

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2014142546, 13.03.2013

(24) Дата начала отсчета срока действия патента:
13.03.2013Дата регистрации:
29.08.2017

Приоритет(ы):

(30) Конвенционный приоритет:
22.03.2012 US 61/614,178;
04.04.2012 US 61/620,273;
29.06.2012 US 61/666,316;
12.03.2013 US 13/796,572

(43) Дата публикации заявки: 20.05.2016 Бюл. № 14

(45) Опубликовано: 29.08.2017 Бюл. № 25

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 22.10.2014(86) Заявка РСТ:
US 2013/030886 (13.03.2013)(87) Публикация заявки РСТ:
WO 2013/142195 (26.09.2013)Адрес для переписки:
129090, Москва, ул. Б. Спасская, 25, строение 3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

ГО Ливэй (US),
КАРЧЕВИЧ Марта (US),
ЧИЭНЬ Вэй-Цзюн (US)

(73) Патентообладатель(и):

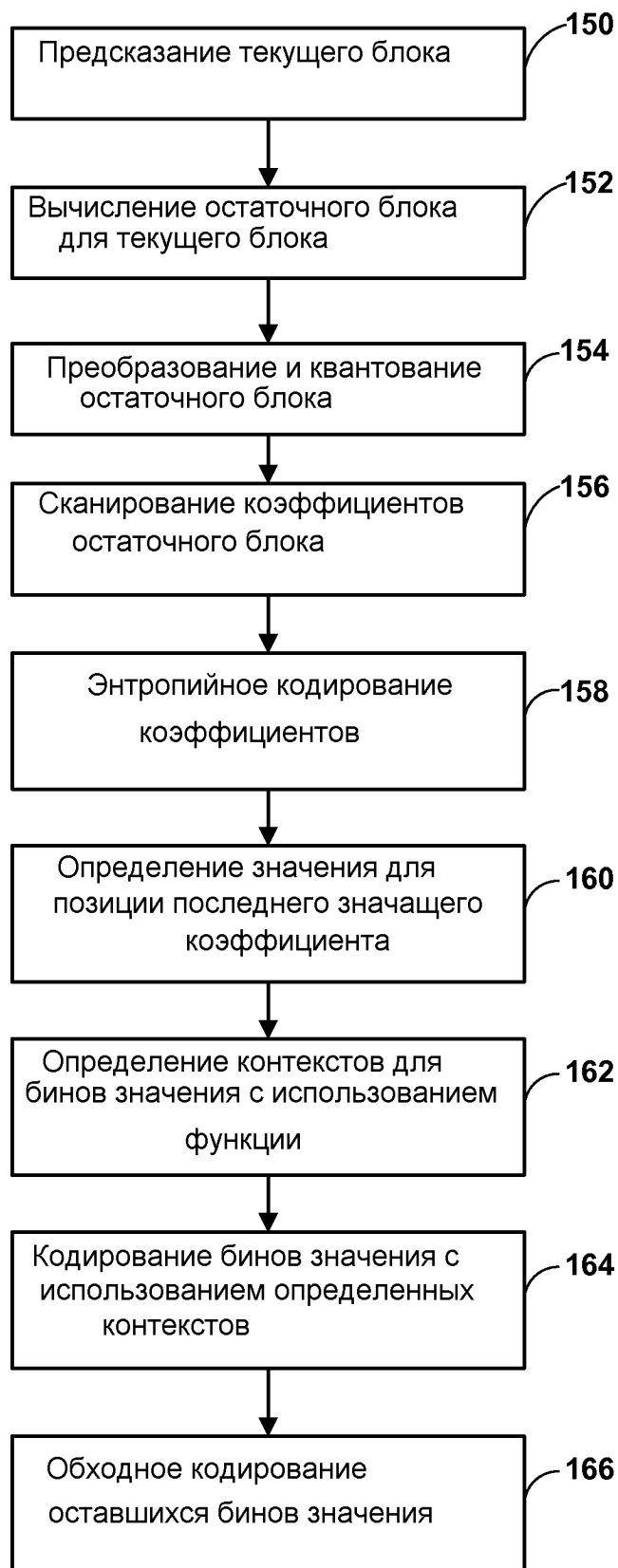
КВЭЛКОММ ИНКОРПОРЕЙТЕД (US)

