опорные изображения из поднабора RefPicSetLtCurr опорных изображений образом, аналогичным тому, каким видеодекодер 30 переупорядочивает опорные изображения из поднаборов RefPicSetStCurr0 и RefPicSetStCurr1 опорных изображений. Видеодекодер 30 многократно добавляет опорные изображения из поднаборов опорных изображений, пока число элементов в начальном List 0 не будет равным максимальному числу допустимых элементов для List 0 (то есть равным num_ref_idx_l0_active_minus1).

Например, можно предположить, что имеется одно опорное изображение в RefPicSetStCurr0, одно опорное изображение в RefPicSetCurr1 и одно опорное

изображение в RefPicSetLtCurr. Кроме того, можно предположить, что num_ref_idx_l0_active_minus1 равно пяти. В этом примере видеодекодер 30 может идентифицировать опорное изображение в RefPicSetStCurr0 в двух элементах в начальном List 0. Например, видеодекодер 30 может идентифицировать опорное изображение в RefPicSetStCurr0 в первом элементе начального List 0, и повторно идентифицировать опорное изображение в RefPicSetStCurr0 в четвертом элементе начального List 0. В этом примере значением индекса для опорного изображения в RefPicSetStCurr0 может быть index[0] для первого элемента в начальном списке 0, и index[3] для четвертого элемента в начальном списке 0. Соответственно, в некоторых примерах, одно опорное изображение из одного из поднаборов опорных изображений может быть внесено (например, идентифицировано) более одного раза в начальные списки опорных изображений.

Видеодекодер 30 может аналогичным образом построить начальный List 1 так, что отсутствуют незаполненные элементы в начальном List 1. Например, видеодекодер 30 может многократно добавлять опорные изображения из поднаборов опорных изображений RefPicSetStCurr1, RefPicSetStCurr0 и RefPicSetLtCurr, в этом порядке, пока число элементов в начальном List 1 не будет равным максимальному числу допустимых элементов в List 1 (то есть равным num_ref_idx_l1_active_minus1).

Таким образом, поскольку циклы "for" являются вложенными внутри цикла "while", в вышеуказанном псевдокоде для построения начального List 0 и начального List 1, видеодекодер 30 может построить начальный List 0 и начальный List 1 так, что отсутствуют незаполненные элементы в начальном List 0 и начальном List 1 (то есть нет незаполненных элементов после процесса инициализации). В некоторых примерах каждый из элементов в начальном List 0 и начальном List 1 может идентифицировать опорное изображение из одного из поднаборов опорных изображений. В некоторых примерах может быть возможным, что одно или более опорных изображений из одного из поднаборов опорных изображений идентифицировано более одного раза в конечных списках опорных изображений, но в различных элементах с различными значениями индекса.

Хотя в предшествующих примерах описаны способы для построения списка опорных изображений без незаполненных элементов, как выполняемое видеодекодером 30, аспекты данного раскрытия не являются этим самым ограниченными, и видеокодер 20 может осуществлять подобные способы для построения списков опорных изображений без незаполненных элементов. Однако, может не требоваться видеокодеру 20 строить списки опорных изображений таким же образом, каким видеодекодер 30 строит список опорных изображений.

Соответственно, кодер видео (например, видеокодер 20 или видеодекодер 30) может быть сконфигурирован, чтобы кодировать (например, закодировать или декодировать) информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. Как описано выше, набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования.

Кодер видео может строить множество поднаборов опорных изображений, идентифицирующих каждое нуль или более опорных изображений (например, поднаборы RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений). Кодер видео может вносить (например, идентифицировать или добавлять) опорные изображения из

множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений. Кодер видео может определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений.

Если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, кодер видео может многократно повторно добавлять (например, повторно идентифицировать) одно или более опорных изображений из по меньшей мере одного из (поднаборов) опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений. Кодер видео может затем кодировать текущее изображение на основании построенного списка опорных изображений.

Как описано выше, в некоторых примерах, видеокодер 20 может сигнализировать синтаксические элементы, которые дают команду видеодекодеру 30 модифицировать начальный список или списки опорных изображений. Например, видеодекодер 30 может строить начальный список или списки опорных изображений описанным выше образом. Затем, в некоторых случаях, видеодекодер 30 может декодировать синтаксические элементы из кодированного битового потока, которые дают команду видеодекодеру 30 модифицировать начальный список или списки опорных изображений, чтобы построить конечный список или списки опорных изображений. В общем, в модификации, видеодекодер 30 может отобразить одно или несколько изображений, идентифицированных в одном или нескольких поднаборов опорных изображений из совокупности, в элемент в одном из списков опорных изображений после инициализации списка опорных изображений.

Например, после построения видеодекодером 30 начального списка или списков опорных изображений описанным выше образом видеодекодер 30 может модифицировать, по меньшей мере, один из элементов в одном из начальных списков опорных изображений способом, уведомляемым кодированным битовым потоком. Например, видеокодер 20 может указать в виде части синтаксических элементов модификации, какое изображение из одной из множества поднаборов опорных изображений должно идентифицироваться в элементе списка опорных изображений, даже если этот элемент списка опорных изображений уже может идентифицировать опорное изображение в качестве части процесса инициализации. В некоторых примерах способы модификации списка опорных изображений, описанные в этом раскрытии, могут позволять модификацию гибкую по существу. Например, видеодекодеру 30 может быть возможным идентифицировать в одном, или обоих из списков опорных изображений опорное изображение, которое не находится в начальных списках опорных изображений.

Как используется в этом раскрытии, фраза “модифицированный список опорных изображений”, относится к списку опорных изображений после модификации начального списка опорных изображений. Модифицированный список опорных изображений может быть конечным списком опорных изображений. Числом элементов в модифицированном списке опорных изображений является num_ref_idx_l0_active_minus1+1 для List 0 и num_ref_idx_l1_active_minus1+1 для List 1. Опорное изображение может появляться на нескольких индексах (например, элементе) в модифицированных опорных списках для List 0 и List 1.

Для модификации списка опорных изображений видеокодер 20 может

сигнализировать синтаксические элементы по Таблице 6.

Таблица 6
Синтаксис модификации списка опорных изображений

ref_pic_list_modification( ) {	Дескриптор
if(slice_type !=2) {	
ref_pic_list_modification_flag_l0	u(1)
if(ref_pic_list_modification_flag_l0)	
do {	
modification_of_ref_pic_ idc	ue(v)
if(modification_of_ref_pic_idc !=3)	
ref_pic_set_idx	
while(modification_of_ref_pic_idc !=3)	
}	
if(slice_type = =1) {	
ref_pic_list_modification_flag_l1	u(1)
if(ref_pic_list_modification_flag_l1)	
do {	
modification_of_ref_pic_ idc	ue(v)
if(modification_of_ref_pic_idc !=3)	
ref_pic_set_idx	
} while(modification_of_ref_pic_idc !=3)	
}	
}	

Синтаксические элементы modification_of_ref_pic_idc, short_term_ref_pic_set_idx и long_term_ref_pic_set_idx могут указывать переход от начальных списков опорных изображений к спискам опорных изображений, которые будут использоваться для декодирования часть.

Синтаксический элемент ref_pic_list_modification_flag_l0, равный 1, может указывать, что modification_of_ref_pic_idc присутствует для указания списка 0 опорных изображений. Флаг ref_pic_list_modification_flag_l0, равный 0, указывает, что этот синтаксический элемент не присутствует.

Когда ref_pic_list_modification_flag_l0 равен 1, число раз, которое modification_of_ref_pic_idc не равно 3 после ref_pic_list_modification_flag_l0, не может превышать num_ref_idx_l0_active_minus1 + 1.

Флаг ref_pic_list_modification_flag_l1, равный 1, может указывать, что синтаксис modification_of_ref_pic_idc присутствует для указания списка 1 опорных изображений. Флаг ref_pic_list_modification_flag_l1, равный 0, может указывать, что этот синтаксический элемент не присутствует.

Когда флаг ref_pic_list_modification_flag_l1 равен 1, число раз, которое modification_of_ref_pic_idc не равно 3 после ref_pic_list_modification_flag_l1, не может превышать num_ref_idx_l1_active_minus1 + 1.

Значения modification_of_ref_pic_idc вместе с short_term_ref_pic_set_idx или long_term_ref_pic_set_idx могут указывать, какие из опорных изображений отображаются повторно. Значения modification_of_ref_pic_idc указаны в Таблице 7. Значение первого modification_of_ref_pic_idc, которое следует непосредственно после ref_pic_list_modific ation_flag_l0 или ref_pic_list_modification_flag_l1, не может быть равным 3.

Значение ref_pic_set_idx указывает индекс к RefPicSetStCurr0, RefPicSetStCurr1 или RefPicSetLtCurr для опорного изображения, перемещаемого на текущий индекс в списке опорных изображений. Значение ref_pic_set_idx может находиться в диапазоне от 0 до max_num_ref_frames, включительно.

Таблица 7
Операции modification_of_ref_pic_idc для модификации списка опорных изображений

modification_of_ref_pic_idc	Указанная модификация
0	Для списка 0: ref_pic_set_idx присутствует и соответствует индексу к RefPicSetStCurr0; Для списка 1: ref_pic_set_idx присутствует и соответствует индексу к RefPicSetStCurr1
1	Для списка 0: ref_pic_set_idx присутствует и соответствует индексу к RefPicSetStCurr1; Для списка 1: ref_pic_set_idx присутствует и соответствует индексу к RefPicSetStCurr0
2	ref_pic_set_idx присутствует и соответствует индексу к RefPicSetLtCurr
3	ref_pic_idx не присутствует и завершается цикл для модификации начального списка опорных изображений

Для модификации списка опорных изображений, когда флаг ref_pic_list_modification_flag_l0 равен 1, видеодекодер 30 может модифицировать начальный список 0 опорных изображений (то есть начальный List 0), и когда ref_pic_list_modification_flag_l1 равен 1, видеодекодер 30 может модифицировать начальный список 1 опорных изображений (то есть начальный List 1). Чтобы помочь пониманию модификации списка опорных изображений, можно предположить, что переменная refIdxL0 является индексом в начальный List 0, и переменная refIdxL1 является индексом в начальный List 1. Другими словами, refIdxL0 может идентифицировать элемент начального List 0 (то есть индекс на начальный List 0 идентифицирует элемент начального List 0), и refIdxL1 может идентифицировать элемент начального List 1. Переменные refIdxL0 и refIdxL1 могут быть установлены равными 0 первоначально.

Видеодекодер 30 может обрабатывать синтаксические элементы для modification_of_ref_pic_idc в обработке появления синтаксических элементов в битовом потоке. Например, если видеокодер 20 сигнализирует, что требуется модификация списка опорных изображений для начального List 0, тогда видеодекодер 30 может обрабатывать очередность, в которой видеокодер 20 сигнализировал синтаксические элементы modification_of_ref_pic_idc для модификации начального List 0. Точно так же, если видеокодер 20 сигнализирует, что требуется модификация списка опорных изображений для начального List 1, тогда видеодекодер 30 может обрабатывать очередность, в которой видеокодер 20 сигнализирует синтаксические элементы modification_of_ref_pic_idc для модификации начального List 1.

Значением синтаксического элемента modification_of_ref_pic_idc может быть 0, 1, 2 или 3, как указано в Таблице 7. Если значением синтаксического элемента modification_of_ref_pic_idc является 3, то видеодекодер 30 может остановить модификацию начального списка опорных изображений. Иначе, видеодекодер 30 может продолжить модифицировать начальный список опорных изображений, пока значением синтаксического элемента modification_of_ref_pic_idc не будет 3. Например, видеокодер 20 может сигнализировать набор значений для синтаксического элемента modification_of_ref_pic_idc, и видеодекодер 30 может обрабатывать каждое из значений в порядке, в котором значения присутствуют в кодированном битовом потоке. Когда видеодекодер 30 обрабатывает значение синтаксического элемента modification_of_ref_pic_idc, равное 3, видеодекодер 30 может определить, что не требуется дальнейшая модификация.

Значение синтаксического элемента modification_of_ref_pic_idc, являющееся чем-то отличным от 3 (то есть 0, 1, или 2), может указывать, на основе какого поднабора опорных изображений видеодекодер 30 должен идентифицировать опорное изображение, которое подлежит внесению (например, добавлено в) в текущий элемент списка опорных изображений. Как описано выше, текущий элемент списка опорных изображений может быть идентифицирован значением refIdxLX, где LX представляет или List 0, или List 1.

Например, если видеодекодер 30 модифицирует начальный List 0, и modification_of_ref_pic_idc есть 0, то согласно Таблице 7, видеодекодер 30 может определить, какое опорное изображение из RefPicSetStCurr0 должно быть идентифицировано в текущем элементе списка опорных изображений на основании значения ref_pic_set_idx. Если видеодекодер 30 модифицирует начальный List 1, и modification_of_ref_pic_idc 0, то согласно Таблице 7 видеодекодер 30 может определить, какое опорное изображение из RefPicSetStCurr1 должно быть идентифицировано в текущем элементе списка опорных изображений, на основании значения ref_pic_set_idx. Например, переменная curRefPicSet может задавать, какой поднабор опорных изображений видеодекодер 30 должен использовать для модификации начального List 0 или начального List 1.

Например, если modification_of_ref_pic_idc равен 0, и видеодекодер 30 модифицирует начальный List 0, то curRefPicSet соответствует поднабору опорных изображений RefPicSetStCurr0. Если modification_of_ref_pic_idx есть 0, и видеодекодер 30 модифицирует начальный List 1, то curRefPicSet соответствует поднабору опорных изображений RefPicSetStCurr1.

Если видеодекодер 30 модифицирует начальный List 0, и значением modification_of_ref_pic_idc является 1, то согласно Таблице 7 видеодекодер 30 может определить, какое опорное изображение из RefPicSetStCurr1 должно быть идентифицировано в текущем элементе списка опорных изображений, на основании значения ref_pic_set_idx. Если видеодекодер 30 модифицирует начальный List 1, и значением modification_of_ref_pic_idc является 1, то согласно Таблице 7 видеодекодер 30 может определить, какое опорное изображение из RefPicSetStCurr0 должно быть идентифицировано в текущем элементе списка опорных изображений, на основании значения ref_pic_set_idx.

В этом случае, если modification_of_ref_pic_idc равно 1, и видеодекодер 30 модифицирует начальный List 0, то curRefPicSet соответствует поднабору опорных изображений RefPicSetStCurr1. Если индекс modification_of_ref_pic_idx равен 1, и видеодекодер 30 модифицирует начальный List 1, то curRefPicSet соответствует поднабору опорных изображений RefPicSetStCurr0.

Если видеодекодер 30 модифицирует начальный List 0 или начальный List 1, и modification_of_ref_pic_idc имеет значение 2, то согласно Таблице 7 видеодекодер 30 может определить, какое опорное изображение из RefPicSetLtCurr должно быть идентифицировано в текущем элементе списка опорных изображений, на основании значения ref_pic_set_idx. В этом примере, если modification_of_ref_pic_idc равно 2, и видеодекодер 30 модифицирует начальный List 0 или начальный List 1, то curRefPicSet соответствует поднабору опорных изображений RefPicSetLtCurr.

Как описано выше, синтаксический элемент ref_pic_set_idx может указывать индекс в одной из множества поднаборов опорных изображений. Другими словами, синтаксический элемент ref_pic_set_idx может указывать видеодекодеру 30 элемент из одного набора поднаборов опорных изображений. Видеодекодер 30 может определять опорное изображение, идентифицированное в элементе одной из множества поднаборов опорных изображений в качестве опорного изображения, которое подлежит идентификации в текущем индексе начального List 0 или начального List 1.

Переменная curRefPicPoc может соответствовать PicOrderCnt (curRefPicSet [ref_pic_set_idx]). Таким образом, значение curRefPicPoc может быть значением POC для опорного изображения, идентифицированного в элементе ref_pic_set_idx в curRefPicSet. Как описано выше, curRefPicSet может соответствовать RefPicSetStCurr0,

RefPicSetStCurr1 или RefPicSetLtCurr, на основании значения синтаксического элемента modification_of_ref_pic_idc и на основании того, модифицирует ли видеодекодер 30 начальный List 0 или начальный List 1.

Видеодекодер 30 может осуществлять следующий псевдокод для модификации списка опорных изображений. Например, в следующем псевдокоде видеодекодер 30 может идентифицировать изображение со значением POC, равным curRefPicPoc, в элементе начального списка опорных изображений. Переменная refIdxLX указывает позицию индекса для элемента в начальном списке опорных изображений. Например, когда видеодекодер 30 модифицирует начальный List 0, переменной refIdxLX может быть refIdxL0, и когда видеодекодер 30 модифицирует начальный List 1, переменной refIdxLX может быть refIdxL1.

После идентификации видеодекодером 30 опорного изображение со значением POC, равным curRefPicPOC, в начальном списке опорных изображений (то есть начальном List 0 или начальном List 1), видеодекодер 30 может сдвинуть позицию других оставшихся изображений на последующую в списке. Например, видеодекодер 30 может переместить опорные изображения, идентифицированные в начальном списке опорных изображений, в элементы, следующие после текущего элемента, на следующий элемент, чтобы построить модифицированный список опорных изображений. В качестве иллюстративного примера, можно предположить, что текущим элементом в начальном списке опорных изображений является третий элемент с индексом [2]. Видеодекодер 30 может переместить опорное изображение, в настоящий момент идентифицированное в третьем элементе с индексом [2], в следующий элемент (например, четвертый элемент с индексом [3]). Видеодекодер 30 может переместить опорное изображение, в настоящий момент идентифицированный в четвертом элементе с индексом [3], в пятый элемент с индексом [4]. В некоторых примерах видеодекодер 30 может начинать с последнего элемента в начальном списке опорных изображений, и переместить опорное изображение, идентифицированное в этом элементе, во временный новый элемент. Затем переместить опорное изображение, идентифицированное в предпоследнем элементе, в последний элемент, и т.д., пока видеодекодер 30 не дойдет до текущего элемента.

Видеодекодер 30 может затем увеличивать на приращение значение переменной refIdxLX. В этом псевдокоде длину RefPicListX (то есть RefPicList0 или RefPicList1) временно делают на один элемент больше длины, необходимой для конечного списка опорных изображений. Например, как описано выше, видеодекодер 30 может начинать с последнего элемента в начальном списке опорных изображений, переместить этот последний элемент во временный элемент, переместить предпоследний элемент в последний элемент, и т.д., чтобы модифицировать начальный список опорных изображений. После выполнения псевдокода видеодекодер 30 может оставить только элементы в индексе от 0 до num_ref_idx_lX_active_minus1, где значением num_ref_idx_lX_active_minus1 является num_ref_idx_l0_active_minus1 для List 0 и num_ref_idx_l1_active_minus1 для List 1.

```
for(cIdx = num_ref_idx_lX_active_minus1+1; cIdx>refIdxLX; cIdx− − )
RefPicListX[cIdx]=RefPicListX[cIdx−1]
RefPicListX[refIdxLX++]=reference picture with PicOrderCnt equal to currRefPicPoc
nIdx=refIdxLX
for(cIdx=refIdxLX; cIdx <= num_ref_idx_lX_active_minus1+1; cIdx++)
if(PicOrderCnt(RefPicListX[ cIdx]) !=currRefPicPoc)
RefPicListX[ nIdx++ ]=RefPicListX[ cIdx]
```

В вышеуказанном псевдокоде RefPicListX ссылается либо на RefPicList0 (то есть конечный List 0), либо на RefPicList1 (то есть конечный List 1) на основании того, модифицирует ли видеодекодер 30 начальный List 0 или начальный List 1. Переменная num_ref_idx_lX_active_minus1 ссылается на либо num_ref_idx_l0_active_minus1, либо ref_idx_l1_active_minus1 на основании того, модифицирует ли видеодекодер 30 начальный List 0 или начальный List 1.

Вышеуказанные способы описывают примерный образ действия, которым видеодекодер 30 может модифицировать начальный список опорных изображений. В течение процесса кодирования видеокодеру 20 может также потребоваться декодировать кодированное изображение с целями кодирования последующих изображений. Соответственно, в некоторых примерах, видеокодер 20 также может быть сконфигурирован для построения начальных списков опорных изображений и модифицирования начальных списков опорных изображений описанным выше образом. Однако, видеокодеру 20 может не требоваться модифицировать начальный список или списки опорных изображений в каждом примере. В некоторых примерах видеодекодер 30 может быть только кодером, который модифицирует начальное опорное изображение, используя способы, описанные выше.

Соответственно, в некоторых примерах, кодер видео (например, видеокодер 20 или видеодекодер 30) может построить начальный список опорных изображений (например, начальный List 0 или начальный List 1), используя способы, описанные выше. Кодер видео может определять, необходима ли модификация списка опорных изображений, на основании кодированных синтаксических элементов в кодированном битовом потоке. Если модификация списка опорных изображений является необходимой, кодер видео может модифицировать начальный список опорных изображений.

Например, если модификация списка опорных изображений является необходимой, кодер видео может идентифицировать опорное изображение в, по меньшей мере, одном из построенных поднаборов опорных изображений. Кодер видео может вносить (например, добавлять) идентифицированные поднабора опорных изображений в текущий элемент начального списка опорных изображений, чтобы построить модифицированный список опорных изображений. Кодер видео может затем кодировать (например, закодировать или декодировать) текущее изображение на основании модифицированного списка опорных изображений.

В качестве одного примера, кодер видео может строить поднаборы RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений. Чтобы идентифицировать опорное изображение в, по меньшей мере, одном из этих поднаборов опорных изображений, кодер видео может определить индекс в, по меньшей мере, одном из этих поднаборов опорных изображений. Кодер видео может затем определить опорное изображение, идентифицированное в элементе по меньшей мере одного из этих поднаборов опорных изображений на основании определенного индекса.

Например, кодер видео может кодировать (например, закодировать или декодировать) первый синтаксический элемент, такой как синтаксический элемент modification_of_ref_pic_idc, с помощью которого видеокодер идентифицирует одно из поднаборов опорных изображений (например, одно из поднаборов RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений). Кодер видео может осуществлять кодирование второго синтаксического элемента, такого как синтаксический элемент ref_pic_set_idx, который указывает индекс в идентифицированное поднабор опорных изображений (например, одного из поднаборов RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений).