(56) Список документов, цитированных в отчете
о поиске: US 7535387 B1, 19.05.2009. US
7379608 B2, 27.05.2008. US 2012/0014454 A1,
19.01.2012. WO 2010/067973 A2, 17.06.2010. RU
2336661 C2, 20.10.2008.(54) ПОЛУЧЕНИЕ КОНТЕКСТА ДЛЯ КОДИРОВАНИЯ ПОСЛЕДНЕЙ ПОЗИЦИИ ПРИ
ВЫПОЛНЕНИИ ВИДЕОКОДИРОВАНИЯ

(57) Реферат:

Изобретение относится к вычислительной технике. Технический результат заключается в повышении эффективности использования памяти видеокодера/декодера. Способ кодирования видеоданных содержит определение контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса упомянутого бина, в котором функция создает контекстный индекс для контекста путем сдвига вправо индекса бина на величину k и

добавления сдвинутого вправо значения к значению смещения, причем значение смещения определяют согласно формуле: $\text{offset} = 3 * n + ((n + 1) \gg 2)$, причем значение k определяют согласно формуле: $k = (n + 3) \gg 2$, и причем значение n определяют согласно формуле: $n = (\log_2(\text{block_size}) - 2)$, и причем значение block_size содержит значение, указывающее размер блока, и кодирование упомянутого бина с использованием определенного контекста. 4 н. и 17 з.п. ф-лы, 5 ил., 10 табл.



Фиг. 4



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**(21)(22) Application: **2014142546**, 13.03.2013(24) Effective date for property rights:
13.03.2013Registration date:
29.08.2017

Priority:

(30) Convention priority:
22.03.2012 US 61/614,178;
04.04.2012 US 61/620,273;
29.06.2012 US 61/666,316;
12.03.2013 US 13/796,572(43) Application published: **20.05.2016** Bull. № 14(45) Date of publication: **29.08.2017** Bull. № 25(85) Commencement of national phase: **22.10.2014**(86) PCT application:
US 2013/030886 (13.03.2013)(87) PCT publication:
WO 2013/142195 (26.09.2013)Mail address:
129090, Moskva, ul. B. Spasskaya, 25, stroenie 3,
OOO "Yuridicheskaya firma Gorodisskij i Partnery"

(72) Inventor(s):

GO Livej (US),
KARCHEVICH Marta (US),
CHIEN Vej-Tszyun (US)

(73) Proprietor(s):

KVELKOMM INKORPOREJTED (US)(54) **OBTAINING THE CONTEXT TO ENCODE THE LAST POSITION WHEN PERFORMING THE ENCODING**

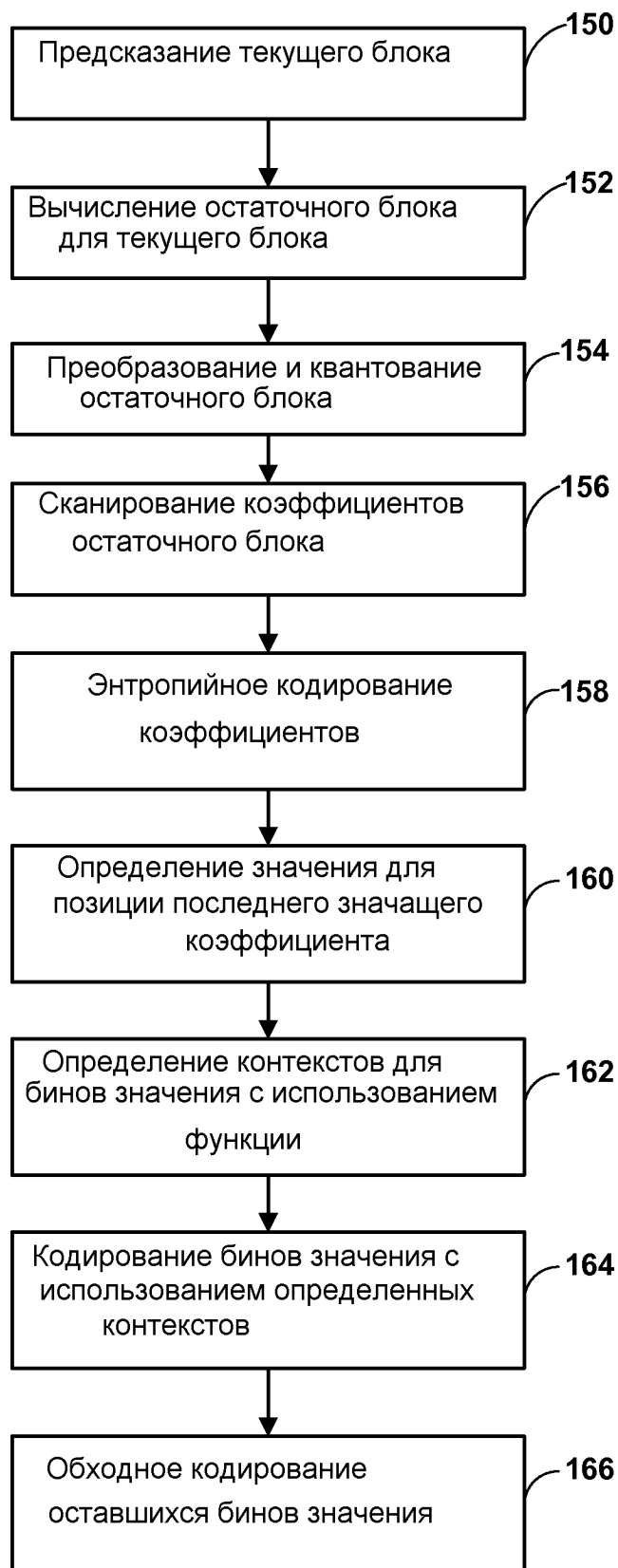
(57) Abstract:

FIELD: physics.

SUBSTANCE: way of encoding video data contains the context definition for entropy coding bin value that indicates the last significant block coefficient, using index referred to bin, which feature creates a context index on the context by bin index right shift on the value of k and adding the right shifted values to the value of the offset. Define the offset value according to the formula: $\text{offset} = 3 * n + ((n + 1) \gg 2)$. The value of k isdetermined according to the formula: $k = (n + 3) \gg 2$. And where the value of n is determined according to the formula: $n = (\log_2(\text{block_size}) - 2)$. And where the value of block_size contains a value indicating the size of the block, and encoding the named bin using a specific context.

EFFECT: increasing the efficiency of using the memory of a video encoder, video decoder.

21 cl, 5 dwg, 10 tbl



Фиг. 4

Данная заявка испрашивает приоритет предварительной заявки США №61/614,178, поданной 22 марта 2012 года, предварительной заявки США №61/620,273, поданной 04 апреля 2012 года и предварительной заявки США №61/666,316, поданной 29 июня 2012 года, содержание каждой из которых включено сюда по ссылке.

5 Область техники

Настоящее изобретение относится к видеокодированию.

Уровень техники

Возможности цифрового видео могут быть предусмотрены в устройства самого широкого применения, в том числе цифровые телевизионные приемники, системы
10 прямого цифрового вещания, системы беспроводного вещания, персональные цифровые помощники (PDA), портативные компьютеры или настольные компьютеры, планшетные компьютеры, устройства считывания электронных книг, цифровые камеры, цифровые записывающие устройства, цифровые медиаплееры, видеоигровые устройства, видеоигровые консоли, сотовые или спутниковые радиотелефоны, так называемые
15 «смартфоны», устройства для проведения видеотелеконференций, устройства потокового видео и т.п. Устройства цифрового видео реализуют методики видеокодирования, такие как описаны в стандартах, определенных как MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Часть 10, стандарт усовершенствованного видеокодирования (AVC), стандарт высокоэффективного видеокодирования (HEVC), находящийся в процессе
20 разработки, а также расширения указанных стандартов. Видеоустройства могут передавать, принимать, кодировать, декодировать и/или запоминать цифровую видеoinформацию более эффективным образом путем реализации указанных методик видеокодирования.