В некоторых примерах кодер видео может быть дополнительно сконфигурирован для перемещения опорных изображений в начальном списке опорных изображений. Например, кодер видео может переместить опорные изображения, идентифицированные в начальном списке опорных изображений в элементах, следующих за текущим элементом, на следующий элемент в модифицированном списке опорных изображений.

В предшествующих примерах описан способ, которым видеокодер 20 и видеодекодер 30 может получать набор опорных изображений, а также примерные способы для построения списков опорных изображений, когда модификация не требуется, и когда модификация является необходимой. Однако, способы, описанные в этом раскрытии, не являются ограниченными этим. В некоторых примерах способы, описанные в этом раскрытии, могут быть направлены на управление буфером декодированных изображений (DPB). DPB может быть буфером, который сохраняет декодированные изображения.

Каждый из видеокодера 20 и видеодекодера 30 может включать в себя соответственные DPB. Например, в качестве части процесса кодирования, видеокодер 20 может декодировать текущее изображение, сохранять декодированное изображение в DPB видеокодера 20 и использовать декодированное изображение, сохраненное в DPB, для внешнего предсказания последующего изображения. Подобным образом, в качестве части процесса декодирования, видеодекодер 30 может декодировать текущее изображение и сохранять декодированное изображение в DPB видеодекодера 30. Видеодекодер 30 затем может использовать декодированное изображение для внешнего предсказания последующего изображения.

В некоторых примерах DPB либо для видеокодера 20, либо для видеодекодера 30 может сохранять декодированные изображения для переупорядочения вывода или задержки вывода. Например, видеодекодер 30 может определять, что декодированные изображения должны быть переупорядочены для вывода или что вывод декодированного изображения должен быть задержан. В этих примерах DPB в видеодекодере 30 может сохранять декодированные изображения для переупорядочения вывода или задержки вывода.

Способы управления DPB, описанные в этом раскрытии, могут быть направлены на способ действия, которым DPB выводит и удаляет декодированные изображения. Синтаксический элемент output_flag может воздействовать на процесс вывода и удаления декодированного изображения, и его определение может даваться в виде части семантики модуля уровня сетевой абстракции (NAL). Модуль NAL может быть определен в виде синтаксической структуры, которая включает в себя указание типа данных, которому следовать, и байтов, которые включают эти данные, в форме полезной нагрузки необработанной последовательности байтов (RBSP) с рассеянными по мере необходимости байтами предотвращения эмуляции. RBSP может быть синтаксической структурой, включающей в себя целое число байтов, которая инкапсулируется в модуле NAL. RBSP может быть либо пустой, либо имеющей форму строки битов данных, которая включает в себя синтаксические элементы, за которыми следует стоповый бит RBSP и за которыми следует нуль или более последующих битов, равных 0. Таблица 8 определяет синтаксис модуля NAL.

Таблица 8
Синтаксис модуля NAL

nal_unit(NumBytesInNALunit) {	Дескриптор
forbidden_zero_bit	f(1)
nal_ref_idc	u(2)
nal_unit_type	u(5)

NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
if(nal_unit_type==1||nal_unit_type==4||nal_unit_type==5){	
temporal_id	u(3)
output_flag	u(1)
reserved_one_4bits	u(4)
nalUnitHeaderBytes +=1	
}	
for(i=nalUnitHeaderBytes; i<NumBytesInNALunit; i++) {	
if(i+2<NumBytesInNALunit && next_bits(24)==0x000003){	
rbsp_byte[NumBytesInRBSP++]	b(8)
rbsp_byte[NumBytesInRBSP++]	b(8)
i+=2	
emulation_prevention_three_byte/*equal to 0x03 */	f(8)
} else	
rbsp_byte[NumBytesInRBSP++]	b(8)
}	
}	

В Таблице 8 output_flag может влиять на процесс вывода и удаления декодированного изображения, как описано более подробно ниже. Для любого изображения, если output_flag равен 1, изображение предназначено для вывода. Иначе изображение никогда не выводится. В способах, описанных в этом раскрытии, переменная OutputFlag соответствует синтаксическому элементу output_flag.

В некоторых примерах любой модуль NAL кодированного слайса для кодированного изображения текущего блока доступа может отличаться от какого-либо модуля NAL кодированного слайса для кодированного изображения предшествующего блока доступа одним или несколькими из следующего. Например, могут быть различными значения pic_parameter_set_id, могут быть различными значения nal_ref_idc, при одном из значений nal_ref_idc, являющимся равным 0. Значения pic_order_cnt_lsb могут быть различными. Значения IdrPicFlag могут быть различными. IdrPicFlag может быть равным 1 для обоих, и значения idr_pic_id могут быть различными.

В способах, описанных в этом раскрытии, блок (единица) доступа может быть определен в виде набора модулей NAL, которые являются последовательными в очередности декодирования и содержат одно кодированное изображение. В дополнение к кодированному изображению одно вспомогательное кодированное изображение, или другие модули NAL могут не содержать слайсы кодированного изображения. В некоторых примерах декодирование блока доступа может иметь результатом декодированное изображение. Кодированное изображение может быть кодированным представлением изображения, подлежащим использованию процессом декодирования.

Как указано в Таблице 4, синтаксис заголовка слайса может включать в себя синтаксический элемент pic_parameter_set_id, синтаксический элемент pic_order_cnt_lsb, синтаксический элемент IdrPicFlag и синтаксический элемент idr_pic_id. Как указано в Таблице 8, синтаксис модуля NAL может включать в себя синтаксический элемент nal_ref_idc.

С целью иллюстрации способы управления DPB описаны с точки зрения гипотетического опорного декодера (HRD). HRD может быть определен в виде гипотетической модели декодера, которая указывает ограничения на изменчивость удовлетворяющих техническим условиям потоков модулей NAL или удовлетворяющих техническим условиям потокам байтов, которые может выдавать процесс кодирования. Однако, в соответствии со способами, описанными в этом раскрытии, видеодекодер 30 может осуществлять способы управления DPB, и в некоторых примерах, является

возможным видеокодеру 20 также осуществлять способы управления DPB.

Модель HDR может давать определение буфера кодированного изображения (CPB), процесса мгновенного декодирования и буфера декодированных изображений (DPB). CPB может быть подобным CPB из моделей HDR, определенных в других предшествующих стандартах (то есть CPB может сохранять кодированные изображения). Способы, описанные в этом раскрытии, направлены на операции DPB, которые отличаются от операций в других стандартах. Снова следует понимать, что видеодекодер 30 и возможно видеокодер 20 могут осуществлять операции DPB, как описано ниже.

В общем, способы описанных в этом раскрытии, относятся к выводу и удалению декодированных изображений в DPB. Вывод декодированного изображения в этом контексте означает вывод декодированного изображения для отображения, сохранения или других целей. Однако, декодированное изображение, которое является выводимым, не должно обязательно удаляться из DPB. Например, видеодекодер 30 может не удалять декодированное изображение, которое является выводимым из DPB, поскольку видеодекодеру 30 может потребоваться использовать это декодированное изображение в качестве опорного изображения для внешнего предсказания последующего изображения. Удаление декодированного изображения в этом контексте означает удаление декодированного изображения из DPB.

Например, видеодекодер 30 может сохранять декодированные изображения в DPB видеодекодера 30 в порядке, в котором изображение декодируется. Однако порядок декодирования изображений может не являться одинаковым с очередностью вывода изображений. Например, могут быть изображения, следующие после текущего изображения в очередности декодирования, которые подлежат выводу ранее текущего изображения. Соответственно, в некоторых примерах, видеодекодер 30 может выполнять переупорядочение, согласно которому видеодекодер 30 переупорядочивает изображения в DPB, которые упорядочиваются в очередности декодирования в порядок вывода. Видеодекодер 30 может затем выводить декодированные изображения в их очередности вывода. Видеодекодер 30 может также удалять изображения из декодированного изображения, если изображение не является требуемым для вывода (то есть оно было выведено или не предназначается для вывода), и не является требуемым для внешнего предсказания (то есть не является необходимым для использования в качестве опорного изображения для внешнего предсказания).

В способах, описанных в этом раскрытии, видеодекодер 30 может удалять декодированное изображение из DPB, если декодированное изображение было выведено или не предназначается для вывода, и если декодированное изображение не идентифицировано в полученном наборе опорных изображений, каковое эквивалентно, что более не является необходимым для опорного в внешнем предсказании (то есть более не требуется для использования в качестве опорного изображения для внешнего предсказания). Снова, как описано выше, набор опорных изображений может идентифицировать опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения, и которые потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. В соответствии со способами, описанными в этом раскрытии, если декодированное изображение не идентифицировано в полученном наборе опорных изображений, то декодированное изображение может не требоваться в качестве опорного изображения для внешнего предсказания (например, декодирования) текущего изображения и одного или более изображений, следующих после текущего изображения в очередности

декодирования. Следовательно, такое декодированное изображение может быть удалено из DPB, если декодированное изображение не является необходимым для вывода, поскольку может не требоваться сохранять такое декодированное изображение в DPB при условии, что декодированное изображение не будет использоваться для внешнего предсказания.

Кроме того, в способах, описанных в этом раскрытии, видеодекодер 30 может удалять декодированное изображение до декодирования текущего изображения. Например, как описано выше, видеодекодер 30 может получить набор опорных изображений и построить список(ки) опорных изображений до декодирования текущего изображения. Поскольку видеодекодер 30 может получать набор опорных изображений до декодирования текущего изображения, видеодекодер 30 может быть сконфигурирован для определения, должно ли декодированное изображение, не являющееся необходимым для вывода, удаляться до декодирования текущего изображения. Например, после получения набора опорных изображений и до декодирования текущего изображения, видеодекодер 30 может определять, находится ли выводимое декодированное изображение или декодированное изображение, не предназначенное для вывода, в наборе опорных изображений. Затем, до декодирования текущего изображения, видеодекодер 30 может удалить декодированное изображение, не являющееся необходимым для вывода (то есть уже полученное или не предназначенное для вывода), если декодированное изображение не идентифицировано в наборе опорных изображений.

В некоторых примерах видеодекодер 30 может удалять декодированное изображение до декодирования текущего изображения. Однако, видеодекодер 30 может удалять декодированное изображение после построения списка(ов) опорных изображений. Например, видеодекодер 30 может получить набор опорных изображений и может построить списки опорных изображений на основании набора опорных изображений. Затем, до декодирования текущего изображения видеодекодер 30 может удалить декодированное изображение. В некоторых примерах видеодекодер 30 может также выводить декодированное изображение после построения списка(ов) опорных изображений.

Это раскрытие описывает способы удаления декодированных изображений в DPB, по меньшей мере, с двух точек зрения. С одной точки зрения видеодекодер 30 может удалять декодированные изображения на основании времени вывода, если изображения предназначены для вывода. С другой точки зрения видеодекодер 30 может удалять декодированные изображения на основании значений POC, если изображения предназначены для вывода. С любой точки зрения видеодекодер 30 может удалять декодированные изображения, которые не являются необходимыми для вывода (то есть уже выведены или не предназначены для вывода), когда декодированное изображение не находится в наборе опорных изображений, и до декодирования текущего изображения.

DPB может включать в себя несколько буферов, и каждый буфер может сохранять декодированное изображение, которое должно использоваться в качестве опорного изображения или сохраняться для будущего вывода. Первоначально, DPB пуст (то есть заполненность DPB обнулена). В описанных примерных способах удаление декодированных изображений из DPB может происходить до декодирования текущего изображения, но после синтаксического анализа видеодекодером 30 заголовка слайса для первого слайса текущего изображения.

Во втором аспекте следующие способы могут происходить мгновенно во время $t_r(n)$ в следующей последовательности. В этом примере $t_r(n)$ является временем удаления

CPB (то есть временем декодирования) блока доступа n, содержащего текущее изображение. Как описано в этом раскрытии, способы, происходящие мгновенно, могут означать, что в модели HDR полагается, что декодирование изображения является мгновенным (то есть неограниченно быстрым) с периодом времени для декодирования изображение, равным нулю.

Если текущим изображением является IDR изображение, и когда IDR изображение не является первым IDR изображением и величина значения pic_width_in_luma_samples или pic_height_in_luma_samples или max_dec_frame_buffering, полученная из активного набора параметров последовательности, отличается от значения pic_width_in_luma_samples или pic_height_in_luma_samples или max_dec_frame_buffering, полученного из набора параметров последовательности, которое был активным для предшествующего изображения, соответственно, видеодекодер 30 может сделать вывод, что синтаксический элемент no_output_of_prior_pics_flag равен 1 независимо от фактического значения no_output_of_prior_pics_flag. Если текущим изображеним является IDR изображение, и когда no_output_of_prior_pics_flag равен 1 или принимается, что будет равным 1, видеодекодер 30 может очистить все буферы DPB без вывода изображения в DPB и может установить в 0 заполненность DPB.

Как указано выше в Таблице 1, набор параметров последовательности может включать в себя синтаксические элементы pic_width_in_luma_samples и pic_height_in_luma_samples. Набор параметров последовательности может также включать в себя синтаксический элемент max_dec_frame_buffering. Как указано выше в Таблице 4, синтаксис заголовка слайса может включать в себя синтаксический элемент no_output_of_prior_pics_flag.

Когда текущее изображение не является IDR изображением, видеодекодер 30 может удалять все изображения (m) в DPB, для которых справедливы следующие условия. Первое условие может состоять в том, что изображение не включено в набор опорных изображений для текущего изображения. Второе условие может состоять в том, что OutputFlag изображения равен 0, или время его вывода DPB меньше или равно времени удаления CPB для текущего изображения. В этом примере временем удаления CPB является $t_r(n)$, представляющий момент времени, в который процесс удаления происходит (например, время до декодирования текущего изображения). Время вывода DPB для декодированного изображения m может быть задано переменной $t_{o,dpb}(m)$.

Следовательно, время вывода DPB, являющееся меньшим или равным времени удаления CPB, можно представить в виде $t_{o,dpb}(m) \leq t_r(n)$. Определение получения времени вывода DPB ($t_{o,dpb}$) дается более подробно ниже.

Таким образом видеодекодер 30 может удалять декодированные изображения из DPB до декодирования изображения на основании времени вывода декодированного изображения, и если декодированное изображение не идентифицировано в наборе опорных изображений. Когда видеодекодер 30 удаляет декодированное изображение из DPB, видеодекодер 30 может уменьшить заполненность DPB на единицу.

Последующее описывает способ, которым видеодекодер 30 может определять время, когда выводить декодированное изображение (например, время вывода DPB для декодированного изображения), и также описывает, когда видеодекодер 30 может сохранять декодированное изображение в DPB. Как описано выше, время вывода DPB для изображения может быть фактором определения, удаляется ли это изображение из DPB.

Когда видеодекодер 30 декодирует изображение, видеодекодер 30 сохраняет

изображение в DPB, и увеличивает заполненность DPB на единицу. Когда OutputFlag изображения равен 1, видеодекодер 30 может получить время вывода DPB для изображения на основании следующего уравнения.

$t_{o,dpb}(n)=t_r(n)+t_c*dpb\_output\_delay(n)$

В уравнении dpb_output_delay(n) может указываться в сообщении SEI хронирования изображения, связанном с блоком доступа, который включает в себя изображение. Сообщение SEI может быть определено в некоторых стандартах, таких как стандарт H.264/AVC.

Значение $t_{o,dpb}(n)$ может задавать, когда изображение подлежит выводу. Например, если OutputFlag равен 1 и $t_{o,dpb}(n)$ равен $t_r(n)$, видеодекодер 30 может выводить изображение. Иначе, если OutputFlag равен 0, видеодекодер 30 может не выводить изображение. В случаях, где OutputFlag равен 1 и $t_{o,dpb}(n)$ больше чем $t_r(n)$, видеодекодер 30 может выводить изображение в более позднее время (например, во время $t_{o,dpb}(n)$).

В некоторых примерах, когда видеодекодер 30 выводит изображение, видеодекодер 30 может обрезать изображение. Например, видеодекодер 30 может использовать прямоугольник обрезки, указанный в активном наборе параметров последовательности для изображения. Способы обрезки изображения обычно точно установлены и описаны в стандартах, таких как стандарт H.264/AVC.

В некоторых примерах видеодекодер 30 может определять разность между временем вывода DPB для изображения и временем вывода DPB для изображения, следующего после изображения в очередности вывода. Например, когда изображением (n) является изображение, которое видеодекодер 30 выводит, и не является последним изображением для битового потока, который выводится, видеодекодер 30 может определять значение $\Delta t_{o,dpb}(n)$, определенное как:

$\Delta t_{o,dpb}(n)=t_{o,dpb}(n_n)-t_{o,dpb}(n)$

В вышеуказанном уравнении n_n обозначает изображение, которое следует после изображения (n) в очередности вывода и имеет OutputFlag, равный 1. Кроме того, в вышеуказанном уравнении $\Delta t_{o,dpb}(n)$ представляет разность времени вывода DPB между изображением и последующим изображением в очередности вывода.

Во втором аспекте для удаления декодированных изображений, HDR может осуществлять способы мгновенно, когда блок доступа удаляется из CPB. Снова видеодекодер 30 может осуществлять удаление декодированных изображений из DPB, и видеодекодер 30 может не обязательно включать в себя CPB. В общем в этом раскрытии удаление декодированных изображений выполняется видеодекодером 30, и может также выполняться видеокодером 20. В этих примерах видеодекодер 30 и видеокодер 20 могут не требовать наличия CPB. Предпочтительнее CPB описывается в виде части модели HDR только с целью иллюстрации.

Как указано выше, во втором аспекте для удаления декодированных изображений видеодекодер 30 может удалять изображения из DPB до декодирования текущего изображения, но после синтаксического анализа заголовка слайса для первого слайса текущего изображения. Кроме того, подобно первому аспекту для удаления декодированных изображений, во втором аспекте видеодекодер 30 может выполнять функции, подобные описанным выше по отношению к первому аспекту, когда текущим изображением является IDR изображение.

Иначе, если текущее изображение не является IDR изображением, видеодекодер 30 может очистить без вывода буферы DPB, которые хранят изображение, которое

маркируется “не требуемым для вывода” и которые хранят изображения, не включенные в набор опорных изображений для текущего изображения. Видеодекодер 30 может также уменьшить заполненность DPB на число буферов, которые этот видеодекодер 30 очистил. Когда нет пустого буфера (то есть заполненность DPB соответствует размеру DBP), видеодекодер 30 может осуществлять процесс "выталкивания", описанный ниже. В некоторых примерах, когда пустого буфера нет, видеодекодер 30 может осуществлять процесс выталкивания модуль многократно, пока не будет пустого буфера, в котором видеодекодер 30 может сохранить текущее декодированное изображение.

Когда текущим изображением является IDR изображение, для которого no_output_of_prior_pics_flag не равен 1, и не полагается равным 1, видеодекодер 30 может выполнять следующее. Видеодекодер 30 может очистить без вывода буферы DPB, которые хранят изображение, которое маркируется “не требуемое для вывода” и которое не включено в набор опорных изображений для текущего изображения. Видеодекодер 30 может очистить все непустые буферы в DPB, многократно активизируя процесс "выталкивания", и может установить в 0 заполненность DPB.

Другими словами, когда текущим изображением является IDR изображение, видеодекодер 30 может осуществлять способы для очистки всех буферов в DPB. Когда текущим изображением является не IDR изображение, видеодекодер 30 может осуществлять способы для перемещения декодированных изображений в свободные буферы, чтобы сохранить текущее декодированное изображение.

Например, после того, как видеодекодер 30 декодирует текущее изображение, видеодекодер 30 может сохранить текущее изображение в DPB и увеличить заполненность DPB на единицу. В некоторых примерах, если OutputFlag текущего изображения равен 1, видеодекодер 30 может маркировать текущее изображение как “требуемое для вывода”. Иначе, если OutputFlag текущего изображения равен 0, видеодекодер 30 может маркировать текущее изображение как “не требуемое для вывода”.

Как описано выше, в некоторых примерах, видеодекодер 30 может осуществлять процесс выталкивания. В общем процесс выталкивания включает в себя осуществление вывода декодированного изображения. Например, видеодекодер 30 может осуществлять процесс выталкивания, когда текущим изображением является IDR изображение, и флаг no_output_of_prior_pics_flag не равен 1, и не полагается равным 1. Видеодекодер 30 может также осуществлять процесс выталкивания, если нет пустого буфера в DPB (то есть заполненность DPB равна размеру DPB), и пустой буфер требуется для сохранения декодированного (не IDR) изображения.

В общем, видеодекодер 30 может осуществлять следующие этапы для реализации процесса выталкивания. Видеодекодер 30 может сначала определить изображение, подлежащее выводу. Например, видеодекодер 30 может выбирать изображение, имеющее меньшее значение PicOrderCnt (POC) из всех изображений в DPB, которые маркируются “требуемое для вывода”. Видеодекодер 30 может обрезать выбранное изображение, используя прямоугольник обрезки, указанный в активном наборе параметров последовательности для изображения. Видеодекодер 30 может получить обрезанное изображение, и может маркировать изображение как “не требуемое для вывода”. Видеодекодер 30 может проверить буфер DPB, который сохранил обрезанное и полученное изображение. Если изображение не включено в набор опорных изображений, видеодекодер 30 может очистить этот буфер и может уменьшить на единицу заполненность DPB.

Хотя вышеуказанные способы для управления DPB описаны на основе контекста

видеодекодера 30, в некоторых примерах видеокодер 20 может осуществлять подобные способы. Однако видеокодер 20, реализующий подобные способы, не требуется в каждом примере. В некоторых примерах видеодекодер 30 может осуществлять эти способы, а видеокодер 20 может не осуществлять эти способы.

Таким образом кодер видео (например, видеокодер 20 или видеодекодер 30) может кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений. Снова, набор опорных изображений может идентифицировать опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования.

Кодер видео может получать набор опорных изображений любым образом, включая примерные способы, описанные выше. Кодер видео может определять, является ли декодированное изображение, сохраненное в буфере декодированных изображений, не требуемым для вывода и не идентифицированным в наборе опорных изображений. Когда декодированное изображение было выведено и не идентифицировано в наборе опорных изображений, кодер видео может удалять декодированное изображение из буфера декодированных изображений. После удаления декодированного изображения кодер видео может кодировать текущее изображение. Например, кодер видео может построить список(ки) опорных изображений, как описано выше, и кодировать текущее изображение на основании списка(ов) опорных изображений.