Методики видеокодирования включают в себя пространственное (интракадровое)
25 предсказание и/или временное (интеркадровое) предсказание для уменьшения или устранения избыточности, присущей видеопоследовательностям. Для видеокодирования на основе блоков видеослайсы (например, видеокадр или часть видеокадра) могут быть разделены на видеоблоки, которые также могут называться древовидными блоками, единицами кодирования (CU), и/или узлами кодирования. Видеоблоки в
30 интракодированном (I) слайсе изображения кодируют с использованием пространственного предсказания применительно к опорным отсчетам в соседних блоках одного и того же изображения. В видеоблоках в интеркодированном (P или B) слайсе изображения может использоваться пространственное предсказание в отношении опорных отсчетов в соседних блоках одного и того же изображения или временное
35 предсказание в отношении опорных отсчетов других опорных изображений. Изображения могут называться кадрами, а опорные изображения могут называться опорными кадрами.

В результате пространственного или временного предсказания получают блок предсказания для блока, подлежащего кодированию. Остаточные данные представляют
40 пиксельные разности между исходным блоком, подлежащим кодированию, и блоком предсказания. Интеркодированный блок кодируют в соответствии с вектором движения, который указывает блок опорных отсчетов, образующих блок предсказания, и в соответствии с остаточными данными, показывающими разность между кодированным блоком и блоком предсказания. Интракодированный блок кодируют в соответствии
45 с режимом интракодирования и остаточными данными. Для дополнительного сжатия остаточные данные можно преобразовать из пиксельной области в область преобразования, в результате чего получают остаточные коэффициенты преобразования, которые затем могут быть подвергнуты квантованию. Квантованные коэффициенты

преобразования, первоначально скомпонованные в двумерной матрице, могут быть отсканированы для создания одномерного вектора коэффициентов преобразования, к которым можно применить энтропийное кодирование для достижения еще большего сжатия.

5 Сущность изобретения

В общем случае в этом изобретении описываются методики для кодирования синтаксических элементов, связанных с видеоданными, с использованием одной или нескольких функций. Например, устройство может реализовать одну или несколько методик для кодирования значения, указывающего позицию последнего значащего коэффициента блока видеоданных (такого как единица преобразования, или «TU»). Для кодирования этого значения устройство может использовать функцию индекса каждого бита (или «бина») в бинаризованном значении, соответствующем последнему значащему коэффициенту, где индекс указывает позицию бина в матрице бинов, представляющих упомянутое бинаризованное значение.

15 В одном примере способ включает в себя определение контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса этого бина, и кодирование указанного бина с использованием определенного контекста.

В другом примере устройство для кодирования видеоданных включает в себя 20 видеокодер, выполненный с возможностью определения контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса этого бина, и кодирование бина с использованием определенного контекста.

Еще в одном примере устройство включает в себя средство для определения контекста 25 для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса этого бина, и средство для кодирования бина с использованием определенного контекста.

В следующем примере считываемый компьютером носитель данных закодирован командами. При выполнении эти команды предписывают программируемому 30 процессору вычислительного устройства определять контекст для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса этого бина, и кодирования бина с использованием определенного контекста.

Далее на сопроводительных чертежах и приведенном ниже описании излагаются 35 детали одного или нескольких примеров. Другие признаки, задачи и преимущества изобретения станут очевидными из упомянутого описания, чертежей, а также из формулы изобретения.

Краткое описание чертежей

Фиг. 1 - блок-схема, иллюстрирующая приведенную в качестве примера систему 40 видеокодирования и декодирования, в которой могут использоваться методики для определения контекста с целью его использования для кодирования значения, представляющего последний значащий коэффициент блока видеоданных;

Фиг. 2 - блок-схема, иллюстрирующая приведенный в качестве примера видеокодер, 45 которой может реализовать методики для определения контекста с целью его использования для кодирования значения, представляющего последний значащий коэффициент блока видеоданных;

Фиг. 3 - блок-схема, иллюстрирующая приведенный в качестве примера декодер, которой может реализовать методики для определения контекста с целью его

использования для кодирования значения, представляющего последний значащий коэффициент блока видеоданных;

Фиг. 4 - блок-схема, иллюстрирующая приведенный в качестве примера способ кодирования текущего блока видеоданных;

Фиг. 5 - блок-схема, иллюстрирующая приведенный в качестве примера способ декодирования текущего блока видеоданных.

Подробное описание изобретения

В общем случае методики этого изобретения относятся к видеокодированию. При видеокодировании последовательность изображений кодируют индивидуально, используя либо пространственное предсказание (интра-предсказание), либо временное предсказание (интер-предсказание). В частности, видеокодеры кодируют отдельные блоки изображений, используя интра- или инте-предсказание. Видеокодеры также кодируют остаточные данные для блоков, где остаточные данные в общем случае соответствуют остаточным блокам, которые представляют разности по пикселям между предсказанными данными и не обработанными, не кодированными данными.

Видеокодеры могут выполнить преобразование и квантование остаточных данных для создания квантованных коэффициентов преобразования для остаточных блоков.

Видеокодеры дополнительно кодируют синтаксические данные такие как, являются ли эти коэффициенты значащими (то есть имеют абсолютные значения, превышающие 0), местоположения значащих коэффициентов, местоположение последнего значащего коэффициента в порядке сканирования и значения уровня для значащих коэффициентов.

В настоящем изобретении описаны методики для кодирования значения, указывающего последний значащий коэффициент в блоке видеоданных, таком как единица преобразования (TU). В частности, для кодирования синтаксических элементов, таких как значение, указывающее последний значащий коэффициент в блоке, видеокодеры могут быть выполнены с возможностью применения контекстно-адаптивного бинарного арифметического кодирования (CABAC). CABAC кодирование включает в себя использование различных контекстов, указанных контекстными индексами, которые обычно указывают вероятность того, что отдельный бит (или «бин») бинаризованной цепочки будет иметь конкретное значение (например 0 или 1). В частности контекст для кодирования бина значения, указывающего последний значащий коэффициент в блоке, определяют отдельно для каждого бина указанного значения, то есть на основе местоположения этого бина в данном значении (например, индекс бина в предположении, что упомянутое значение представлено в виде матрицы бинов).

Методики этого изобретения включают в себя использование некой функции для определения контекстного индекса контекста для использования с целью кодирования бина, а не использование таблицы отображения, которая обеспечивает указания контекстных индексов для контекстов с целью их использования для кодирования конкретных бинов. В частности, такой функцией может быть функция индекса бина. Например, если предположить, что упомянутый бин является i -м бином кодируемого значения, то может быть определена функция в виде $f(i)$, где $f(i)$ возвращает значение контекстного индекса, соответствующего контексту для использования с целью кодирования бина i бинаризованного значения. Как было описано выше, контекст может указывать вероятность того, что бин i будет иметь конкретное значение, например, 0 или 1.

Таким путем в настоящем изобретении описаны методики кодирования согласно CABAC позиции последнего значащего коэффициента (последняя позиция). Для бина

последней позиции, подлежащего кодированию, индекс контекста САВАС можно получить, используя функцию, такую как таблица отображения бинов последней позиции на контексты САВАС, на которой можно сэкономить (например, не хранить).

Кодирование САВАС обычно включает в себя две части: бинаризацию и кодирование САВАС. Процесс бинаризации выполняют для преобразования местоположения последнего значащего коэффициента блока в двоичную цепочку, например, матрицу бинов. Способом бинаризации, используемым в тестовой модели высокоэффективного видеокодирования (НМ), является усеченное унарное кодирование с фиксированной длиной. Для части усеченного унарного кода кодирование бинов осуществляется с использованием контекстов САВАС. Для части с фиксированной длиной кодирование бинов осуществляется с использованием обходного режима (без контекстов). Пример ТУ (единица преобразования/блок преобразования) размером 32×32 показан в Таблице 1.

Таблица 1			
Величина компоненты последней позиции	Усеченный унарный (контекстная модель)	Фиксированный бинарный (обходной)	f_value
0	1	-	0
1	01	-	0
2	001	-	0
3	0001	-	0
4-5	00001	X	0-1
6-7	000001	X	0-1
8-11	0000001	XX	0-3
12-15	00000001	XX	0-3
16-23	000000001	XXX	0-7
24-31	0000000001	XXX	0-7

В приведенной ниже Таблице 2 показан пример таблицы контекстного отображения, используемой в стандартной модели НМ. В Таблице 2 показано, что последние позиции в разных местах могут иметь одинаковые контексты. Для некоторых бинов, например, бины 6-7 блока 8×8 контекст не присвоен, поскольку они кодированы без контекста (обходной режим), как показано в приведенной выше Таблице 1.