В предшествующих примерах описаны способы, которые видеокодер 20 и видеодекодер 30 могут использовать для получения набора опорных изображений, построения списков опорных изображений на основе списков опорных изображений, когда модификация не требуется и когда модификация необходима, а также способы для управления буфером декодированных изображений (DPB). Однако, аспекты данного раскрытия не являются этим самым ограниченными. В некоторых примерах способы, описанные в этом раскрытии, могут относиться к способу, которым видеокодер 20 сигнализирует, какие изображения находятся в наборе опорных изображений и являются долговременными опорными изображениями (или другими словами, какое изображение относится к набору долгосрочных опорных изображений), и способ, которым видеодекодер 30 определяет, какое изображение относится к набору долгосрочных опорных изображений.

Например, Таблица 2 включает в себя синтаксические элементы num_long_term_ref_pics_pps и long_term_ref_pic_id_pps[i], как являющиеся частью набора параметров изображения. Однако, аспекты данного раскрытия не являются ограниченными этим самым. В некоторых других примерах набор параметров последовательности (например, Таблица 1) может включать в себя синтаксические элементы num_long_term_ref_pics_pps и long_term_ref_pic_id_pps[i]. В примерах, где набор параметров последовательности включает в себя эти синтаксические элементы, данное раскрытие может ссылаться на синтаксические элементы как на num_long_term_ref_pics_sps и long_term_ref_pic_id_sps[i] во избежание путаницы. С целью иллюстрации способы описаны с помощью примеров, где набор параметров последовательности включает в себя эти синтаксические элементы.

Подобно определению num_long_term_ref_pics_pps, синтаксический элемент num_long_term_ref_pics_sps может указывать число долгосрочных опорных изображений-кандидатов, которые включены в набор параметров последовательности. Значение num_long_term_ref_pics_sps может находиться в диапазоне от 0 до 32, включительно.

Подобно определению синтаксического элемента long_term_ref_pic_id_pps[i], синтаксический элемент long_term_ref_pic_id_sps[i] может указывать идентификационную информацию i-го долгосрочного опорного изображения, включенную в набор параметров последовательности.

В некоторых примерах синтаксический элемент long_term_ref_pic_id_sps[i] может указывать долгосрочные опорные изображения-кандидаты, которые принадлежат набору опорных изображений для текущего изображения. Долгосрочные опорные изображения-кандидаты являются одним или несколькими долгосрочными опорными изображениями-кандидатами, которые могут быть долгосрочными опорными изображениями, которые видеодекодер 30 может использовать для внешнего предсказания текущего изображения или одного или более изображений, следующих после текущего изображения в очередности декодирования. Другими словами, долгосрочные опорные изображения-кандидаты могут указывать изображения, которые являются долгосрочными опорными изображениями и которые возможно могут быть использованы для внешнего предсказания текущее изображение и использоваться для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования. В некоторых примерах синтаксический элемент long_term_ref_pic_id_sps[i] может включать в себя значения POC для долгосрочных опорных изображений-кандидатов.

Однако не все долгосрочные опорные изображения-кандидаты обязательно используются для внешнего предсказания. Например, не все долгосрочные опорные изображения-кандидаты находятся в наборе опорных изображений для текущего изображения. Предпочтительнее нуль или более долгосрочных опорных изображений-кандидатов находятся в наборе опорных изображений.

В способах, описанных в этом раскрытии, видеокодер 20 может сигнализировать синтаксический элемент long_term_ref_pic_id в наборе параметров (например, синтаксический элемент long_term_ref_pic_id_sps в наборе параметров последовательности, или синтаксический элемент long_term_ref_pic_id_pps в наборе параметров изображения). Видеодекодер 30 может принимать синтаксический элемент long_term_ref_pic_id и идентифицировать долгосрочные опорные изображения-кандидаты. В соответствии со способами, описанными в этом раскрытии, видеодекодер 30 может дополнительно определять, какие из долгосрочных опорных изображений-кандидатов принадлежат набору опорных изображений. Например, видеодекодер 30 может быть сконфигурирован для выполнения этого определения на основании дополнительных синтаксических элементов, сигнализированных видеокодером 20 в кодированном битовом потоке.

Как указано в Таблице 4, видеокодер 20 может сигнализировать синтаксическую структуру long_term_ref_pic_set() в заголовке слайса текущего изображения. Таблица 5 описывает синтаксическую структуру long_term_ref_pic_set(). Например, синтаксическая структура long_term_ref_pic_set() может включать в себя синтаксические элементы num_long_term_pps_curr и num_long_term_pps_foll. Снова, следует отметить, что хотя синтаксические элементы num_long_term_pps_curr и num_long_term_pps_foll задаются в виде числа долгосрочных опорных изображений, включенных в набор параметров изображения, в примерах, где долгосрочные опорные изображения-кандидаты включены в набор параметров последовательности, эти синтаксические элементы могут указывать число долгосрочных опорных изображений-кандидатов, включенных в набор параметров последовательности. Например, во избежание путаницы синтаксический элемент num_long_term_pps_curr может именоваться как синтаксический элемент

num_long_term_sps_curr и синтаксический элемент num_long_term_pps_foll может именоваться как синтаксический элемент num_long_term_sps_curr.

Подобно синтаксическому элементу num_long_term_pps_curr синтаксический элемент num_long_term_sps_curr может указывать число всех долгосрочных опорных изображений, идентификационная информация которых включена в указываемое набор параметров последовательности в виде долгосрочных опорных изображений-кандидатов, и которые могут быть использованы для внешнего предсказания текущего изображения и одного или более изображений, следующих после текущего изображения в очередности декодирования. Подобно синтаксическому элементу num_long_term_pps_foll, синтаксический элемент num_long_term_sps_foll может задавать число всех долгосрочных опорных изображений, идентификационная информация которых включена в набор параметров последовательности, в качестве долгосрочных опорных изображений-кандидатов, которые не используются для внешнего предсказания текущего изображения, и которые могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования.

Кроме того, синтаксическая структура long_term_ref_pic_set(), сигнализированная в идентификационной информации опорного изображения, может включать в себя синтаксический элемент long_term_ref_pic_set_idx_pps[i]. Снова, в примерах, где долгосрочные опорные изображения-кандидаты сигнализируются в наборе параметров последовательности, синтаксический элемент long_term_ref_pic_set_idx_pps[i] можно рассматривать в качестве синтаксического элемента long_term_ref_pic_set_idx_sps[i]. Подобно синтаксическому элементу long_term_ref_pic_set_idx_pps[i], синтаксический элемент long_term_ref_pic_set_idx_sps[i] может задавать индекс в список идентификационной информации долгосрочного опорного изображения-кандидата, включенной в указываемое набор параметров последовательности, для i-го долгосрочного опорного изображения, наследуемого от набора параметров изображения для долгосрочного опорного изображения к набору опорных изображений для текущего изображения. Другими словами, синтаксический элемент long_term_ref_pic_set_idx_sps[i] может идентифицировать индекс в список долгосрочных опорных изображений-кандидатов в наборе параметров последовательности. На основе индекса видеодекодер 30 может идентифицировать долгосрочное опорное изображение в долгосрочных опорных изображениях-кандидатах, и может определять, что идентифицированное долгосрочное опорное изображение принадлежит набору опорных изображений для текущего изображения.

Например, видеодекодер 30 может осуществлять следующий псевдокод, подобный таковому в Таблице 5, для определения, какие из долгосрочных опорных изображений-кандидатов находятся в наборе опорных изображений для текущего изображения.

```
for(i=0; i<num_long_term_sps_curr+num_long_term_sps_foll; i++)
    long_term_ref_pic_set_idx_sps[i]
```

Таким образом видеодекодер 30 может декодировать синтаксические элементы, указывающие долгосрочные опорные изображения-кандидаты, которые идентифицированы в наборе параметров. Например, если набором параметров является набор параметров последовательности, видеодекодер 30 может декодировать синтаксические элементы long_term_ref_pic_id_sps[i], которые указывают долгосрочные опорные изображения-кандидаты, которые идентифицированы в наборе параметров последовательности. Если набором параметров является набор параметров изображения, видеодекодер 30 может декодировать синтаксические элементы

long_term_ref_pic_id_pps[i], которые указывают долгосрочные опорные изображения-кандидаты, которые идентифицированы в наборе параметров изображения.

Видеодекодер 30 может также декодировать синтаксические элементы, которые указывают, какие долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров, находятся в наборе опорных изображений для текущего изображения. Например, если набором параметров является набор параметров последовательности, видеодекодер 30 может декодировать синтаксические элементы num_long_term_sps_curr, num_long_term_sps_foll и long_term_ref_pic_set_idx_sps[i], и если набором параметров является набор параметров изображения, видеодекодер 30 может декодировать синтаксические элементы num_long_term_pps_curr, num_long_term_pps_foll и long_term_ref_pic_set_idx_pps[i]. В любом примере видеодекодер 30 может декодировать эти синтаксические элементы из заголовка слайса текущего изображения.

В соответствии со способами, описанными в этом раскрытии, синтаксические элементы long_term_ref_pic_id_sps[i] и long_term_ref_pic_id_pps[i] можно рассматривать в виде списка значений счетчика очередности изображения (POC) для долгосрочных опорных изображений-кандидатов, которые находятся в наборе опорных изображений, и можно кодировать (то есть закодировать или декодировать) в виде части набора параметров (например, набора параметров изображения и набора параметров последовательности). Синтаксический элемент long_term_ref_pic_set_idx_sps[i] или long_term_ref_pic_set_idx_pps[i] можно рассматривать обеспечивающим значение индекса в список значений POC для долгосрочных опорных изображений-кандидатов (например, индекс в long_term_ref_pic_id_sps[i] или long_term_ref_pic_id_pps[i]). В некоторых примерах синтаксический элемент long_term_ref_pic_set_idx_sps[i] или long_term_ref_pic_set_idx_pps [i] можно кодировать как часть заголовка слайса для текущего изображения.

Вышеуказанное описывает один путь, которым видеокодер 20 и видеодекодер 30 могут закодировать или декодировать, соответственно, синтаксические элементы для указания, какие изображения принадлежат набору долгосрочных опорных изображений для текущего изображения. На основе синтаксических элементов видеодекодер 30 и видеокодер 20 может определять, какие изображения принадлежат набору долгосрочных опорных изображений для текущего изображения. После того как видеодекодер 30 и видеокодер 20 определяет, какие изображения принадлежат набору долгосрочных опорных изображений, видеокодер 20 и видеодекодер 30 могут построить по меньшей мере один поднабор опорных изображений из множества поднаборов опорных изображений, и получить набор опорных изображений описанным выше образом. Например, на основании определения относительно того, какие изображения принадлежат набору долгосрочных опорных изображений, видеокодер 20 и видеодекодер 30 могут построить поднабор RefPicSetLtCurr опорных изображений, которое видеокодер 20 и видеодекодер 30 используют для получения набора опорных изображений.

В некоторых примерах могут иметься изображения, которые принадлежат набору долгосрочных опорных изображений, которые не включены в долгосрочные опорные изображения-кандидаты. Соответственно, могут быть дополнительные пути, которыми видеокодер 20 и видеодекодер 30 могут определять, какие изображения принадлежат набору долгосрочных опорных изображений для текущего изображения.

Например, как указано в Таблице 5, синтаксическую структуру long_term_ref_pic_set() заголовка слайса включает в себя синтаксический элемент long_term_ref_pic_id_delta_add [i]. Этот синтаксический элемент может указывать идентификацию долговременного опорного изображения, такую как значения POC, для i-го долгосрочного опорного

изображения, которое не наследуется от набора параметров опорного изображения, но включено в набор опорных изображений текущего изображения. Снова, в примерах, где долгосрочные опорные изображения-кандидаты идентифицированы в наборе параметров последовательности, синтаксический элемент long_term_ref_pic_id_delta_add [i] может указывать идентификацию долговременного опорного изображения i-го долгосрочного опорного изображения, которое не наследуется от набора параметров последовательности, но включено в набор опорных изображений текущего изображения.

Другими словами, синтаксический элемент long_term_ref_pic_id_pps[i] или long_term_ref_pic_id_sps[i] может идентифицировать долгосрочные опорные изображения-кандидаты, но может не обязательно идентифицировать все долгосрочные опорные изображения в наборе опорных изображений для текущего изображения. Например, могут иметься долгосрочные опорные изображения, которые подлежат использованию для внешнего предсказания текущего изображения и одного или более изображений, следующих после текущего изображения в очередности декодирования, которые не включаются в составление списка долгосрочных опорных изображений-кандидатов. Для таких долгосрочных опорных изображений видеокодер 20 и видеодекодер 30 может закодировать или декодировать, соответственно, идентификационную информацию, которая идентифицирует долгосрочные опорные изображения, которые принадлежат набору опорных изображений текущего изображения.

Например, как указано в Таблице 5, видеокодер 20 и видеодекодер 30 может закодировать или декодировать, соответственно, синтаксические элементы num_long_term_add_curr и num_long_term_add_foll. Синтаксический элемент num_long_term_add_curr может задавать число всех долгосрочных опорных изображений, идентификационная информация которых не включена в указываемое набор параметров изображения или набор параметров последовательности (как применимо), и которые могут быть использованы для внешнего предсказания текущего изображения и одного или более изображений, следующих после текущего изображения в очередности декодирования. Синтаксический элемент num_long_term_add_foll может задавать число всех долгосрочных опорных изображений, идентификационная информация которых не включена в указываемое набор параметров изображения или набор параметров последовательности (как применимо), которые могут не использоваться для внешнего предсказания текущего изображения, и которые могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования.

Видеодекодер 30 может осуществлять следующий псевдокод для определения, какие долгосрочные опорные изображения находятся в наборе опорных изображений. В этом примере долгосрочные опорные изображения могут не включаться в долгосрочные опорные изображения-кандидаты.

for (i=0; i<num_long_term_add_curr+num_long_term_add_foll; i++)

long_term_ref_pic_id_delta_add[i]

Как описано выше, способы, описанные в этом раскрытии, могут выполняться в соответствии со стандартом HEVC. Последующее является кратким описанием стандарта HEVC, для содействия пониманию. Кроме того, хотя способы описаны в контексте стандарта HEVC, способы могут быть расширены для других стандартов, включая частные стандарты.

JCT-VC работает над развитием стандарта HEVC. Работы по стандартизации HEVC основываются на развивающейся модели устройства кодирования видео, называемой Экспериментальной моделью HEVC (HM). HM предполагает несколько дополнительных

возможностей устройств кодирования видео относительно существующих устройств согласно, например, ITU-T H.264/AVC. Например, тогда как H.264 обеспечивает девять режимов кодирования с внутрикадровым предсказанием, HM может обеспечивать целых тридцать три режима кодирования с внутрикадровым предсказанием.

В общем, рабочая модель HM описывает, что видеокадр или изображение могут быть разделены на последовательность древовидных блоков или наибольших модулей кодирования (LCU), которые включают в себя выборки и яркости, и цветности. Древовидный блок имеет назначение подобное макроблоку по стандарту H.264. Слайс включает в себя ряд последовательных древовидных блоков в очередности декодирования. Видеокадр или изображение может сегментироваться на один или большее число слайсов. Каждый древовидный блок может быть расщеплен на блоки кодирования (CU) согласно дереву квадрантов. Например, древовидный блок, в качестве корневого узла дерева квадрантов, может быть расщеплен на четыре дочерних узла, и каждый дочерний узел может в свою очередь быть родительским узлом и расщепляться еще на четыре дочерних узла. Конечный, нерасщепленный дочерний узел, в качестве листового узел дерева квадрантов, содержит узел кодирования, то есть блок кодированного видео. Синтаксические данные, связанные с кодированным битовым потоком, могут задавать максимальное число раз, которое можно расщеплять древовидный блок, и могут также задавать минимальный размер узлов кодирования. Древовидные блоки могут именоваться блоками LCU в некоторых примерах.

CU включает в себя узел кодирования и блоки предсказания (PU), и блоки преобразования (TU), связанные с узлом кодирования. Размер CU соответствует размеру узла кодирования и должен быть квадратным по форме. Размер CU может иметь значения от 8x8 пикселов до размера древовидных блоков максимально в 64x64 пикселов или больше. Каждый CU может содержать один или большее число PU и один или большее число TU. Синтаксические данные, связанные с CU, могут описывать, например, разделение CU на один или большее число PU. Режимы разделения могут различаться между тем, является ли CU кодированным в режиме с пропуском или прямом режиме, кодированным в режиме внутрикадрового предсказания или кодированным в режиме внешнего предсказания. PU можно сегментировать, чтобы были неквадратными по форме. Синтаксические данные, связанные с CU, могут также описывать, например, сегментирование CU на один или большее число TU согласно дереву квадрантов. TU может быть квадратным или неквадратным по форме.

Стандарт HEVC позволяет преобразования в соответствии с блоками TU, которые могут отличаться для различных CU. Размер блоков TU обычно устанавливается на основании размера PU в пределах данного CU, заданного для сегментированного LCU, хотя, возможно, это не всегда имеет место. Блоки TU обычно имеют такой же или меньший размер, чем PU. В некоторых примерах остаточные выборки, соответствующие CU, можно подразделять на меньшие единицы, используя структуру дерева квадрантов, известную как "остаточное квадро дерево" (RQT). Листовые узлы RQT могут именоваться блоками преобразования (TU). Значения пикселов разности, связанные с блоками TU, могут быть преобразованы, чтобы выдавать коэффициенты преобразования, которые могут квантоваться.

В общем, PU включает в себя данные, относящиеся к процессу предсказания. Например, когда PU кодируется во внутрикадровом режиме, PU может включать в себя данные, описывающие режим внутрикадрового предсказания для PU. В качестве другого примера, когда PU кодируется во внешнем режиме, PU может включать в себя данные, задающие вектор движения для PU. Данные, задающие вектор движения для

PU, могут описывать, например, горизонтальный компонент вектора движения, вертикальный компонент вектора движения, разрешающую способность для вектора движения (например, точность в одну четвертую пиксела или точность в одну восьмую пиксела), опорное изображение, на которое вектор движения указывает, и/или список опорных изображений (например, List 0, List 1, или List C) для вектора движения.

В общем, TU используется для процессов преобразования и квантования. Данный CU, имеющий один или большее число PU, может также включать в себя один или несколько блоков преобразования (TU). Следуя предсказанию, видеокодер 20 может вычислять остаточные значения, соответствующие PU. Остаточные значения содержат значения разности пикселов, которые могут преобразовываться в коэффициенты преобразования, квантоваться, и сканироваться с использованием блоков TU, чтобы выдавать приведенные в последовательную форму коэффициенты преобразования для энтропийного кодирования. Это раскрытие обычно использует термин “видеоблок” для ссылки на узел кодирования CU. В некоторых конкретных случаях это раскрытие может также использовать термин “видеоблок” для ссылки на древовидный блок, то есть LCU, или CU, который включает в себя узел кодирования и блоки PU и блоки TU.

Видеопоследовательность обычно включает в себя последовательность видеокадров или изображений. Группа изображений (кадров) (GOP) обычно включает в себя последовательность из одного или более видеокадров. GOP может включать в себя синтаксические данные в заголовке GOP, заголовок одного или более изображений, или в другом месте, которая описывает ряд изображений, включенных в GOP. Каждый слайс изображения может включать в себя синтаксические данные слайса, которые описывают режим кодирования для соответственного слайса. Видеокодер 20 обычно оперирует видеоблоками в отдельных видео слайсов, чтобы кодировать видеоданные. Видеоблок может соответствовать узлу кодирования в пределах CU. Видеоблоки могут иметь фиксированные или переменные размеры и могут быть различными по размеру согласно предписанному стандарту кодирования.

В качестве примера, HM поддерживает предсказание в PU различных размеров. При условии, что размером конкретного CU является 2Nx2N, HM поддерживает внутрикадровое предсказание в PU размерами 2Nx2N или NxN, и внешнее предсказание в симметричных PU размерами 2Nx2N, 2NxN, Nx2N или NxN. HM также поддерживает асимметричное сегментирование для внешнего предсказания в PU размеров 2NxnU, 2NxnD, nLx2N и nRx2N. В асимметричном сегментировании одно направление CU не сегментируется, тогда как другое направление разделено на 25% и 75%. Порция CU, соответствующего 25%-ому разделу, обозначается “n”, за которым следует указание “Up” (вверх), “Down” (вниз), “Left” (влево) или “Right” (вправо). Таким образом, например, “2NxnU” относится к CU размером 2Nx2N, который разделен горизонтально с PU размером 2Nx0,5N в верхней части и PU размером 2Nx1,5N в нижней части.

В этом раскрытии “NxN” и “N на N” может использоваться взаимозаменяемо для обращения к размерностям в пикселах видеоблока в терминах размерностей по вертикали и горизонтали, например, 16x16 пикселов или 16 на 16 пикселов. В общем, блок 16x16 будет иметь 16 пикселов в вертикальном направлении (y=16) и 16 пикселов в горизонтальном направлении (x=16). Аналогично, блок NxN обычно имеет N пикселов в вертикальном направлении и N пикселов в горизонтальном направлении, где N представляет неотрицательное целочисленное значение. Пикселы в блоке могут располагаться в виде строк и столбцов. Кроме того, блоки не должны обязательно иметь такое же число пикселов в горизонтальном направлении, как и в вертикальном направлении. Например, блоки могут содержать NxM пикселов, где M не обязательно

равно N.

Следуя кодированию с внутрикадровым предсказанием или с внешнем предсказанием, использующим блоки PU в CU, видеокодер 20 может вычислять остаточные данные для блоков TU в CU. Блоки PU могут содержать пиксельные данные в пространственном домене (также называемом пиксельным доменом), и блоки TU могут содержать коэффициенты в домене преобразования, следуя применению преобразования, например, дискретного косинусного преобразования (DCT), целочисленного преобразования, вейвлет-преобразования или концептуально подобного преобразования к остаточным видеоданным. Остаточные данные могут соответствовать пиксельным разностям между пикселами незакодированного изображения и значениями предсказания, соответствующим блокам PU. Видеокодер 20 может формировать блоки TU, включающие в себя остаточные данные для CU, и затем преобразовать блоки TU, чтобы создать коэффициенты преобразования для CU.