Таблица 2										
Индекс бина	0	1	2	3	4	5	6	7	8	9
ТУ 4×4	0	1	2							
ТУ 8×8	3	4	5	5	2					
ТУ 16×6	6	7	8	8	9	9	2			
ТУ 32×32	10	11	12	14	13	13	14	14	2	

Хотя в стандартной модели НМ таблица, такая как таблица 2, используется для определения контекстов для кодирования бинов, значения последней позиции (то есть значения, указывающего позицию последнего значащего коэффициента в блоке видеоданных), раскрытые здесь методики включают в себя использование функции для определения контекстов для кодирования бинов значения последней позиции. Таким образом, в видеокодере, сконфигурированном согласно методикам, предложенным в настоящем изобретении, нет необходимости иметь таблицу, подобную Таблице 2. Для получения контекстного индекса САВАС для бинов при кодировании последней позиции можно использовать функцию, так что таблицу отображения (Таблица 2) можно исключить. Далее более подробно описываются различные примеры устройств кодирования, выполненных с возможностью реализации функции, предназначенной для определения контекстов для кодирования бинов синтаксических элементов.

На фиг. 1 представлена блок-схема, где показан пример системы 10 видеокодирования и видеodeкодирования, в которой могут быть использованы методики для определения контекста с целью его использования для кодирования значения, представляющего последний значащий коэффициент блока видеоданных. Как показано на фиг. 1 система 10 включает в себя устройство-источник 12, которое обеспечивает кодированные видеоданные, в последствии подлежащие декодированию устройством-адресатом 14. В частности, устройство-источник 12 подает видеоданные на устройство-адресат 14, используя считываемую компьютером среду 16. Устройство-источник 12 и устройство-адресат 14 могут содержать любое из самых разных устройств, в том числе, настольные компьютеры, компьютеры типа «ноутбук» (например, «лэптоп»), планшетные компьютеры, телевизионные приставки, телефонные трубки, такие как так называемые «смартфоны», так называемые «смартпэды», телевизионные приемники, камеры, устройства отображения, цифровые медиаплееры, видеоигровые консоли, устройство потокового видео или т.п. В некоторых случаях устройство-источник 12 и устройство-адресат 14 могут быть оборудованы средствами беспроводной связи.

Устройство-адресат 14 может принимать кодированные видеоданные, подлежащие декодированию, через считываемую компьютером среду 16. Считываемая компьютером среда 16 может содержать среду или устройство любого типа, способное перемещать кодированные видеоданные от устройства-источника 12 на устройство-адресат 14. В одном примере считываемая компьютером среда 16 может содержать среду связи, предоставляющую устройству-источнику 12 возможность передавать кодированные видеоданные непосредственно на устройство-адресат 14 в режиме реального времени. Кодированные видеоданные могут модулироваться согласно стандарту связи, такому как протокол беспроводной связи, и передаваться на устройство-адресат 14. Среда связи может содержать любой носитель проводной или беспроводной связи, например, радиочастотный (RF) спектр или одно или несколько физических линий передачи. Среда связи может составлять часть сети пакетной передачи, такой как локальная сеть, региональная сеть или глобальная сеть, например Интернет. Среда связи может включать в себя маршрутизаторы, коммутаторы, базовые станции или любое другое оборудование, которое может быть полезно для осуществления связи между устройством-источником 12 и устройством-адресатом 14.

В некоторых примерах кодированные данные могут выводиться через выходной интерфейс 22 в запоминающее устройство. Аналогичным образом, доступ к кодированным данным из запоминающего устройства может осуществляться через входной интерфейс. Это запоминающее устройство может включать в себя самые разные распределенные или локально доступные запоминающие среды, такие как накопитель на жестких дисках, диски типа Blu-ray, диски DVD, диски CD-ROM, флэш-память, энергозависимую или энергонезависимую память, либо любую другую подходящую цифровую запоминающую среду для хранения кодированных видеоданных. В еще одном примере запоминающее устройство может соответствовать файловому серверу или другому промежуточному запоминающему устройству, которое может хранить кодированное видео, созданное устройством-источником 12. Устройство-адресат 14 в этом примере может обращаться к кодированным видеоданным из запоминающего устройства посредством потоковой передачи или загрузки. Файловый сервер может представлять собой сервер любого типа, способный запоминать и хранить кодированные видеоданные и передавать эти кодированные видеоданные на устройство-адресат 14. Примеры файловых серверов включают в себя Web-сервер (например, для Web-сайта), сервер Протокола пересылки файлов (FTP), подключенные к сети

запоминающие устройства (NAS) и локальные накопители на диске. Устройство-адресат 14 может иметь доступ к кодированным видеоданным через любое стандартное соединение для передачи данных, в том числе Интернет соединение. Такие соединения могут включать в себя беспроводные каналы (например, Wi-Fi соединения), проводные соединения (например, DSL, кабельный модем и т.д.), либо комбинацию обоих видов соединений, которая подходит для доступа к кодированным видеоданным, хранящимся на файловом сервере. Передача кодированных видеоданных от запоминающего устройства может быть реализована в виде потоковой передачи, передачи с загрузкой или их комбинации.

Раскрытые здесь методики не обязательно ограничиваются беспроводными приложениями или установками. Эти методики можно применить к видеокодированию для поддержки любого из разнообразных мультимедийных приложений, таких как эфирное телевизионное вещание, передачи кабельного телевидения, передачи спутникового телевидения, передачи потокового видео, например, через Интернет, такие как динамическая адаптивная потоковая передача по Протоколу HTTP (DASH), цифровое видео, кодированное в запоминающей среде, декодирование цифрового видео, хранящегося в запоминающей среде, или другие приложения. В некоторых примерах система 10 может быть выполнена с возможностью поддержки однонаправленной или двунаправленной передачи видео для поддержки приложений, таких как потоковое видео, видео воспроизведение, видеовещание и/или видеотелефония.

В примере по фиг. 1 устройство-источник 12 включает в себя источник 18 видео, видеокодер 20 и выходной интерфейс 22. Устройство-адресат 14 включает в себя входной интерфейс 28, видеodeкодер 30 и устройство 32 отображения. Согласно настоящему изобретению видеокодер 20 устройства-источника 12 может быть выполнен с возможностью применения методик определения контекста с целью использования для кодирования значения, представляющего последний значащий коэффициент блока видеоданных. В других примерах устройство-источник и устройство-адресат могут включать в себя другие компоненты или компоновки. Например, устройство-источник 12 может принимать видеоданные от внешнего источника 18 видео, такого как внешняя камера. Аналогичным образом устройство-адресат 14 может взаимодействовать с внешним устройством отображения, а не содержать встроенное устройство отображения.

Показанная на фиг. 1 система 10 является просто одним из примеров. Методики определения контекста с целью его использования для кодирования значения, представляющего последний значащий коэффициент блока видеоданных, могут быть реализованы с помощью любого устройства цифрового видеокодирования и/или видеodeкодирования. Хотя в общем случае раскрытые здесь методики реализуются устройством видеокодирования, они также могут быть реализованы видеodeкодером/ видеodeкодером, называемым, как правило, «кодек». Кроме того, раскрытые здесь методики могут быть реализованы процессором предварительной обработки видео.

Устройство-источник 12 и устройство-адресат 14 являются лишь примерами указанных устройств кодирования, в которых устройство-источник 12 создает кодированные видеоданные для передачи на устройство-адресат 14. В некоторых примерах устройства 12, 14 могут функционировать по существу симметричным образом, так что каждое из устройств 12, 14 включает в себя компоненты видеокодирования и видеodeкодирования. Таким образом, система 10 может поддерживать однонаправленную или двунаправленную передачу видео между устройствами 12, 14, например, для потокового видео, воспроизведения видео, видеовещания или видеотелефонии.

Источник 18 видео устройства-источника 12 может включать в себя устройство

захвата изображения, такое как видеокамера, видеоархив, содержащий ранее зафиксированное видео, и/или интерфейс подачи видео для приема видео от провайдера видеоконтента. В качестве дополнительной альтернативы, источник 18 видео может создавать данные на основе компьютерной графики в качестве исходного видео или комбинации реального видео, архивированного видео и видео, созданного компьютером. В некоторых случаях, если источником 18 видео является видеокамера, устройство-источник 12 и устройство-адресат 14 могут образовывать так называемые «камерофоны» или «видеофоны». Однако, как упоминалось выше, раскрытые здесь методики можно применить к видеокодированию в целом, а также к приложениям для беспроводной и/или проводной связи. В каждом случае зафиксированное видео, предварительно зафиксированное видео или созданное компьютером видео может быть кодировано видеокодером 20. Затем кодированная видеoinформация может быть выведена выходным интерфейсом 22 в считываемую компьютером среду 16.