Следуя всяким преобразованиям для выдачи коэффициентов преобразования, видеокодер 20 может выполнять квантование коэффициентов преобразования. Квантование обычно относится к процессу, в котором коэффициенты преобразования квантуются, чтобы возможно снизить объем данных, используемый для представления коэффициентов, обеспечивая дополнительную компрессию. Процесс квантования может снизить битовую глубину, связанную с некоторыми или всеми коэффициентами. Например, n-битовое значение может быть округлено в меньшую сторону к m-битовому значению в течение квантования, где n больше, чем m.

В некоторых примерах видеокодер 20 может использовать предопределенный порядок сканирования, чтобы сканировать квантованные коэффициенты преобразования для создания сериализированного вектора, который может быть энтропийно кодированным. В других примерах видеокодер 20 может выполнять адаптивное сканирование. После сканирования квантованных коэффициентов преобразования, чтобы формировать одномерный вектор, видеокодер 20 может энтропийно кодировать одномерный вектор, например, в соответствии с контекстнозависимым адаптивным кодированием с переменной длиной кодового слова (CAVLC), контекстнозависимым адаптивным двоичным арифметическим кодированием (CABAC), синтаксическим контекстнозависимым адаптивным двоичным арифметическим кодированием (SBAC), вероятностным с разбиением на интервалы энтропийным кодированием (PIPE) или другой методикой энтропийного кодирования. Видеокодер 20 может также энтропийно кодировать синтаксические элементы, связанные с кодированными видеоданными для использования видеодекодером 30 в декодировании видеоданных.

Для выполнения CABAC видеокодер 20 может назначать контекст в рамках контекстной модели символу, подлежащему передаче. Контекст может относиться, например, являются ли соседние значения символа ненулевыми или не являются. Чтобы выполнять CAVLC, видеокодер 20 может выбирать код переменной длины для символа, подлежащего передаче. Кодовые слова в VLC могут строиться так, что относительно более короткие коды соответствуют более вероятным символам, тогда как более длинные коды соответствуют менее вероятным символам. Таким образом, использование VLC может обеспечить экономию битов, например, используя кодовые слова равной длины для каждого символа, подлежащего передаче. Определение вероятности может быть основано на контексте, назначенном символу.

На Фиг. 2 показана концептуальная схема, иллюстрирующая примерную видео последовательность 33, которая включает в себя множество изображений, которые кодируются и передаются. В некоторых случаях, видео последовательность 33 может

именоваться группой изображений (GOP). Видеопоследовательность 33, как проиллюстрировано, включает в себя изображения 35A, 36A, 38A, 35B, 36B, 38B, и 35C, и конечное изображение 39 в очередности отображения. Изображение 34 является конечным изображением в очередности отображения для последовательности, имеющей место до последовательности 33. На Фиг. 2 в целом представлена иллюстративная структура предсказания для видео последовательности и предназначенная только для иллюстрации ссылок на изображения, используемые для предсказания видеоблоков других типов слайса или изображения (например, P- изображения или слайса, или B-изображения или слайса). Фактическая видео последовательность может содержать больше или меньше видеоизображений различных типов изображения и в различной очередности отображения. Видео последовательность 33 может включать в себя больше или меньше изображений, чем иллюстрируемые на Фиг. 2, и изображения, иллюстрируемые в видео последовательности 33, иллюстрируются с целью понимания и в качестве примеров.

Для блочного кодирования видео каждое из видеоизображений (видеокадров), включенных в последовательность 33, может быть разделено на видеоблоки, такие как блоки кодирования (CU) или блоки предсказания (PU). Например, каждый CU видео может включать в себя один или большее число PU. Видеоблоки в изображении с внутренним кодированием (I) предсказываются с использованием пространственного предсказания по отношению к соседним блокам в том же изображении. Видеоблоки в изображении с внешним кодированием (P или B) могут использовать пространственное предсказание по отношению к соседним блокам в том же изображении или временное предсказание по отношению к другим опорным изображениям.

Видеоблоки в B-изображении могут быть предсказаны с использованием двунаправленного предсказания для вычисления двух векторов движения на основе двух различных списков опорных изображений (например, списков 0 и 1 опорных изображений, именуемых List 0 и List 1). В некоторых случаях видеоблоки в B-изображении могут быть предсказаны с использованием однонаправленного предсказания на основе одного из двух различных списков опорных изображений (например, однонаправленных B-кодированных). Видеоблоки в P-изображении могут быть предсказаны с использованием однонаправленного предсказания для вычисления единственного вектора движения на основе единственного списка опорных изображений. В соответствии с появлением стандарта HEVC, видеоблоки могут кодироваться с использованием или однонаправленного предсказания для вычисления единственного вектора движения на основе одного из двух списков опорных изображений, или двунаправленного предсказания для вычисления двух векторов движения на основе двух списков опорных изображений. Два списка опорных изображений могут содержать прошлые опорные изображения или будущие опорные изображения, или и прошлые, и будущие опорные изображения в очередности отображения или вывода, и всегда прошлые опорные изображения в очередности декодирования, например.

В примере по Фиг. 2 конечное изображение 39 предназначено для кодирования во внутрикадровом режиме в виде I-изображения. В других примерах конечное изображение 39 может быть кодировано с помощью кодирования во внешнем режиме (например, как P-изображение) со ссылкой на конечное изображение 34 предшествующей последовательности, которым может быть I-изображение. Видеоизображения 35A-35C (все вместе “видеоизображение 35”) предназначены для кодирования как B-изображения с использованием двунаправленного предсказания со ссылкой на прошлое изображение и будущее изображение. В иллюстрируемом примере изображение 35A кодируется как

B-изображение со ссылкой на конечное изображение 34 и изображение 36A, как указано стрелками от изображений 34 и 36A к видеоизображению 35A. Изображения 35B и 35C аналогично кодируются.

Видеоизображения 36A-36B (собирательно "видеоизображение 36") могут предназначаться для кодирования в качестве изображений с использованием однонаправленного предсказания со ссылкой на прошлое изображение. В иллюстрируемом примере изображение 36A кодируется как P-изображение со ссылкой на конечное изображение 34, как указано стрелкой от изображения 34 к видеоизображению 36A. Изображение 36B кодируется аналогично.

Видеоизображения 38A-38B (собирательно "видеоизображение 38") могут предназначаться для кодирования, использующего двунаправленное предсказание со ссылкой на одно и то же прошлое изображение. В других примерах видеоизображения 38 могут кодироваться с использованием двунаправленного предсказания со ссылкой на по существу подобные прошлые изображения, включенные в списки опорных изображений. В иллюстрируемом примере изображение 38A кодируется с двумя ссылками на изображение 36A, как указано этими двумя стрелками от изображения 36A к видеоизображению 38A. Изображение 38B кодируется аналогично.

В соответствии со способами, описанными в этом раскрытии, видеокодер 20 может сигнализировать набор опорных изображений для каждого из изображений, находящихся в последовательности 33. Например, для изображения 35A это набор опорных изображений может идентифицировать все опорные изображения, которые могут быть использованы для внешнего предсказания изображения 35A, а также все опорные изображения, которые потенциально могут быть использованы для внешнего предсказания изображения, следующего после изображения 35A в очередности декодирования. Например, набор опорных изображений для изображения 35A может включать в себя значение POC для изображения 34 и изображения 36A, а также значения POC для дополнительных опорных изображений, таких как те, которые потенциально могут быть использованы для внешнего предсказания изображения, следующего после изображения 35A в очередности декодирования. Изображения после изображения 35A могут быть такими изображениями, которые следуют за изображением 35A в очередности декодирования, и которые находятся внутри видеопоследовательности 33, в этом примере.

Видеодекодер 30 может затем получить набор опорных изображений для изображения 35A описанным выше образом. Например, видеодекодер 30 может определять значения POC для опорных изображений, которые принадлежат набору опорных изображений, как описано выше. Видеодекодер 30 может далее построить, по меньшей мере, четыре или, по меньшей мере, пять поднаборов опорных изображений, и в некоторых примерах, до шести поднаборов опорных изображений, описанных выше. Видеодекодер 30 может расположить шесть поднаборов опорных изображений в конкретном порядке для получения набора опорных изображений для изображения 35A.

Видеодекодер 30 может далее построить начальные списки опорных изображений описанным выше образом, причем переупорядочение изображений, подлежащих включению в начальные списки опорных изображений, не требуется. Когда модификация списка опорных изображений отключена, видеодекодер 30 может установить конечные списки опорных изображений соответственными начальным спискам опорных изображений. Кроме того, видеодекодер 30 может построить списки опорных изображений таким образом, что отсутствуют незаполненные элементы в списках опорных изображений. Например, видеодекодер 30 может многократно вносить опорные

изображения из поднаборов опорных изображений до тех пор, пока число элементов в списках опорных изображений не будет равным максимальному числу допустимых элементов для списка опорных изображений. В некоторых примерах видеодекодер 30 может модифицировать начальные списки опорных изображений описанным выше образом (например, на основании опорных изображений в, по меньшей мере, одном из построенных поднаборов опорных изображений). Кроме того, в некоторых примерах, видеодекодер 30 может удалять декодированные изображения из DPB в видеодекодере 30, используя примерные способы, описанные в этом раскрытии, такие как удаление декодированных изображений, которые не идентифицированы в наборе опорных изображений для текущего изображения, подлежащего декодированию, и которые не требуются для вывода. Кроме того, в некоторых примерах видеодекодер 30 может определять, какие долгосрочные опорные изображения находятся в наборе опорных изображений, описанном выше образом, причем идентификационная информация списка долгосрочных опорных изображений-кандидатов может быть включена в набор параметров.

На Фиг. 3 показана блок-схема, иллюстрирующая примерный видеокодер 20, который может осуществлять способы, описанные в этом раскрытии. Видеокодер 20 может выполнять внутрикадровое и внешнее кодирование видеоблоков в рамках видео слайсов. Внутрикадровое кодирование основывается на пространственном предсказании для уменьшения или удаления пространственной избыточности в видео внутри данного видеокадра или изображения. Внешнее кодирование основывается на временном предсказании для уменьшения или удаления временной избыточности в видео в рамках смежных кадров или изображений в видео последовательности. Внутрикадровый режим (I-режим) может относиться к любому из нескольких режимов пространственного сжатия. Внешнее режимы, такие как однонаправленное предсказание (P-режим) или двунаправленное предсказание (B-режим), могут относиться к любому из нескольких режимов временного сжатия.

В примере по Фиг. 3 видеокодер 20 включает в себя блок 35 разделения, модуль 41 предсказания, буфер 64 декодированных изображений (DPB), сумматор 50, модуль 52 преобразования, блок 54 квантования и блок 56 энтропийного кодирования. Модуль 41 предсказания включает в себя блок 42 оценки движения, блок 44 компенсации движения и модуль 46 внешнего предсказания. Для восстановления видеоблока видеокодер 20 также включает в себя модуль 58 обратного квантования, модуль 60 обратного преобразования, и сумматор 62. Деблокирующий фильтр (не показан на Фиг. 3) также может включаться в состав, чтобы отфильтровывать границы блоков для удаления артефактов блочности из восстановленного видео. Если требуется, деблокирующий фильтр обычно отфильтровывает вывод сумматора 62. Дополнительные контурные фильтры (в контуре обработки или постобработки) могут также использоваться в дополнение к деблокирующему фильтру.

Как показано на Фиг. 3, видеокодер 20 принимает видеоданные, и блок 35 разделения разделяет данные на видеоблоки. Это разделение может также включать в себя разделение на слайсы, мозаичные фрагменты (tile), или другие более крупные блоки, а также разделение на видеоблоки, например, согласно структуре дерева квадрантов блоков LCU и CU. Видеокодер 20 в целом иллюстрирует компоненты, которые кодируют видеоблоки в рамках слайса видео, подлежащего кодированию. Слайс можно делить на множество видеоблоков (и возможно на наборы видеоблоков, называемых мозаичными фрагментами (tile)). Модуль 41 предсказания может выбирать один из ряда возможных режимов кодирования, таких как один из ряда режимов

внутрикадрового кодирования или одного из ряда режимов кодирования, для текущего видеоблока на основании результатов ошибки (например, скорости кодирования и степени искажения). Модуль 41 предсказания может предоставлять результирующий внутрикадрово- или внешне кодированный блок на сумматор 50 для формирования остаточных данных блока, и на сумматор 62, для восстановления кодированного блока для использования в качестве опорного изображения.

Модуль 46 внутрикадрового предсказания в рамках модуля 41 предсказания может выполнять кодирование с внутрикадровым с предсказанием для текущего видеоблока относительно одного или более соседних блоков в том же изображении или слайсе, что и текущий блок, который подлежит кодированию, чтобы обеспечить пространственное сжатие. Блок 42 оценки движения и блок 44 компенсации движения в рамках модуля 41 предсказания выполняют кодирование с внешнем предсказанием текущего видеоблока относительно одного или более предсказанных блоков в одном или нескольких опорных изображениях, чтобы обеспечивать временное сжатие.

Блок 42 оценки движения может быть сконфигурирован для определения режима внешнего предсказания для видео слайса согласно заранее заданной конфигурации для видео последовательности. Заранее заданная конфигурация может обозначать видео слайсы в последовательности в виде P-слайсов или B-слайсов. Блок 42 оценки движения и блок 44 компенсации движения могут быть тесно интегрированными, но иллюстрируются отдельно с концептуальными целями. Оценка движения, выполняемая блоком 42 оценки движения, является процессом формирования векторов движения, которые оценивают движение для видеоблоков. Вектор движения, например, может указывать смещение PU для видеоблока внутри текущего видеоизображения относительно предсказанного блока внутри опорного изображения.

Предсказанным блок является блок, который считается близко соответствующим PU в видеоблоке, который подлежит кодированию, в терминах разности пикселов, которая может быть определяться суммой абсолютной разности (SAD), сумма квадрата (ов) разности (SSD), или другой метрикой разности. В некоторых примерах видеокодер 20 может вычислять значения для целочисленных суб-пиксельных позиций для опорных изображений, сохраненных в буфере 64 декодированных изображений. Например, видеокодер 20 может интерполировать значения позиций в одну четвертую пиксела, позиций в одну восьмую пиксела, или других дробных позиций пикселов для опорного изображения. Следовательно, блок 42 оценки движения может выполнять поиск движения относительно позиций полных пикселов и позиций дробных пикселов и выводить вектор движения с точностью дробного пиксела.

Блок 42 оценки движения вычисляет вектор движения для PU в видеоблоке в внешне-кодированном слайсе, сравнением позиции PU с позицией предсказанного блока для опорного изображения. Опорное изображение может выбираться из первого списка опорных изображений (List 0) или второго списка опорных изображений (List 1), каждый из которых идентифицирует одно или более опорных изображений, сохраненных в буфере 64 декодированных изображений. Блок 42 оценки движения посылает вычисленный вектор движения на блок 56 энтропийного кодирования и блок 44 компенсации движения.

Компенсация движения, выполняемая блоком 44 компенсации движения, может предусматривать извлечение или формирование предсказанного блока на основании вектора движения, определенного согласно оценке движения, возможно выполняя интерполяции с суб-пиксельной точностью. По приему вектора движения для PU в текущем видеоблоке блок 44 компенсации движения может определять местоположение

предсказанного блока, на который указывает вектор движения в одном из списков опорных изображений. Видеокодер 20 формирует остаточный видеоблок вычитанием пиксельных значений предсказанного блока из пиксельных значений текущего кодируемого видеоблока, формирующим значения пиксельной разности. Значения разности пикселов образуют разностные данные для блока, и могут включать в себя разностные составляющие и яркости, и цветности. Сумматор 50 представляет компонент или компоненты, которые выполняют эту операцию вычитания. Блок 44 компенсации движения может также формировать синтаксические элементы, связанные с видеоблоками и слайсом видео, для использования видеодекодером 30 в декодировании видеоблоков для слайса видео.

Модуль 46 внутрикадрового предсказания может во внутрикадровом режиме предсказывать текущий блок в качестве альтернативы внешнему предсказанию, выполняемому блоком 42 оценки движения и блоком 44 компенсации движения, как описано выше. В частности модуль 46 внутрикадрового предсказания может определять режим внутрикадрового предсказания, чтобы использовать для кодирования текущего блока. В некоторых примерах модуль 46 внутрикадрового предсказания может кодировать текущий блок, используя различные режимы внутрикадрового предсказания, например, в течение отдельных проходов кодирования, и модуль 46 внутрикадрового предсказания (или блок 40 выбора режима, в некоторых примерах) может выбрать надлежащий режим внутрикадрового предсказания для использования, из протестированных режимов. Например, модуль 46 внутрикадрового предсказания может вычислять значения скорость-искажение, используя анализ скорость-искажение для различных протестированных режимов внутрикадрового предсказания, и выбирать режим внутрикадрового предсказания, имеющий наилучшие характеристики скорость-искажение из числа протестированных режимов. Анализ скорость-искажение в целом определяет величину искажения (или ошибку) между кодированным блоком и исходным, некодированным блоком, который был кодирован для создания кодированного блока, а также битовую скорость (то есть количество битов), используемых для создания кодированного блока. Модуль 46 внутрикадрового предсказания может вычислять отношения исходя из искажений и скоростей для различных кодированных блоков, чтобы определить, какой режим внутрикадрового предсказания показывает наилучшее значение скорость-искажение для блока.

После выбора режима внутрикадрового предсказания для блока модуль 46 внутрикадрового предсказания может предоставить информацию, указывающую выбранный режим внутрикадрового предсказания для блока, на блок 56 энтропийного кодирования. Блок 56 энтропийного кодирования может кодировать информацию, указывающую выбранный режим внутрикадрового предсказания, в соответствии со способами данного раскрытия. Видеокодер 20 может включать передаваемый битовый поток данные конфигурации, которые могут включать в себя ряд таблиц индексов режимов внутрикадрового предсказания, и ряд модифицированных таблиц индексов режимов внутрикадрового предсказания (также называемых таблицами отображения кодовых слов), определения контекстов кодирования для различных блоков, и указания наиболее вероятного режима внутрикадрового предсказания, таблицу индексов режима внутрикадрового предсказания и модифицированную таблицу индексов режима внутрикадрового предсказания для использования для каждого из контекстов.

После формирования модулем 41 предсказания предсказанного блока для текущего видеоблока посредством либо внешнего предсказания, либо внутрикадрового предсказания, видеокодер 20 формирует остаточный видеоблок, вычитая предсказанный

блок из текущего видеоблока. Остаточные видеоданные в остаточном блоке могут включаться в один или большее число TU и подаваться на модуль 52 преобразования. Модуль 52 преобразования преобразовывает остаточные видеоданные в остаточные коэффициенты преобразования, используя преобразование, такое как дискретное косинусное преобразование (DCT) или концептуально сходное преобразование. Модуль 52 преобразования, может преобразовывать остаточные видеоданные из области пикселов в область преобразования, такую как частотная область.

Модуль 52 преобразования может посылать результирующие коэффициенты преобразования на блок 54 квантования. Блок 54 квантования квантует коэффициенты преобразования, чтобы дополнительно снизить скорость передачи битов. Процесс квантования может снизить битовую глубину, связанную с некоторыми или всеми из коэффициентов. Степень квантования может быть изменена путем регулировки параметра квантования. В некоторых примерах блок 54 квантования может затем выполнять сканирование матрицы, включающей в себя квантованные коэффициенты преобразования. Альтернативно, блок 56 энтропийного кодирования может выполнять сканирование.

После квантования блок 56 энтропийного кодирования энтропийно кодирует квантованные коэффициенты преобразования. Например, блок 56 энтропийного кодирования может выполнять контекстно-зависимое адаптивное кодирование с переменной длиной слова (CAVLC), контекстно-зависимое адаптивное двоичное арифметическое кодирование (CABAC), синтаксическое контекстно-зависимое адаптивное двоичное арифметическое кодирование (SBAC), вероятностное с разбиением на интервалы энтропийное кодирование (PIPE) или другую методику или способ энтропийного кодирования. После энтропийного кодирования блоком 56 энтропийного кодирования, кодированный битовый поток может передаваться на видеодекодер 30 или архивироваться для более поздней передачи или извлечения видеодекодером 30. Блок 56 энтропийного кодирования может также энтропийно кодировать векторы движения и другие синтаксические элементы для текущего видео слайса, являющегося кодируемым.

Блок 58 обратного квантования и модуль 60 обратного преобразования применяют обратное квантование и обратное преобразование, соответственно, чтобы восстановить остаточный блок в пиксельной области для более позднего использования в качестве опорного блока для опорного изображения. Блок 44 компенсации движения может вычислять опорный блок путем суммирования остаточного блока с предсказанным блоком одного из опорных изображений в рамках одного списка из списков опорных изображений. Блок 44 компенсации движения может также применять один или большее число интерполяционных фильтров к восстановленному остаточному блоку, чтобы вычислять суб-пиксельные целые значения для использования в оценке движения. Сумматор 62 суммирует восстановленный остаточный блок с скомпенсированным по движению предсказанным блоком, выдаваемым блоком 44 компенсации движения, чтобы выдать опорный блок для сохранения в буфере 64 декодированных изображений. Опорный блок может использоваться блоком 42 оценки движения и блоком 44 компенсации движения в качестве опорного блока для внешнего предсказания блока в последующем видеокадре или изображении.

В соответствии с этим раскрытием, модуль 41 предсказания представляет один пример блока для выполнения примерных функций, описанных выше. Например, модуль 41 предсказания может определять, какие опорные изображения принадлежат набору опорных изображений, и предписывать видеокодеру 20 кодировать информацию,

указывающую опорные изображения, которые принадлежат набору опорных изображений. Кроме того, в течение процесса восстановления (например, процесса, используемого для восстановления изображения для использования в качестве опорного изображения и сохранения в буфере 64 декодированных изображений), модуль 41 предсказания может строить множество поднаборов опорных изображений, каждый из которых идентифицирует одно или более опорных изображений. Модуль 41 предсказания может также получать набор опорных изображений на основе построенной множества поднаборов опорных изображений. Кроме того, модуль 41 предсказания может осуществлять любой один или несколько из наборов примерного псевдокода, описанных выше, чтобы осуществлять один или большее число примерных способов, описанных в этом раскрытии.

В некоторых примерах модуль 41 предсказания может строить начальные списки опорных изображений описанным выше образом. В некоторых примерах не требуется переупорядочение изображений, подлежащих включению в начальные списки опорных изображений. Кроме того, модуль 41 предсказания может строить списки опорных изображений таким образом, что отсутствуют незаполненные элементы в списках опорных изображений. В некоторых примерах модуль 41 предсказания может также модифицировать начальные списки опорных изображений описанным выше образом, чтобы построить модифицированный список опорных изображений. Кроме того, в некоторых примерах модуль 41 предсказания может осуществлять удаление декодированных изображений из DPB 64 описанным выше образом. Кроме того, в некоторых примерах, модуль 41 предсказания может быть сконфигурирован для определения, какие долгосрочные опорные изображения принадлежат набору опорных изображений для текущего изображения, описанным выше образом.

В других примерах модуль, отличный от модуля 41 предсказания, может осуществлять примеры, описанные выше. В некоторых других примерах модуль 41 предсказания совместно с одним или несколькими другими модулями видеокодера 20 может осуществлять примеры, описанные выше. В еще некоторых других примерах процессор или видеокодера 20 (не показан на Фиг. 3) может один или совместно с другими блоками видеокодера 20 осуществлять примеры, описанные выше.

На Фиг. 4 показана блок-схема, иллюстрирующая примерный видеодекодер 30, который может осуществлять способы, описанные в этом раскрытии. В примере по Фиг. 4 видеодекодер 30 включает в себя блок 80 энтропийного декодирования, модуль 81 предсказания, блок 86 обратного квантования, блок обратного преобразования 88, сумматор 90 и буфер 92 декодированных изображений (DPB). Модуль 81 предсказания включает в себя блок 82 компенсации движения и модуль 84 внешнего предсказания. Видеодекодер 30 может, в некоторых примерах, выполнять проход декодирования в целом обратно к проходу кодирования, описанному относительно видеокодера 20 по Фиг. 3.

В течение процесса декодирования видеодекодер 30 принимает битовый поток кодированного видео, который представляет видеоблоки для кодированного видеослайса и связанные синтаксические элементы от видеокодера 20. Модуль 80 энтропийного декодирования в видеодекодере 30 энтропийно декодирует битовый поток, чтобы сформировать квантованные коэффициенты, векторы движения и другие синтаксические элементы. Модуль 80 энтропийного декодирования пересылает векторы движения и другие синтаксические элементы на модуль 81 предсказания. Видеодекодер 30 может принимать синтаксические элементы на уровне видеослайса и/или уровне видеоблока.

Когда видео слайс кодируется как слайс с внутренним кодированием (I), модуль 84 внешнего предсказания в модуле 81 предсказания может сформировать данные предсказания для видеоблока в текущем видео слайсе на основании сигнализированного режима внутрикадрового предсказания и данных от предварительно декодированных блоков из текущего изображения. Когда видеоизображение кодируется как слайс с внешнем кодированием (то есть B или P), блок 82 компенсации движения в модуле 81 предсказания выдает предсказанные блоки для видеоблока для текущего видео слайса, на основании векторов движения и других синтаксических элементов, принятых от модуля 80 энтропийного декодирования. Предсказанные блоки могут создаваться на основе одного из опорных изображений в одном из списков опорных изображений. Видеодекодер 30 может построить списки опорных кадров, List 0 и List 1, используя способы построения «по умолчанию» на основании опорных изображений, сохраненных в буфере 92 декодированных изображений. В некоторых примерах видеодекодер 30 может построить List 0 и List 1 на основе опорных изображений, идентифицированных в полученном наборе опорных изображений.

Блок 82 компенсации движения определяет информацию предсказания для видеоблока текущего видео слайса путем синтаксического анализа векторов движения и других синтаксических элементов, и использует информацию предсказания, чтобы выдать предсказанные блоки для текущего декодируемого видеоблока. Например, блок 82 компенсации движения использует некоторые из принятых синтаксических элементов, чтобы определять режим предсказания (например, внутри- или меж кадровое предсказание), используемый для кодирования видеоблоков для видео слайса, тип слайса для внешнего предсказания (например, B-слайс или P-слайс), информацию построения для одного или более списков опорных изображений для слайса, векторы движения для каждого внешне-кодированного блока видео слайса, состояние внешнего предсказания для каждого внешне- кодированного блока видео слайса, и другую информацию, чтобы декодировать видеоблоки в текущем видео слайсе.

Блок 82 компенсации движения может также выполнять интерполяцию на основе интерполяционных фильтров. Блок 82 компенсации движения может использовать интерполяционные фильтры, как используются видеокодером 20 в течение кодирования видеоблоков, чтобы вычислять интерполированные значения для суб-пиксельных целых для опорных блоков. В этом случае блок 82 компенсации движения может определять интерполяционные фильтры, используемые видеокодером 20, исходя из принятых синтаксических элементов, и использовать интерполяционные фильтры, чтобы выдавать предсказанные блоки.

Блок 86 обратного квантования обращенно квантует, то есть выполняет обратное квантование квантованных коэффициентов преобразования, обеспеченных в битовом потоке и декодированных блоком 80 энтропийного декодирования. Процесс обратного квантования может включать в себя использование параметра квантования, вычисленного видеокодером 20 для каждого видеоблока в видео слайсе, чтобы определить степень квантования и, аналогично, степень обратного квантования, которая подлежит применению. Модуль 88 обратного преобразования применяет обратное преобразование, например, обратное DCT, обратное целочисленное преобразование, или концептуально подобный процесс обратного преобразования, к коэффициентам преобразования, чтобы создать остаточные блоки в пиксельной области.

После формирования модулем 81 предсказания предсказанного блока для текущего видеоблока на основании либо меж-, либо внутрикадрового предсказания видеодекодер 30 формирует декодированный видеоблок, суммируя остаточные блоки от модуля 88

обратного преобразования с соответствующими предсказанными блоками, сформированными модулем 81 предсказания. Сумматор 90 представляет компонент или компоненты, которые выполняют эту операцию суммирования. Если требуется, деблокирующий фильтр также может применяться, чтобы фильтровать декодированные блоки для удаления артефактов блочности. Другие контурные фильтры (либо в цикле кодирования, или после цикла кодирования) также могут быть использованы для сглаживания переходов пикселов, или иначе повышения качества видео. Декодированные видеоблоки в данном изображении затем сохраняются в буфере 92 декодированных изображений, который хранит опорные изображения, используемые для последующей компенсации движения. Буфер 92 декодированных изображений также хранит декодированное видео для последующего представления на устройстве отображения, таком как устройство 32 отображения по Фиг. 1.

В соответствии с этим раскрытием, модуль 81 предсказания представляет один примерный модуль выполнения примерных функций, описанные выше. Например, модуль 81 предсказания может определять, какие опорные изображения принадлежат набору опорных изображений. Кроме того, модуль 81 предсказания может построить множество поднаборов опорных изображений, так что каждое идентифицирует одно или более опорных изображений. Модуль 81 предсказания может также получать набор опорных изображений на основе построенной множества поднаборов опорных изображений. Кроме того, модуль 81 предсказания может осуществлять один любое или большее число наборов примеров псевдокода, описанных выше, для осуществления одного или более примерных способов, описанных в этом раскрытии.

В некоторых примерах модуль 81 предсказания может построить начальные списки опорных изображений описанным выше образом. В некоторых примерах не требуется переупорядочение изображений, подлежащих включению в начальные списки опорных изображений. Кроме того, модуль 81 предсказания может построить списки опорных изображений таким образом, что отсутствуют незаполненные элементы в списках опорных изображений. В некоторых примерах модуль 81 предсказания может также модифицировать начальные списки опорных изображений описанным выше образом, чтобы построить модифицированный список опорных изображений. Кроме того в некоторых примерах модуль 81 предсказания может осуществлять удаление декодированных изображений из DPB 94 описанным выше образом. Кроме того в некоторых примерах модуль 81 предсказания может быть сконфигурирован для определения, какие долгосрочные опорные изображения принадлежат набору опорных изображений для текущего изображения, описанным выше образом.

В других примерах блок, отличный от модуля 81 предсказания, может осуществлять примеры, описанные выше. В некоторых других примерах модуль 81 предсказания совместно с одним или несколькими другими модулями видеодекодера 30 может осуществлять примеры, описанные выше. В еще некоторых других примерах процессор или устройство видеодекодера 30 (не показано на Фиг. 4) может одно или совместно с другими модулями видеодекодера 30 осуществлять примеры, описанные выше.

На Фиг. 5 показана блок-схема, иллюстрирующая примерную операцию получения набора опорных изображений. С целью лишь иллюстрации способ по Фиг. 5 может выполняться видеокодером, соответствующим либо видеокодеру 20, либо видеодекодеру 30. Например, кодер видео (например, видеокодер 20 или видеодекодер 30) может кодировать (например, закодировать или декодировать) информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений (94). Набор опорных изображений может идентифицировать опорные изображения, которые

потенциально могут быть использованы для внешнего предсказания текущего изображения и для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования.

Например, когда видеокодер 20 выполняет этап 94, видеокодер 20 может закодировать значения, которые указывают идентификаторы для опорных изображений, которые принадлежат набору опорных изображений. Например, видеокодер 20 может сигнализировать в битовом потоке синтаксический элемент pic_order_cnt_lsb и синтаксический элемент log2_max_pic_order_cnt_lsb_minus4. Когда видеодекодер 30 выполняет этап 94, из синтаксического элемента log2_max_pic_order_cnt_lsb_minus4 видеодекодер 30 может определить значение MaxPicOrderCntLsb. Видеодекодер 30 может затем определить идентификаторы (например, значения POC) для опорных изображений, которые принадлежат набору опорных изображений.

Кодер видео может построить множество поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений (96). Например, кодер видео может построить поднабора опорных изображений RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr и RefPicSetLtFoll. Однако, аспекты данного раскрытия не являются этим самым ограниченными. В некоторых примерах кодер видео может построить пять поднаборов опорных изображений, четыре из которых могут быть четырьмя из поднаборов опорных изображений RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr и RefPicSetLtFoll и пятое может быть комбинацией двух из оставшихся шести поднаборов опорных изображений (например, комбинацией поднаборов опорных изображений RefPicSetFoll0 и RefPicSetFoll1).

В некоторых примерах кодер видео может построить по меньшей мере два из следующих четырех поднаборов опорных изображений. В других примерах кодер видео может построить, по меньшей мере, следующие четыре поднабора опорных изображений. Первый поднабор опорных изображений может идентифицировать краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и до текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения, и одно или несколько из одного или более изображений, следующих после текущего изображения в очередности декодирования. Второй поднабор опорных изображений может идентифицировать краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и после текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из одного или более изображений, следующих после текущего изображения в очередности декодирования.

Третий поднабор опорных изображений может идентифицировать долгосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования, и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из одного или более изображений, следующих после текущего изображения в очередности декодирования. Четвертое поднабор опорных изображений может идентифицировать долгосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования которые не могут быть использованы для внешнего предсказания текущего изображения, и потенциально могут быть использованы для внешнего предсказания одного или более из одного или более изображений, следующих после

текущего изображения в очередности декодирования.

Кодер видео может получать набор опорных изображений на основе множества поднаборов опорных изображений (98). Например, кодер видео может упорядочить по меньшей мере два из поднаборов опорных изображений RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr и RefPicSetLtFoll в конкретном порядке, чтобы получить набор опорных изображений.

В некоторых примерах упорядочение, выполняемое кодером видео, может означать, что изображения в каждом из поднаборов опорных изображений могут быть идентифицированы последовательно в рамках набора опорных изображений. В этих примерах кодер видео может обращаться к опорным изображениям в наборе опорных изображений по значению индекса в наборе опорных изображений.

Кодер видео может осуществлять кодирование текущего изображения на основании полученного набора опорных изображений (100). Следует понимать, что поскольку кодер видео получает набор опорных изображений в поднаборах опорных изображений, кодер видео можно рассматривать в качестве кодирующего текущее изображение на основании множества поднаборов опорных изображений. Например, кодер видео может построить, по меньшей мере, один список из первого списка опорных изображений и второго списка опорных изображений на основании множества поднаборов опорных изображений (например, из полученного набора опорных изображений, которое получено из множества поднаборов опорных изображений). Кодер видео может затем осуществлять кодирование текущего изображения на основании по меньшей мере одного списка из первого списка опорных изображений и второго списка опорных изображений.

На Фиг. 6 показана блок-схема, иллюстрирующая примерную операцию построения списка опорных изображений. С целью только иллюстрации, способ по Фиг. 6 может выполняться кодером видео, соответствующим либо видеокодеру 20, либо видеодекодеру 30. Подобно Фиг. 5, кодер видео может осуществлять кодирование информации, указывающей опорные изображения (102) и строить множество поднаборов опорных изображений (104).

Кодер видео может затем добавлять опорные изображения из поднаборов опорных изображений в начальный список опорных изображений, чтобы построить начальный список опорных изображений (106). В некоторых примерах и видеокодер 20, и видеодекодер 30 могут строить начальный список опорных изображений. Например, видеокодер 20 может строить начальный список опорных изображений, чтобы создать восстановленные видеоблоки для сохранения в DPB 64. Видеодекодер 30 может строить начальный список опорных изображений в качестве части своего процесса декодирования, и может осуществлять способ построения «по умолчанию», в котором видеодекодеру 30 не требуется принимать информацию от видеокодера 20 относительно способа, которым можно строить начальный список опорных изображений.

В некоторых примерах, для построения начального списка опорных изображений кодер видео может добавлять опорные изображения из первого поднабора в множества поднаборов опорных изображений в начальный список опорных изображений, за которыми следуют опорные изображения из второго поднабора в начальный список опорных изображений, и затем за которыми следуют опорные изображения из третьего поднабора в начальный список опорных изображений. Кодер видео может добавлять опорные изображения из этих поднаборов опорных изображений, пока общее число опорных изображений, внесенных в начальный список опорных изображений, не больше максимального числа допустимых элементов в начальном списке опорных изображений.

Например, если в какой-либо момент в течение добавления опорных изображений в список опорных изображений, число элементов в начальном списке опорных изображений становится равным максимальному числу допустимых элементов начальных списка опорных изображений, кодер видео может остановить добавление каких-либо дополнительный изображений в начальный список опорных изображений.

Кодер видео может подобным образом строить другой начальный список опорных изображений, такой как в примерах, где видеоблок для текущего изображения является двунаправленно предсказанным. В этом примере, для построения этого другого начального списка опорных изображений, кодер видео может добавлять опорные изображения из второго поднабора в другой начальный список опорных изображений, за которыми следуют опорные изображения из первого поднабора в другой начальный список опорных изображений, и затем за которыми следуют опорные изображения из третьего поднабора в другой начальный список опорных изображений, пока общее число элементов в этом другом начальном списке опорных изображений не больше допустимого числа элементов. В этих примерах первый поднабор может быть поднабором RefPicSetStCurr0опорных изображений, второй поднабор может быть поднабором RefPicSetStCurr1опорных изображений и третий поднабор может быть поднабором RefPicSetLtCurr опорных изображений.

В некотором примере, чтобы добавлять опорные изображения, идентифицированные в поднаборах RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений, кодер видео может кодировать (например, закодировать или декодировать) синтаксические элементы, исходя из которых кодер видео может определять число опорных изображений в каждом из этих поднаборов опорных изображений. Например, кодер видео может кодировать синтаксический элемент num_short_term_curr0 и синтаксический элемент num_short_term_curr1. Синтаксический элемент num_short_term_curr0 и синтаксический элемент num_short_term_curr1 могут указывать число опорных изображений, идентифицированных в поднаборе опорных изображений RefPicSetStCurr0 и поднаборе опорных изображений RefPicSetStCurr1, соответственно.

Кодер видео может также кодировать синтаксический элемент num_long_term_pps_curr и синтаксический элемент num_long_term_add_curr. Синтаксический элемент num_long_term_pps_curr может указывать число долгосрочных опорных изображений, идентификационная информация которых включена в набор параметров изображения (PPS), и синтаксический элемент num_long_term_add_curr может указывать число долгосрочных опорных изображений, идентификационная информация которых не включена в PPS. В этом примере эти долгосрочные опорные изображения потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования.

Кодер видео может определять число опорных изображений в поднаборе RefPicSetLtCurr опорных изображений на основании синтаксического элемента num_long_term_pps_curr и синтаксического элемента num_long_term_add_curr. Например, кодер видео может суммировать значения синтаксического элемента num_long_term_pps_curr и синтаксического элемента num_long_term_add_curr, чтобы определить число опорных изображений в поднаборе опорных изображений RefPicSetLtCurr.

Кодер видео может осуществлять кодирование текущего изображения на основании списка или списков опорных изображений (108). Например, кодер видео может построить, по меньшей мере, один список из первого списка опорных изображений и

второго списка опорных изображений на основании полученного набора опорных изображений. Кодер видео может затем осуществлять кодирование текущего изображения на основании по меньшей мере одного списка из первого списка опорных изображений и второго списка опорных изображений.

На Фиг. 7 показана блок-схема, иллюстрирующая другую примерную операцию построения списка опорных изображений. С целью только иллюстрации, способ по Фиг. 7 может выполняться кодером видео, соответствующим или видеокодеру 20, или видеодекодеру 30. Подобно Фиг. 5, кодер видео может осуществлять кодирование информации, указывающей опорные изображения (110), и строить множество поднаборов опорных изображений (112). Кодер видео может добавлять опорные изображения в первый набор элементов списка опорных изображений (114). Например, число элементов в списке опорных изображений может равняться максимальному числу допустимых элементов, как определено синтаксическими элементами num_ref_idx_l0_active_minus1 или num_ref_idx_l1_active_minus1.

В этом примере кодер видео может вносить в список (например, добавлять) в первый набор элементов опорные изображения, идентифицированные в поднаборах опорных изображений RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr. Например, для List 0, кодер видео может добавлять в первый набор элементов в List 0, по порядку, опорные изображения, идентифицированные в поднаборах опорных изображений RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr. Для List 1, кодер видео может добавлять в первый набор элементов в List 1, по порядку, опорные изображения, идентифицированные в поднаборах опорных изображений RefPicSetStCurr1, RefPicSetStCurr0 и RefPicSetLtCurr.

Кодер видео может затем определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений (116). Если число элементов в списке опорных изображений не меньше максимального числа допустимых элементов ("НЕТ" 116), кодер видео может осуществлять кодирование текущего изображения на основании списка опорных изображений (118).

Иначе, если число элементов в списке опорных изображений меньше максимального числа допустимых элементов ("ДА" 116), кодер видео может повторно вносить в список (например, повторно идентифицировать или повторно добавлять), одно или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов (120). Например, кодер видео может добавлять одно или более опорных изображений, идентифицированных в поднаборе опорных изображений RefPicSetStCurr0, в List 0 в элементы после первого набора элементов в List 0, или одно или более опорных изображений, идентифицированных в поднаборе опорных изображений RefPicSetStCurr1, в List 1 в элементы после первого набора элементов в List 1. Таким образом кодер видео может идентифицировать, по меньшей мере, одно опорное изображение из первого поднабора опорных изображений в более чем одном элементе в списке опорных изображений.

Кодер видео может затем определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений (122). Если число элементов в списке опорных изображений не меньше максимального числа допустимых элементов ("НЕТ" 122), кодер видео может осуществлять кодирование текущего изображения на основании списка опорных изображений (124).

Иначе, если число элементов в списке опорных изображений меньше максимального

числа допустимых элементов ("ДА" 122), кодер видео может повторно внести в список (например, повторно идентифицировать или повторно добавлять) одно или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов (120). Например, в этой ситуации, кодер видео может повторно добавлять дополнительные опорные изображения, идентифицированные в первом поднаборе опорных изображений. Если, кодер видео уже повторно добавил все опорные изображения из первого поднабора опорных изображений, кодер видео может повторно добавлять одно или более опорных изображений из второго поднабора опорных изображений (например, поднабора опорных изображений RefPicSetStCurr0 для List 0, или поднабора опорных изображений RefPicSetStCurr1 для List 1). Этот процесс можно повторять, пока число элементов в списке опорных изображений не меньше максимального числа допустимых элементов ("НЕТ" 122).

Таким образом, если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, кодер видео может многократно вносить в список (например, повторно идентифицировать или повторно добавлять) одно или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений. Это может вызывать, что кодер видео вносит опорные изображения в элементы списка опорных изображений так, что каждый элемент списка опорных изображений идентифицирует одно из опорных изображений, и так, что по меньшей мере два элемента списка опорных изображений идентифицируют одно и то же опорное изображение из опорных изображений.

На Фиг. 8 показана блок-схема, иллюстрирующая примерную операцию модифицирования начального списка опорных изображений. С целью только иллюстрации, способ по Фиг. 8 может выполняться кодером видео, соответствующим или видеокодеру 20, или видеодекодеру 30. Подобно Фиг. 6, кодер видео может осуществлять кодирование информации, указывающей опорные изображения (126), и строить множество поднаборов опорных изображений (128). Множество поднаборов опорных изображений может включать в себя поднаборы RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений. Кодер видео может строить начальный список опорных изображений описанным выше образом на основании поднаборов RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr опорных изображений (130).

Кодер видео может определять, требуется ли модификация списка опорных изображений (132). Например, кодер видео может осуществлять кодирование синтаксических элементов, таких как ref_pic_list_modification_flag_l0 и ref_pic_list_modification_flag_l1, которые указывают, необходимо ли, чтобы были модифицированы начальный List 0 или начальный List 1. Если модификация начального списка опорных изображений не требуется ("НЕТ" 132), кодер видео может осуществлять кодирование текущего изображения на основании начального списка опорных изображений (134).

Если модификация требуется ("ДА" 132), кодер видео может идентифицировать опорное изображение в, по меньшей мере, одном из построенных поднаборов опорных изображений (136). Например, кодер видео может осуществлять кодирование синтаксического элемента modification_of_ref_pic_idc. Значение синтаксического элемента modification_of_ref_pic_idc может указывать, какой поднабор опорных изображений

кодер видео должен использовать для идентификации опорного изображения. Кодер видео может также осуществлять кодирование синтаксического элемента ref_pic_set_idx, который указывает индекс в поднаборе опорных изображений, которое кодер видео должен использовать для идентификации опорного изображения.

Кодер видео может вносить в список (например, добавлять или идентифицировать) идентифицированное опорное изображение в начальном списке опорных изображений в текущий элемент, чтобы строить модифицированный список опорных изображений (138). Текущий элемент может первоначально быть элементом в начальном списке опорных изображений, заданным индексом 0. Для каждого экземпляра синтаксического элемента modification_of_ref_pic_idc в кодированном битовом потоке, где значением синтаксического элемента modification_of_ref_pic_idc не является 3, кодер видео может увеличивать на единицу значение начального элемента (например, следующим значением элемента будет задаваться индексом 1). Кодер видео может осуществлять кодирование текущего изображения на основании модифицированного списка опорных изображений (140).

На Фиг. 9 показана блок-схема, иллюстрирующая примерную операцию удаления декодированного изображения. С целью только иллюстрации, способ по Фиг. 8 может выполняться кодером видео, соответствующим или видеокодеру 20, или видеодекодеру 30. Подобно Фиг. 5, кодер видео (например, видео кодер 20 или видеодекодер 30) может кодировать (например, закодировать или декодировать) информацию, указывающую опорные изображения для набора опорных изображений (142), и может получать набор опорных изображений из кодированной (например, закодированной или декодированной) информации (144).

Кодер видео может определять, является ли декодированное изображение, сохраненное в буфере декодированных изображений (DPB), не требуемым для вывода (то есть уже выведено или не предназначено для вывода) и не идентифицированным в наборе опорных изображений (146). Если декодированное изображение требуется для вывода или идентифицировано в наборе опорных изображений ("НЕТ" 146), кодер видео может не удалять декодированное изображение из DPB, и может поддерживать декодированное изображение сохраняемым в DPB (148).

Если декодированное изображение было выведено и не является идентифицированным в наборе опорных изображений ("ДА" 146), кодер видео может удалить декодированное изображение из DPB (150). Кодер видео может затем осуществлять кодирование текущего изображения после удаления декодированного изображения (152).

Как описано выше, кодер видео может строить список опорных изображений на основании набора опорных изображений для кодирования текущего изображения. В некоторых примерах кодер видео может удалять декодированное изображение из DPB после построения списка опорных изображений. Кроме того, для выводимого декодированного изображения кодер видео может определять время, когда выводить декодированное изображение, и может выводить декодированное изображение на основании определенного времени и до кодирования текущего изображения.

В некоторых примерах кодер видео может сохранять кодированное (например, декодированное) изображение в DPB. В некоторых примерах кодер видео может определять, что DPB полон, до сохранения. В этих примерах кодер видео может выбирать декодированное изображение, в текущий момент хранимое в DPB, которое маркировано как “требуемое для вывода” и имеющее наименьшее значение счетчика очередности изображения (POC) из всех декодированных изображений, сохраненных в DPB. Кодер видео может затем выводить выбранное изображение. Кроме того, кодер

видео может определять, что выводимое изображение не включено в набор опорных изображений для текущего изображения. В этом случае, кодер видео может очистить буфер в DPB, который хранил выводимое изображение, и может сохранить кодированное изображение в этом буфере в DPB.

На Фиг. 10 показана блок-схема, иллюстрирующая примерную операцию определения, какие долгосрочные опорные изображения принадлежат набору опорных изображений для текущего изображения. С целью только иллюстрации, способ по Фиг. 10 может выполняться кодером видео, соответствующим или видеокодеру 20, или видеодекодеру 30.

Кодер видео (например, видеокодер 20, или видеодекодер 30) может кодировать синтаксические элементы, которые указывают долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров (154). В некоторых примерах, одно или более долгосрочных опорных изображений-кандидатов находятся в наборе опорных изображений для текущего изображения. В соответствии со способами, описанными в этом раскрытии, набор параметров может быть набором параметров последовательности или набором параметров изображения. В некоторых примерах для кодирования синтаксических элементов, которые указывают долгосрочные опорные изображения-кандидаты, кодер видео может осуществлять кодирование списка значений POC для долгосрочных опорных изображений-кандидатов в наборе параметров, и осуществлять кодирование значения индекса для списка в заголовке слайса для текущего изображения.

Кодер видео может осуществлять кодирование синтаксических элементов, которые указывают, какие долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров, находятся в наборе опорных изображений для текущего изображения (156). Например, кодер видео может кодировать в заголовке слайса текущего изображения синтаксические элементы, которые указывают, какие долгосрочные опорные изображения-кандидаты находятся в наборе опорных изображений.

В некоторых примерах не все долгосрочные опорные изображения, которые находятся в наборе опорных изображений, включены в долгосрочные опорные изображения-кандидаты. В этих примерах кодер видео может дополнительно кодировать синтаксические элементы, которые указывают долгосрочные опорные изображения, которые находятся в наборе опорных изображений (158). Кодер видео может затем построить по меньшей мере один поднабор из множества поднаборов опорных изображений на основании указания, какие долгосрочные опорные изображения-кандидаты находятся в наборе опорных изображений для текущего изображения (160).

В вышеуказанных примерах описываются различные примеры наряду с возможными альтернативами примерам. Последующее описывает некоторые дополнительные альтернативы примерам, которые могут рассматриваться альтернативными примерам, к одному или более примерам, описанным выше. Кроме того, эти альтернативные примеры могут быть использованы совместно с примерами, описанными выше, или отдельно от примеров, описанных выше.

Например, уже описанные выше альтернативные примеры для понятия набора опорных изображений, альтернативные примеры для сигнализации набора опорных изображений в наборах параметров, альтернативные примеры поднаборов для набора опорных изображений и альтернативные примеры для маркировки опорного изображения. Последующее обеспечивает дополнительные альтернативные примеры.

Например, для заголовка модуля NAL, в вышеуказанных примерах, синтаксис

заголовка модуля NAL может включать в себя синтаксические элементы nal_ref_idc, temporal_id и output_flag. В одном альтернативном примере, nal_ref_idc (2 бита) заменено nal_ref_flag (1 бит). В этом примере, nal_ref_flag, равный 1, имеет семантику, как у nal_ref_idc больше 0, и nal_ref_flag, равный 0, имеет такую же семантику, как у nal_ref_idc, равного 0. В одном альтернативном примере nal_ref_idc (2 бита) удален. Определение опорного изображения может быть изменено на: изображение, которое содержит выборки, которые могут быть использованы для внешнего предсказания в процессе декодирования последующих изображений в очередности декодирования. В одном альтернативном примере temporal_id не присутствует в синтаксисе заголовка модуля NAL, и значение выводят являющимся одинаковым для всех изображений. В одном альтернативном примере output_flag не присутствует в синтаксисе заголовка модуля NAL, и значение полагают равным 1 для всех изображений.

Для сигнализации и вычисления счетчика очередности изображения, вышеуказанные способы могут использовать один способ сигнализации и вычисления счетчика очередности изображения, который может быть подобным типу 0 счетчика очередности изображения в AVC. Два альтернативных способа сигнализации и вычисления счетчика очередности изображения, такие же, как типы 1 и 2 счетчика очередности изображения в AVC, соответственно, могут применяться. Любая комбинация двух из этих трех способов, или комбинация всех трех способов также может применяться.

Для идентификации изображения вышеуказанные способы могут использовать значение счетчика очередности изображения (POC) для идентификации изображения. В одном альтернативном примере временная ссылка (TR) используется в качестве идентификации изображения. Один способ задания TR состоит в том, что оно является одинаковым с POC, когда POC ограничивается так, что разность POC между любыми двумя изображениями является пропорциональной разности времени представления или времени выборки. В одном альтернативном примере frame_num, который указывает порядок декодирования (тогда как POC указывает очередность вывода или отображения), или переменная (например, именуемая UnWrappedFrameNum), которая может быть любым значением из значений 32-разрядного целого без знака, полученная из frame_num, может использоваться в качестве идентификации изображения. В основном, UnWrappedFrameNum может быть разупакованной версией frame_num. Например, если frame_num представлено 8 битами, максимальным значением frame_num является 255. Следующим значением после 255 для frame_num является 0. Значение UnWrappedFrameNum продолжает увеличиваться в позициях, в которых frame_num возвращается в 0.

Для сигнализации долгосрочных опорных изображений в наборах параметров, в вышеуказанных примерах, видеокодер 20 может сигнализировать список абсолютной идентификационной информации долгосрочного опорного изображения в наборе параметров изображения, может сигнализировать информацию о длине, в битах, абсолютной идентификационной информации долгосрочного опорного изображения, и на индекс к списку можно ссылаться в заголовке слайса, таким образом чтобы снижать издержки сигнализации. В одном альтернативном примере список отличительной идентификационной информации долгосрочного опорного изображения может сигнализироваться в наборе параметров последовательности, и на индекс к списку можно ссылаться в заголовке слайса, таким образом чтобы снизить издержки сигнализации. В одном альтернативном примере может сигнализироваться информация о длине, в битах, абсолютной идентификационной информации долгосрочного опорного изображения, но длину можно считать константой, например, 32. В различных примерах

применяется комбинация любого из вышеуказанных примеров.

Для сигнализации краткосрочных опорных изображений, в вышеуказанных примерах для краткосрочных опорных изображений набора опорных изображений для кодированного изображения, либо все сигнализируются в указываемом наборе параметров изображения, либо все сигнализируются в заголовке слайса. В одном альтернативном примере, для краткосрочных опорных изображений набора опорных изображений для кодированного изображения, некоторые сигнализируются в указываемом наборе параметров изображения, и другие сигнализируются в заголовке слайса.

Для инициализации списка опорных изображений, в одном альтернативном примере, в процессе инициализации списка опорных изображений, во-первых краткосрочные опорные изображения могут добавляться в список, и во-вторых, необязательно долгосрочные опорные изображения могут добавляться в список. После этого, если число элементов в списке опорных изображений (или RefPicList0, или RefPicList1) все еще меньше, чем значение num_ref_idx_lx_active_minus1+1 (lx является l0 или l1), оставшиеся элементы могут маркироваться “нет опорного изображения”. В одном альтернативном примере может также быть возможным, если список опорных изображений еще не является заполненным после добавления элементов в RefPicSetStCurr0 и RefPicSetStCurr1, и возможно долгосрочных опорных изображений, изображения в RefPicSetStFoll0 и/или RefPicSetStFoll1 могут добавляться. В одном альтернативном примере процесс инициализации списка опорных изображений может только добавлять краткосрочные опорные изображения в список опорных изображений. В таком случае долгосрочное опорное изображение может только добавляться к списку опорных изображений путем использования команд модификации списка опорных изображений, как сигнализируется в синтаксической таблице Модификации списка опорных изображений (RPLM).

Для модификации списка опорных изображений в одном альтернативном примере, модификацию списка опорных изображений может также сигнализировать ref_pic_set_idx отличительным образом, причем предшествующие индекс используется в качестве прогнозирующего параметра текущего индекса. В этом примере различные значения modification_of_ref_pic_idc могут соответствовать различным категориям индексации (RefPicSetStCurrx для RefPicListx, RefPicSetStCurrx для RefPicList (1-x), или RefPicSetLtCurr), каждый из которых поддерживает различный предсказанный параметр. Предсказанный параметр может быть обновлен, как только синтаксический элемент, принадлежащий той же категории, был непосредственно проанализирован. В одном альтернативном примере модификации списка опорных изображений могут быть основаны на разности номера изображения. В одном альтернативном примере модификации списка опорных изображений могут быть основаны на разности значений POC.

Для операций буфера декодированных изображений (DPB), в вышеуказанном примере, после декодирования текущего изображения и до анализа заголовка слайса следующего кодированного изображения в очередности декодирования, текущее декодированное изображение сохраняется в DPB. В одном альтернативном примере, после декодирования текущего изображения и до анализа заголовка слайса следующего кодированного изображения в очередности декодирования, текущее декодированное изображение сохраняется во временной памяти (не в DPB). Оно сохраняется в DPB после синтаксического анализа заголовка слайса следующего кодированного изображения в очередности декодирования и построения набора опорных изображений этого изображения, если оно все еще необходимо для опорного или для вывода. В этот момент,

если оно не требуется ни для опорного, ни для вывода, декодированное изображение может быть просто отброшено (из временного буфера).

Кроме того, в вышеуказанных примерах удаление декодированного изображения из DPB немедленно после синтаксического анализа заголовка слайса текущего изображения и до декодирования любого слайса текущего изображения. В одном альтернативном примере маркировка, если имеется и удаление декодированного изображения из DPB происходит после того, как текущее изображение декодируется полностью.

В вышеуказанных примерах поднаборы RefPicSetStCurr0 и RefPicSetStCurr1 набора опорных изображений для текущего изображения получают для всех декодированных изображений. Однако, это может не являться необходимым для изображения с внутренним кодированием. Для получения набора опорных изображений для изображения с внутренним кодированием, в одном альтернативном примере, для не IDR изображения, которое изображения с внутренним кодированием (то есть все слайсы кодированного изображения являются I-слайсами), RefPicSetStCurr0 и RefPicSetStCurr1 не получают, поскольку даже если они не являются пустыми после получения, они не требуются в декодировании кодированного изображения. Допущение непустых RefPicSetStCurr0 или RefPicSetStCurr1 для не-IDR изображения с внутренним кодированием может позволить совместное использование экземпляра синтаксической структуры short_term_ref_pic_set() для одного или большего количества изображения с внешнем кодированием, для которых RefPicSetStCurr0 и RefPicSetStCurr1, могут оба не являться пустыми.

Для обнаружения потерь возможны следующие различные способы для обнаружения потери опорного изображения или раннего обнаружения, можно ли корректно декодировать текущее изображение. В различных примерах, после получения набора опорных изображений, видеодекодер 30 (например, стороны декодера) может проверять присутствие опорных изображений, включенных в RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr. Если какое-либо из опорных изображений, включенных в RefPicSetStCurr0, RefPicSetStCurr1 и RefPicSetLtCurr, не присутствует в DPB, сторона декодера может заключить, что это опорное изображение было потеряно, и что текущее изображение вероятно не будет корректно декодировано, и может предпринять некоторые меры, чтобы улучшить ситуацию, например, путем уведомления стороны кодера (например, видеокодера 20) о потере(ях) изображения, и кодер может повторно передать потерянное опорное изображение(я) или кодированное последующее изображение(я), используя только те опорные изображения, которые известны корректными на стороне декодера для внешней ссылки.

В различных примерах, после получения набора опорных изображений, сторона декодера может проверять присутствие опорных изображений, включенных в RefPicSetStFoll0, RefPicSetStFoll1 и RefPicSetLtFoll. Если какое-либо из опорных изображений, включенных в RefPicSetStFoll0, RefPicSetStFoll1 и RefPicSetLtFoll, не присутствует в DPB, сторона декодера может заключить, что это опорное изображение было потеряно, и что некоторые из последующих изображений в очередности декодирования будут вероятно некорректно декодированными, если только не будут предприняты некоторые меры, и могут предприниматься некоторые меры, чтобы исправить ситуацию, например, путем уведомления стороны кодера о потере(ях) изображения, и кодер может повторно передать потерянное опорное изображение(я) или кодированное следующее изображение(я), используя только те опорные изображения, которые известны корректными на стороне декодера для опорного

внешнего предсказания.

Для стороны кодера (например, видеокодера 20) составление набора опорных изображений, относительно вышеуказанных примеров, следующие различные способы для составления набора опорных изображений на стороне кодера могут быть возможными. Например, в различных примерах, кодер составляет относящиеся к набору опорных изображений синтаксические структуры, так что после получения набора опорных изображений на стороне декодера для текущего изображения: (1) RefPicSetStCurr0 включает в себя и включает только идентификационную информацию всех краткосрочных опорных изображений, которые имеют более раннюю очередность вывода, чем текущее изображение, и которые используются для опорного в внешнем предсказании для текущего изображения, (2) RefPicSetStCurr1 включает в себя и включает только идентификационную информацию всех краткосрочных опорных изображений, которые имеют более позднюю очередность вывода, чем текущее изображение, и которые используются для опорного в внешнем предсказании для текущего изображения, и (3) RefPicSetLtCurr включает в себя и включает только идентификационную информацию всех долгосрочных опорных изображений, которые используются для опорного в внешнем предсказании для текущего изображения.

В различных примерах кодер (например, видеокодер 20), может составлять относящиеся к набору опорных изображений синтаксические структуры так, что после получения набора опорных изображений на стороне декодера для текущего изображения: (1) RefPicSetStCurr0 включает в себя и включает только идентификационную информацию о 1) всех краткосрочных опорных изображениях, которые имеют более ранний порядок вывода, чем текущее изображение, и которые используются для опорного в внешнем предсказании для текущего изображения, а также 2) одном или нескольких краткосрочных опорных изображениях, которые имеют более раннюю очередность вывода, чем текущее изображение, и которые не используются для опорного в внешнем предсказании для текущего изображения, (2), RefPicSetStCurr1 включает в себя и включает только идентификационную информацию 1) всех краткосрочных опорных изображений, которые имеют более позднюю очередность вывода, чем текущее изображение, и которые используются для опорного в внешнем предсказании для текущего изображения, а также 2) одного или более краткосрочных опорных изображений, которые имеют более позднюю очередность вывода, чем текущее изображение, и которые не используются для опорного в внешнем предсказании для текущего изображения, и RefPicSetLtCurr включает в себя и включает только идентификационную информацию 1) всех долгосрочных опорных изображений, которые используются для опорного в внешнем предсказании для текущего изображения, а также 2) одного или более долгосрочных опорных изображений, которые не используются для опорного в внешнем предсказании для текущего изображения.

Таким образом вышеуказанные отдельные способы или любая комбинация таковых, включая любую комбинацию альтернативных примеров, могут обеспечить способы, относящиеся к следующему. Однако, ниже для простоты понимания приведен перечень и не должен рассматриваться ограничивающими. Один или несколько вышеуказанных способов могут быть реализованы вместе или отдельно. Кроме того, вышеуказанные способы являются примерами и не должны рассматриваться ограничивающими этими конкретными примерными способами.

С ограничениями на temporal_id для опорных изображений установленными так, что способы управления DPB хорошо подходят для временной масштабируемости, издержки сигнализации могут быть снижены, и может обеспечиваться возможность простого

процесса извлечения битового потока для точно извлеченных поднаборов битового потока.

Поднаборы долговременных опорных изображений, сигнализированные в наборе параметров изображения, и индекс могут быть включены в заголовок слайса. Это может обеспечить эффективную сигнализацию долгосрочных изображений.

Разделение набора опорных изображений на различные поднаборы, включая разделение для текущего изображения или для последующих изображений в очередности декодирования, разделения для тех, которые имеют более ранние или более поздние очередности вывода, чем текущее изображение. Они могут обеспечить повышенную эффективность и сниженную сложность для инициализации опорного списка и модификации списка опорных изображений.

Дважды дифференциальное кодирование в сигнализации идентификации краткосрочного изображения может обеспечить повышенную эффективность. Расширенная и ограниченная идентификация долгосрочного изображения может обеспечить повышенную эффективность и гибкость. Упрощенная инициализация списка опорных изображений может устранить необходимость маркировки “нет опорного изображения” для незаполненных элементов в списке опорных изображений; однако, это может не требоваться во всех примерах.

Упрощенные процессы для вывода декодированного изображения, вставки в DPB и удаления из DPB. Счетчик очередности изображения (POC) может быть отрицательным. Это может давать возможность некоторых важных вариантов использования, которые не могли быть позволяться, если POC не может быть отрицательным. Сигнализация, является ли изображение опорным изображением, не требуемым в процессе декодирования, может не требоваться, хотя она все еще может сигнализироваться. Маркировки опорных изображений как “неиспользуемое для опорного” могут более не являться необходимыми.

В одном или нескольких примерах описанные функции могут быть реализованы в виде аппаратных средств, программного обеспечения, микропрограммного обеспечения, или любой комбинации такового. Если реализовано в программном обеспечении, функции могут в виде одной или нескольких инструкций или кода сохраняться на считываемом компьютером носителе или передаваться по таковому, и исполняться аппаратно-реализованным блоком обработки. Считываемые компьютером носители могут включать в себя считываемый компьютером носитель данных, который соответствует реальному носителю, такому как носители данных, или носитель передачи данных, включая любой носитель, который содействует переносу компьютерной программы из одного места в другое, например, согласно протоколу связи. Таким образом считываемые компьютером носители в целом могут соответствовать (1) реальному считываемому компьютером носителю, который является долговременным или (2) среде передачи, такой как сигнал или несущая. Носители данных могут быть любыми имеющимися в распоряжении носителями, к которым может осуществлять доступ один или большее число компьютеров или один или большее число процессоров, чтобы извлекать инструкции, код и/или структуры данных для реализации способов, описанных в этом раскрытии. Компьютерный программный продукт может включать в себя считываемый компьютером носитель.

В качестве примера, а не ограничения, такие считываемые компьютером носители могут содержать оперативное запоминающее устройство (ОЗУ, RAM), постоянное запоминающее устройство (ПЗУ, ROM), электрически-стираемое программируемое ПЗУ (EEPROM), ПЗУ на компакт-диске (CD-ROM) или другое ЗУ на оптическом диске,

накопитель на магнитных дисках, или другие магнитные запоминающие устройства, флэш-память, или любой другой носитель, который может использоваться для хранения требуемого программного кода в форме инструкций или структур данных и к которому может осуществлять доступ компьютер. Кроме того, любое соединение в сущности называется считываемыми компьютером носителем. Например, если инструкции передаются от веб-сайта, сервера или другого удаленного источника, с использованием коаксиального кабеля, оптического кабеля, витой пар, цифровой абонентской лини (DSL) или беспроводных технологий, таких как инфракрасное излучение, радиосвязь и СВЧ-волна, то коаксиальный кабель, оптический кабель, витая пара, DSL или беспроводные технологии, такие как инфракрасное излучение, радиосвязь и СВЧ-волна, включаются в определение носителя. Следует понимать, однако, что считываемые компьютером носители и носители данных не включают в себя соединения, несущие, сигналы или другие кратковременные носители, но вместо этого ориентированы на долговременные, реальные носители. Магнитный диск и немагнитный диск, как используется в документе, включают в себя компакт-диск (CD), лазерный диск, оптический диск, цифровой универсальный диск (DVD), гибкий диск и диск по технологии Blu-ray, где магнитные диски обычно воспроизводят данные на основе магнитных свойств, тогда как немагнитные диски воспроизводят данные оптически с помощью лазеров. Комбинации вышеуказанного также должны включаться в рамки считываемых компьютером носителей.

Инструкции могут выполняться одним или несколькими процессорами, такими как один или большее число цифровых процессоров сигналов (DSP), универсальных микропроцессоров, специализированных интегральных схем (ASIC), программируемых вентильных матриц (FPGA), или другой эквивалентной интегральной или дискретной логической схемой. Соответственно, термин "процессор", как используется в документе, может ссылаться на любую вышеизложенную структуру или любую другую структуру, подходящую для реализации способов, описанных в документе. Кроме того, в некоторых аспектах функциональность, описанная в документе, может обеспечиваться в рамках специализированных аппаратных средств и/или программно реализованных модулей, сконфигурированных для кодирования и декодирования, или встроенных в комбинированный кодек. Кроме того, способы могут полностью реализовываться в одной или нескольких схемах или логических элементах.

Способы по данному раскрытию могут быть реализованы в большом разнообразии устройств или аппаратур, включая беспроводной телефон, интегральную схему (IC) или набор IC (например, микропроцессорный набор). Различные компоненты, модули, или блоки описываются в этом раскрытии, чтобы подчеркнуть функциональные аспекты устройств, сконфигурированных для выполнения раскрытых способов, но не обязательно требуют реализации посредством различных аппаратных блоков. Предпочтительнее, как описано выше, различные блоки могут быть объединены в аппаратном блоке кодека или обеспечиваться набором взаимодействующих аппаратных модулей, включая один или большее число процессоров, как описано выше, совместно с подходящим программным обеспечением и/или микропрограммным обеспечением.

Были описаны различные примеры. Эти и другие примеры находятся в рамках объема нижеследующей формулы изобретения.

Формула изобретения

1. Способ кодирования видеоданных, содержащий:

кодирование информации, указывающей опорные изображения, которые

принадлежат набору опорных изображений, при этом набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования;

построение множества поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений набора опорных изображений;

добавление опорных изображений из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений;

определение, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений;

если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, многократное повторное добавление одного или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, до тех пор, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений; и

кодирование текущего изображения на основании списка опорных изображений.

2. Способ по п. 1, в котором построение множества поднаборов опорных изображений содержит построение по меньшей мере первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений.

3. Способ по п. 1, в котором построение множества поднаборов опорных изображений содержит:

построение первого поднабора опорных изображений, который идентифицирует краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и до текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования;

построение второго поднабора опорных изображений, который идентифицирует краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и после текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования; и

построение третьего поднабора опорных изображений, который идентифицирует долгосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования.

4. Способ по п. 3,

в котором добавление опорных изображений из множества поднаборов опорных изображений содержит добавление опорных изображений из первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений в первый набор элементов в списке опорных изображений, и

при этом определение, является ли число элементов в списке опорных изображений

равным максимальному числу допустимых элементов в списке опорных изображений, содержит определение, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений после добавления опорных изображений из первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений в первый набор элементов в списке опорных изображений.

5. Способ по п. 3,

в котором многократное повторное добавление одного или более опорных изображений содержит идентификацию по меньшей мере одного опорного изображения из первого поднабора опорных изображений в более чем одном элементе в списке опорных изображений.

6. Способ по п. 1, в котором многократное повторное добавление одного или более опорных изображений содержит добавление опорных изображений в элементы списка опорных изображений так, что каждый элемент списка опорных изображений идентифицирует одно из опорных изображений, и так, что по меньшей мере два элемента списка опорных изображений идентифицируют одно и то же опорное изображение из опорных изображений.

7. Способ по п. 1,

в котором кодирование информации содержит декодирование с помощью видеодекодера информации, указывающей опорные изображения, которые принадлежат набору опорных изображений, при этом набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования;

при этом построение содержит построение с помощью видеодекодера множества поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений набора опорных изображений;

при этом добавление содержит добавление с помощью видеодекодера опорных изображений из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений;

при этом определение содержит определение с помощью видеодекодера, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений;

если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, то многократное повторное добавление содержит многократное повторное добавление с помощью видеодекодера одного или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, до тех пор, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений; и

при этом кодирование содержит декодирование с помощью видеодекодера текущего изображения на основании списка опорных изображений.

8. Способ по п. 1,

в котором кодирование информации содержит кодирование с помощью видеокодера информации, указывающей опорные изображения, которые принадлежат набору опорных изображений, при этом набор опорных изображений идентифицирует опорные

изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования;

при этом построение содержит построение с помощью видеокодера множества поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений набора опорных изображений;

при этом добавление содержит добавление с помощью видеокодера опорных изображений из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений;

при этом определение содержит определение с помощью видеокодера, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений;

если число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений, то многократное повторное добавление содержит многократное повторное добавление с помощью видеокодера одного или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, до тех пор, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений; и

при этом кодирование содержит кодирование с помощью видеокодера текущего изображения на основании списка опорных изображений.

9. Устройство для кодирования видеоданных, устройство содержит кодер видео, сконфигурированный с возможностью:

кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений, при этом набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования;

строить множество поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений набора опорных изображений;

добавлять опорные изображения из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений;

определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений;

если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, то многократно повторно добавлять одно или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, до тех пор, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений; и

кодировать текущее изображение на основании списка опорных изображений.

10. Устройство по п. 9, в котором для построения множества поднаборов опорных изображений, кодер видео сконфигурирован для построения по меньшей мере первого поднабора опорных изображений, второго поднабора опорных изображений и третьего

поднабора опорных изображений.

11. Устройство по п. 9, в котором для построения множества поднаборов опорных изображений, кодер видео сконфигурирован с возможностью:

строить первый поднабор опорных изображений, который идентифицирует краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и до текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования;

строить второй поднабор опорных изображений, который идентифицирует краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и после текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования; и

строить третий поднабор опорных изображений, который идентифицирует долгосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования.

12. Устройство по п. 11,

в котором для добавления опорных изображений из множества поднаборов опорных изображений, кодер видео сконфигурирован с возможностью добавлять опорные изображения из первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений в первый набор элементов в списке опорных изображений, и

при этом для определения, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, кодер видео сконфигурирован для определения, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений после добавления опорных изображений из первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений в первый набор элементов в списке опорных изображений.

13. Устройство по п. 11,

в котором для многократного повторного добавления одного или более опорных изображений кодер видео сконфигурирован для идентификации по меньшей мере одного опорного изображения из первого поднабора опорных изображений в более чем одном элементе в списке опорных изображений.

14. Устройство по п. 9, в котором для многократного повторного добавления одного или более опорных изображений кодер видео сконфигурирован с возможностью добавлять опорные изображения в элементы списка опорных изображений так, что каждый элемент списка опорных изображений идентифицирует одно из опорных изображений, и так, что по меньшей мере два элемента списка опорных изображений идентифицируют одно и то же опорное изображение из опорных изображений.

15. Устройство по п. 9, в котором кодер видео содержит видеодекодер, при этом видеодекодер сконфигурирован с возможностью:

декодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений, при этом набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования;

строить множество поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений набора опорных изображений;

добавлять опорные изображения из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений;

определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений;

если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, то многократно повторно добавлять одно или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, до тех пор, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений; и

декодировать текущее изображение на основании списка опорных изображений.

16. Устройство по п. 9, в котором кодер видео содержит видеокодер, и при этом видеокодер сконфигурирован, чтобы:

закодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений, при этом набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования;

строить множество поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений набора опорных изображений;

добавлять опорные изображения из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений;

определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений;

если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, многократно повторно добавлять одно или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, до тех пор, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений; и

кодировать текущее изображение на основании списка опорных изображений.

17. Устройство по п. 9, в котором устройство содержит одно из:

устройства беспроводной связи;

микропроцессора; и

интегральной схемы.

18. Считываемый компьютером носитель с наличием хранимых на нем инструкций, которые при исполнении предписывают процессору устройства для кодирования

видеоданных:

кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений, при этом набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования;

строить множество поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений набора опорных изображений;

добавлять опорные изображения из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений;

определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений;

если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, то многократно повторно добавлять одно или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, пока число элементов в списке опорных изображений не будет равным максимальному числу допустимых элементов в списке опорных изображений; и

кодировать текущее изображение на основании списка опорных изображений.

19. Считываемый компьютером носитель по п. 18, в котором инструкции, которые предписывают процессору строить множество поднаборов опорных изображений, содержат инструкции, которые предписывают процессору строить по меньшей мере первый поднабор опорных изображений, второй поднабор опорных изображений и третий поднабор опорных изображений.

20. Считываемый компьютером носитель по п. 18, в котором инструкции, которые предписывают процессору строить множество поднаборов опорных изображений, содержат инструкции, которые предписывают процессору:

строить первый поднабор опорных изображений, который идентифицирует краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и до текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования;

строить второй поднабор опорных изображений, который идентифицирует краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и после текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования; и

строить третий поднабор опорных изображений, который идентифицирует долгосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования.

21. Считываемый компьютером носитель по п. 20,

в котором инструкции, которые предписывают процессору добавлять опорные изображения из множества поднаборов опорных изображений, содержат инструкции, которые предписывают процессору добавлять опорные изображения из первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений в первый набор элементов в списке опорных изображений, и

при этом инструкции, которые предписывают процессору определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, содержат инструкции, которые предписывают процессору определять, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, после добавления опорных изображений из первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений в первый набор элементов в списке опорных изображений.

22. Считываемый компьютером носитель по п. 20,

в котором инструкции, которые предписывают процессору многократно повторно добавлять одно или более опорных изображений, содержат инструкции, которые предписывают процессору идентифицировать по меньшей мере одно опорное изображение из первого поднабора опорных изображений в более чем одном элементе в списке опорных изображений.

23. Считываемый компьютером носитель по п. 18, в котором инструкции, которые предписывают процессору многократно повторно добавлять одно или более опорных изображений, содержат инструкции, которые предписывают процессору добавлять опорные изображения в элементы списка опорных изображений с тем, что каждый элемент списка опорных изображений идентифицирует одно из опорных изображений, и так, что по меньшей мере два элемента списка опорных изображений идентифицируют одно и то же опорное изображение из опорных изображений.

24. Устройство для кодирования видеоданных, устройство содержит:

средство для кодирования информации, указывающей опорные изображения, которые принадлежат набору опорных изображений, при этом набор опорных изображений идентифицирует опорные изображения, которые потенциально могут быть использованы для внешнего предсказания текущего изображения и потенциально могут быть использованы для внешнего предсказания одного или более изображений, следующих после текущего изображения в очередности декодирования;

средство для построения множества поднаборов опорных изображений, так что каждое идентифицирует нуль или более опорных изображений набора опорных изображений;

средство для добавления опорных изображений из множества поднаборов опорных изображений в первый набор элементов в списке опорных изображений;

средство для определения, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений;

если число элементов в списке опорных изображений не является равным максимальному числу допустимых элементов в списке опорных изображений, средство для многократного повторного добавления одного или более опорных изображений из по меньшей мере одного из поднаборов опорных изображений в элементы в списке опорных изображений, которые находятся после первого набора элементов, пока число элементов в списке опорных изображений не будет равным максимальному числу

допустимых элементов в списке опорных изображений; и

средство для кодирования текущего изображения на основании списка опорных изображений.

25. Устройство по п. 24, в котором средство для построения множества поднаборов опорных изображений содержит средство для построения по меньшей мере первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений.

26. Устройство по п. 24, в котором средство для построения множества поднаборов опорных изображений содержит:

средство для построения первого поднабора опорных изображений, который идентифицирует краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и до текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования;

средство для построения второго поднабора опорных изображений, который идентифицирует краткосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и после текущего изображения в очередности вывода и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования; и

средство для построения третьего поднабора опорных изображений, который идентифицирует долгосрочные опорные изображения, которые находятся до текущего изображения в очередности декодирования и которые потенциально могут быть использованы для внешнего предсказания текущего изображения и одного или более из упомянутого одного или более изображений, следующих после текущего изображения в очередности декодирования.

27. Устройство по п. 26,

в котором средство для добавления опорных изображений из множества поднаборов опорных изображений содержит средство для добавления опорных изображений из первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений в первый набор элементов в списке опорных изображений, и

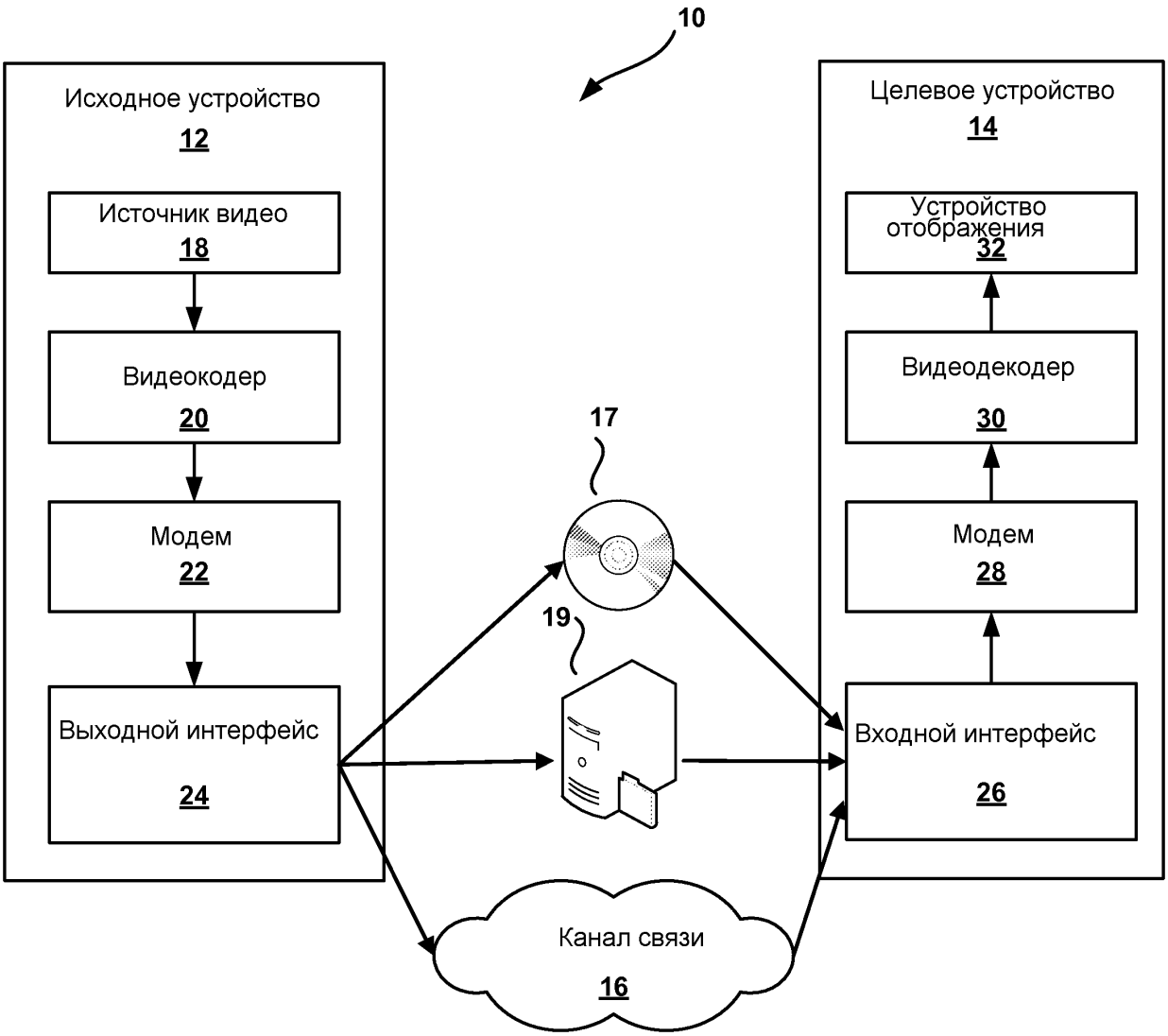
при этом средство для определения, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, содержит средство для определения, является ли число элементов в списке опорных изображений равным максимальному числу допустимых элементов в списке опорных изображений, после добавления опорных изображений из первого поднабора опорных изображений, второго поднабора опорных изображений и третьего поднабора опорных изображений в первый набор элементов в списке опорных изображений.

28. Устройство по п. 26,

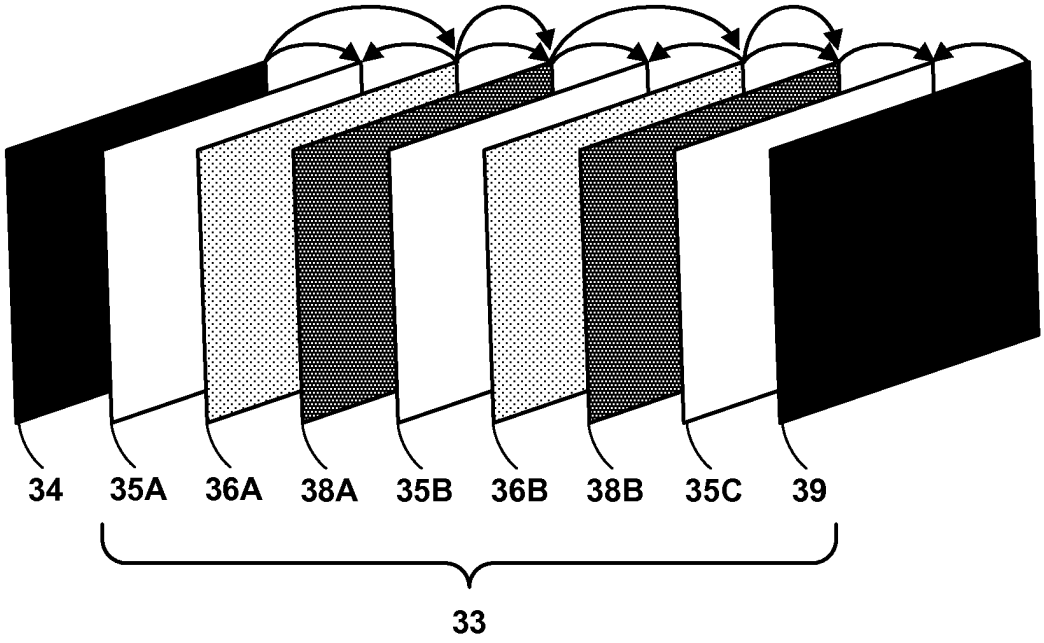
в котором средство для многократного повторного добавления одного или более опорных изображений содержит средство для идентификации по меньшей мере одного опорного изображения из первого поднабора опорных изображений в более чем одном элементе в списке опорных изображений.

29. Устройство по п. 24, в котором средство для многократного повторного добавления одного или более опорных изображений содержит средство для добавления

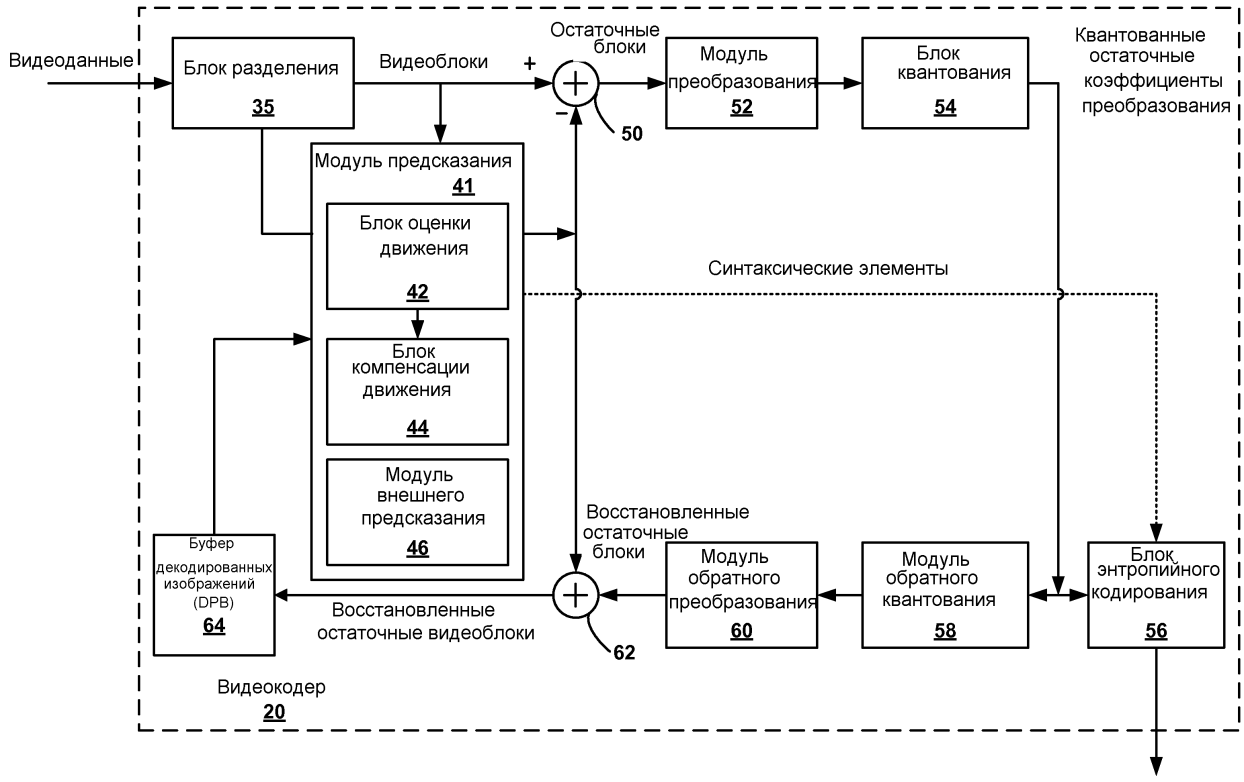
опорных изображений в элементы списка опорных изображений так, что каждый элемент списка опорных изображений идентифицирует одно из опорных изображений, и так, что по меньшей мере два элемента списка опорных изображений идентифицируют одно и то же опорное изображение из опорных изображений.


10
Исходное устройство
12
Источник видео
18
Видеокодер
20
Модем
22
Выходной интерфейс
24
17
19
Канал связи
16
Целевое устройство
14
Устройство отображения
32
Видеодекодер
30
Модем
28
Входной интерфейс
26

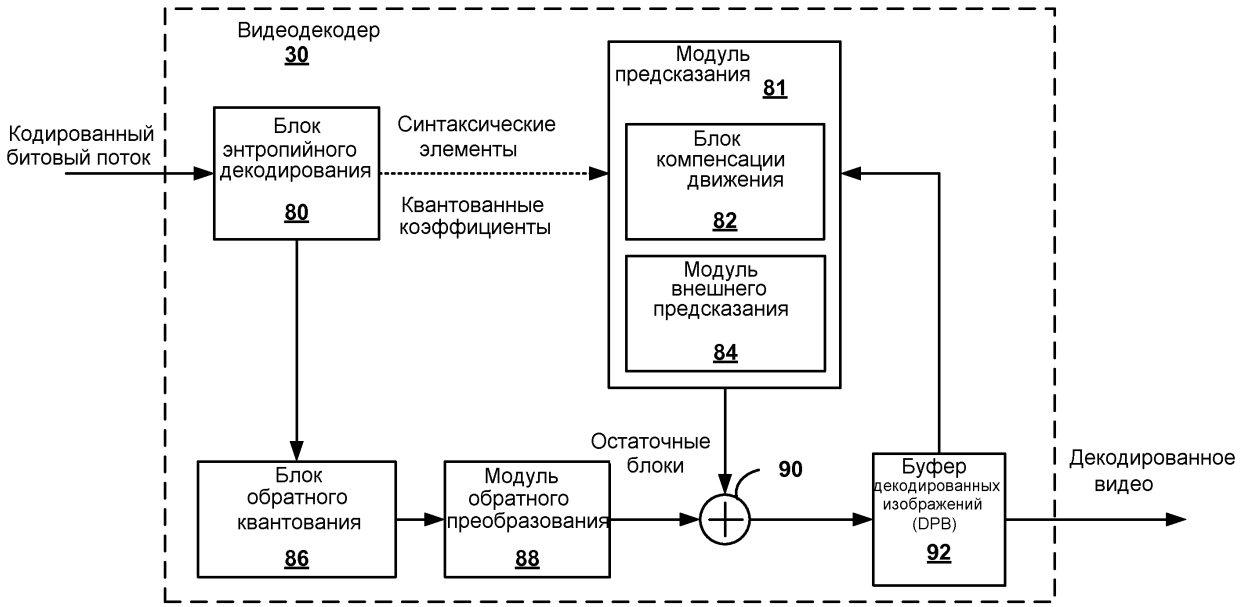
ФИГ.1


34
35A
36A
38A
35B
36B
38B
35C
39
33

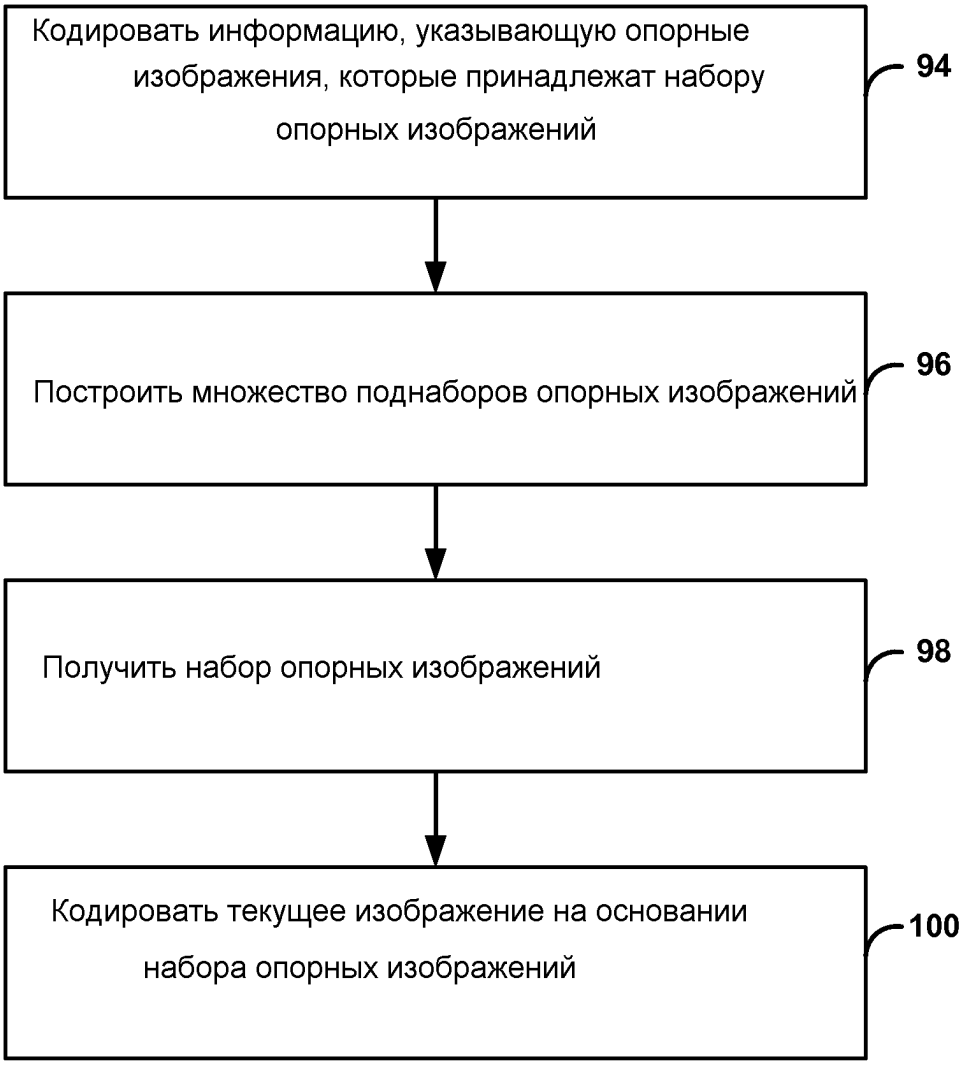
ФИГ.2


Видеоданные
Блок разделения
35
Видеоблоки
+
−
Остаточные блоки
50
Модуль преобразования
52
Блок квантования
54
Квантованные остаточные коэффициенты преобразования
Модуль предсказания
41
Блок оценки движения
42
Блок компенсации движения
44
Модуль внешнего предсказания
46
Синтаксические элементы
Буфер декодированных изображений (DPB)
64
Восстановленные остаточные видеоблоки
Восстановленные остаточные блоки
62
Модуль обратного преобразования
60
Модуль обратного квантования
58
Блок энтропийного кодирования
56
Видеокодер
20

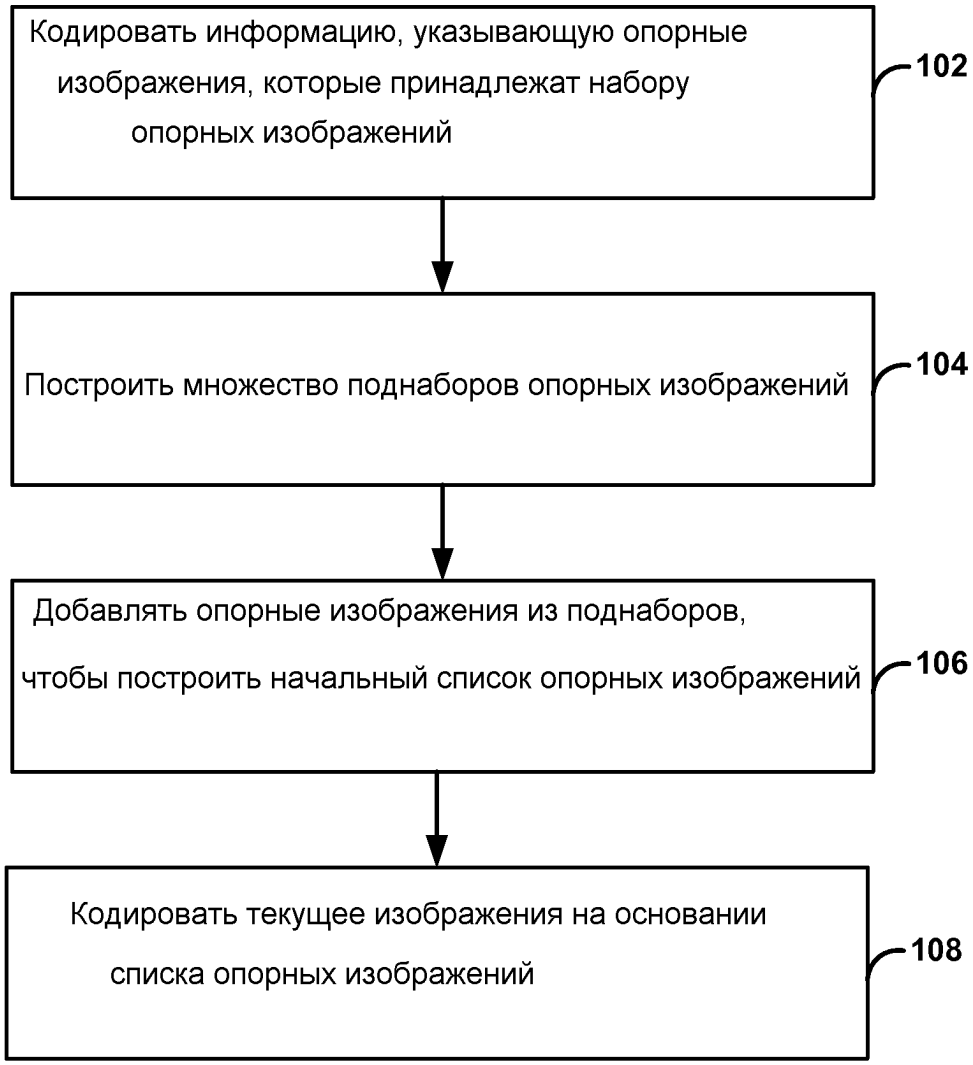
ФИГ.3


Видеодекодер
30
Кодированный битовый поток
Блок энтропийного декодирования
80
Синтаксические элементы
Квантованные коэффициенты
Модуль предсказания
81
Блок компенсации движения
82
Модуль внешнего предсказания
84
Остаточные блоки
90
Блок обратного квантования
86
Модуль обратного преобразования
88
Буфер декодированных изображений (DPB)
92
Декодированное видео

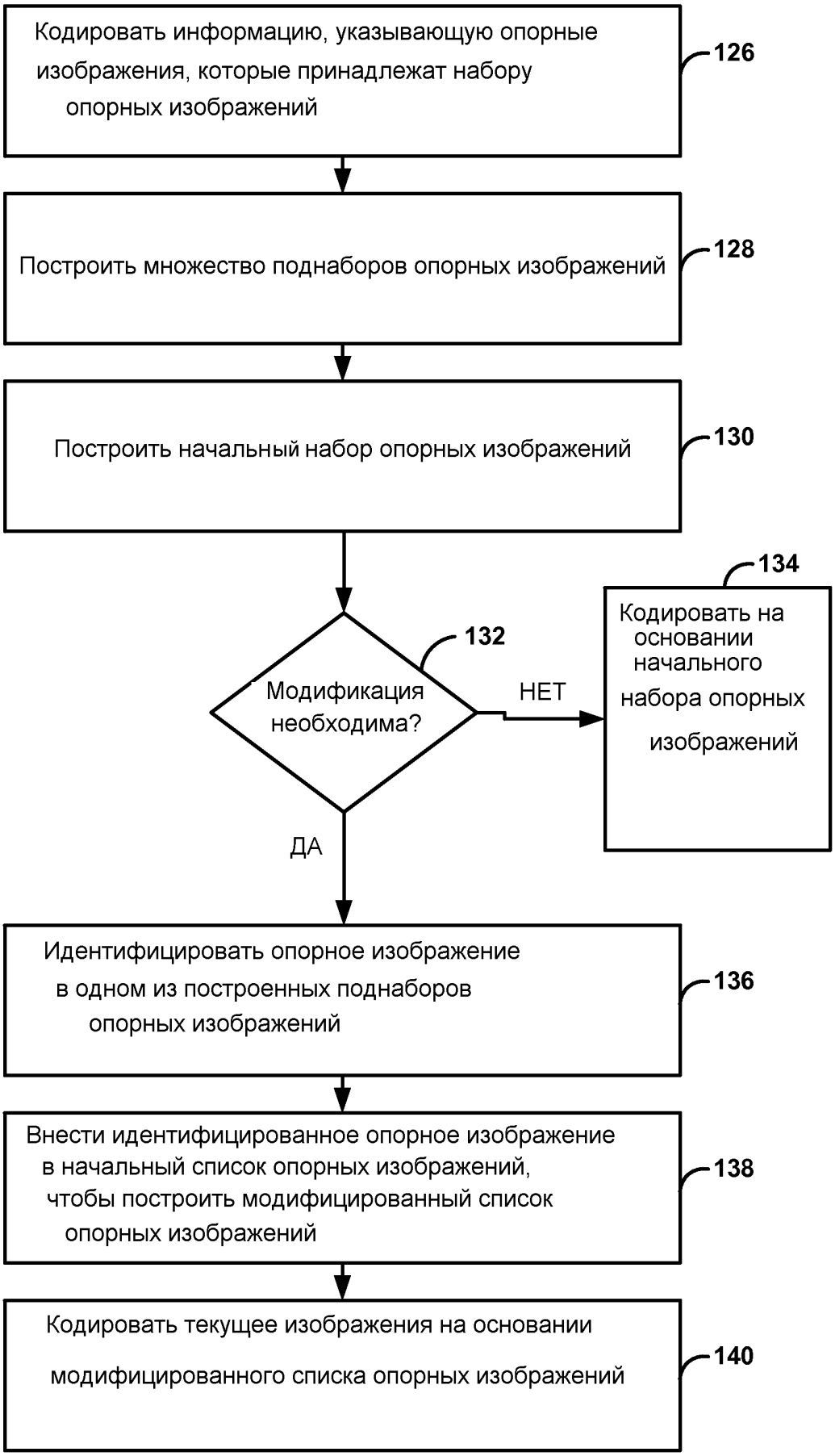
ФИГ.4


Кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений
94
Построить множество поднаборов опорных изображений
96
Получить набор опорных изображений
98
Кодировать текущее изображение на основании набора опорных изображений
100

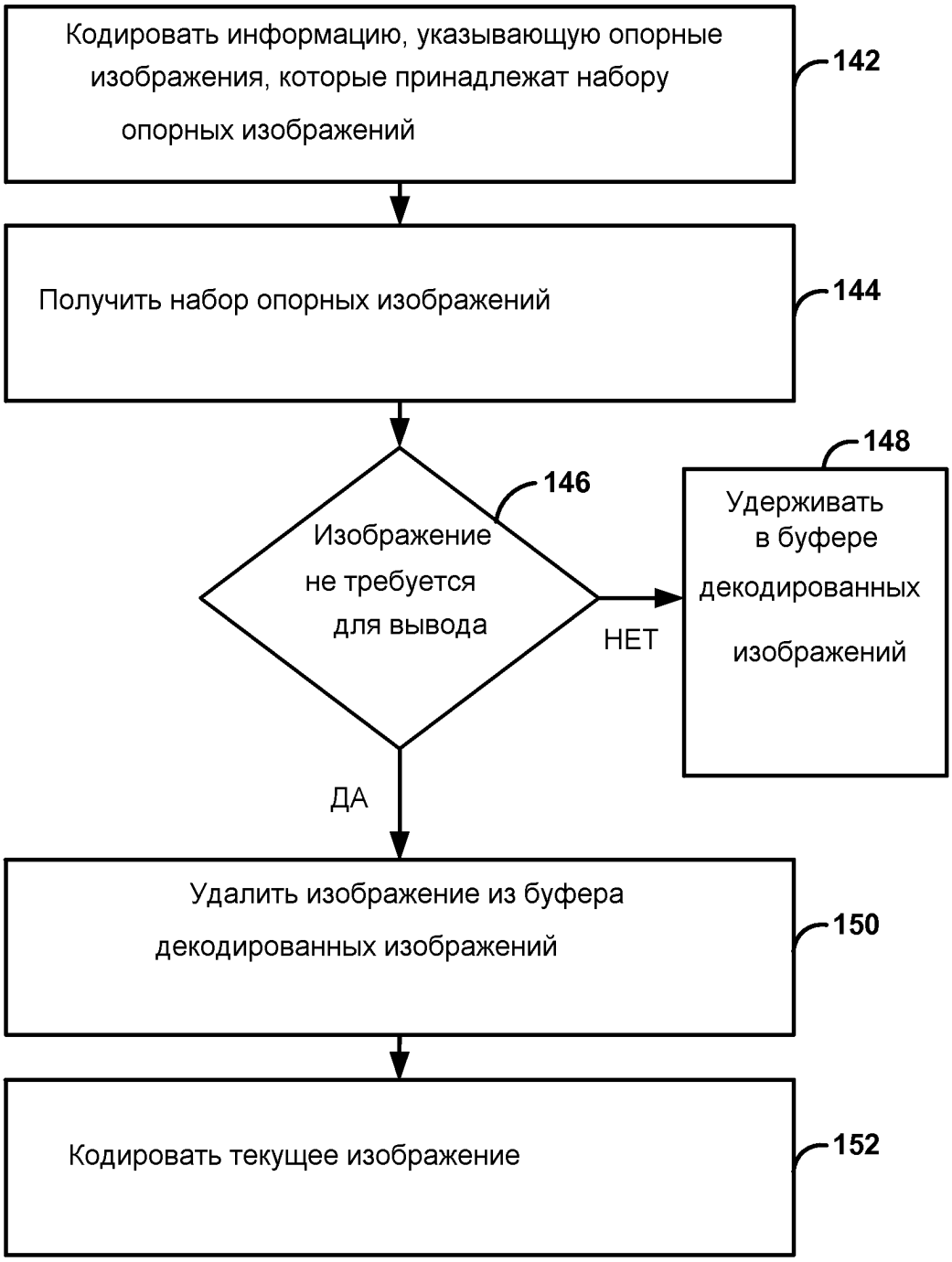
ФИГ.5


Кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений
102
Построить множество поднаборов опорных изображений
104
Добавлять опорные изображения из поднаборов, чтобы построить начальный список опорных изображений
106
Кодировать текущее изображения на основании списка опорных изображений
108

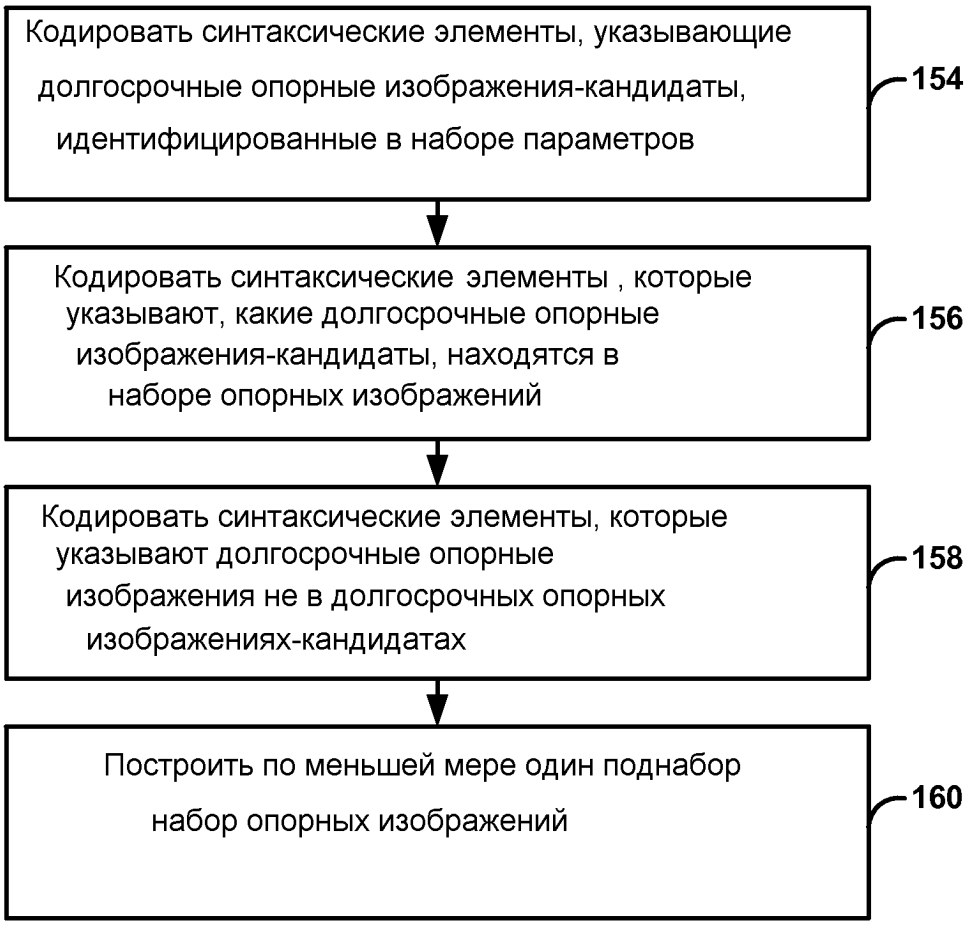
ФИГ.6


Кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений
126
Построить множество поднаборов опорных изображений
128
Построить начальный набор опорных изображений
130
134
Кодировать на основании начального набора опорных изображений
132
Модификация необходима?
НЕТ
ДА
Идентифицировать опорное изображение в одном из построенных поднаборов опорных изображений
136
Внести идентифицированное опорное изображение в начальный список опорных изображений, чтобы построить модифицированный список опорных изображений
138
Кодировать текущее изображения на основании модифицированного списка опорных изображений
140

ФИГ.8


Кодировать информацию, указывающую опорные изображения, которые принадлежат набору опорных изображений
142
Получить набор опорных изображений
144
146
Изображение не требуется для вывода
148
Удерживать в буфере декодированных изображений
НЕТ
ДА
Удалить изображение из буфера декодированных изображений
150
Кодировать текущее изображение
152

ФИГ.9


Кодировать синтаксические элементы, указывающие долгосрочные опорные изображения-кандидаты, идентифицированные в наборе параметров
154
Кодировать синтаксические элементы , которые указывают, какие долгосрочные опорные изображения-кандидаты, находятся в наборе опорных изображений
156
Кодировать синтаксические элементы, которые указывают долгосрочные опорные изображения не в долгосрочных опорных изображениях-кандидатах
158
Построить по меньшей мере один поднабор набор опорных изображений
160

ФИГ.10