Считываемая компьютером среда 16 может включать в себя среду передачи, такую как беспроводное вещание или проводная сетевая передача, либо запоминающую среду (то есть некрatkвременная запоминающая среда), такую как жесткий диск, флэш-память, компакт-диск, цифровой видеодиск, диск Blue-ray или другую, считываемую компьютером среду. В некоторых примерах сетевой сервер (не показан) может принимать кодированные видеоданные от устройства-источника 12 и подавать эти кодированные видеоданные на устройство-адресат 14, например, посредством сетевой передачи. Аналогичным образом, вычислительное устройство средства создания медиа, такого как средство штамповки дисков, может принимать кодированные видеоданные от устройства-источника 12 и формировать диск, содержащий кодированные видеоданные. Таким образом, следует понимать, что в разных примерах считываемая компьютером среда 16 включает в себя одну или более считываемых компьютером сред различных видов.

Входной интерфейс 28 устройства-адресата 14 принимает информацию из считываемой компьютером среды 16. Информация в считываемой компьютером среде 16 может включать в себя синтаксическую информацию, определенную видеокодером 20, которая также используется видеодекодером 30, которая содержит синтаксические элементы, описывающие характеристики, и/или обработку блоков и других кодированных единиц (например, GOP). Устройство 32 отображения отображает декодированные видеоданные пользователю и может содержать любое из множества различных устройств отображения, таких как электронно-лучевая трубка (CRT), жидкокристаллический дисплей (LCD), плазменный дисплей, дисплей на органических светоизлучающих диодах (OLED), либо устройство отображения другого типа.

Видеокодер 20 и видеодекодер 30 могут работать в соответствии со стандартом видеокодирования, таким как стандарт высокоэффективного видеокодирования (HEVC), разрабатываемый в настоящее время, и может соответствовать тестовой модели (НМ) стандарта HEVC. В качестве альтернативы, видеокодер 20 и видеодекодер 30 могут работать согласно другим коммерческим или промышленным стандартам, таким как стандарт ITU-T H.264, имеющий другое название MPEG-4 часть 10, стандарт усовершенствованного видеокодирования (AVC) или расширения указанных стандартов. Однако раскрытые здесь методики не ограничиваются любым конкретным стандартом кодирования. Другие примеры стандартов видеокодирования включают в себя стандарты MPEG-2 и ITU-T H.263. Хотя это на фиг. 1 не показано, согласно некоторым аспектам видеокодер 20 и видеодекодер 30 могут каждый быть интегрированными с аудиокодером и аудиодекодером, причем они могут включать в себя соответствующие

блоки MUX-DEMUX или другие элементы аппаратного и программного обеспечения для обработки кодирования как аудио, так и видео в общем потоке данных или отдельных потоках данных. Если это применимо, то блоки MUX-DEMUX могут соответствовать протоколу мультиплексора ITU H.223 или другим протоколам, таким как протокол дейтаграмм пользователя (UDP).

Стандарт ITU-T H.264/MPEG-4 (AVC) был разработан экспертной группой по видеокодированию ITU-T вместе с экспертной группой по движущимся изображениям (MPEG) ISO/IEC, как продукт коллектива, известного как Объединенная команда разработчиков (JVT). В некоторых аспектах описанные в этом изобретении методики можно применить к устройствам, которые в целом соответствуют стандарту H.264. Стандарт H.264 описан в ITU-T Recommendation H.264, Advanced Video Coding для стандартных аудиовизуальных услуг исследовательской группой ITU-T в марте 2005 года, который далее называется здесь стандартом H.264, спецификацией H.264 или стандартом или спецификацией H.264/AVC. Объединенная команда разработчиков видео (JVT) продолжает работу над расширениями H.264/MPEG-4 AVC.

Видеокодер 20 и декодер 30 могут быть реализованы каждый в виде любого из множества подходящих схем кодера, таких как один или несколько микропроцессоров, цифровых процессоров сигналов (DSP), прикладных специализированных интегральных схем (ASIC), вентильных матриц, программируемых пользователем (FPGA), дискретной логики, программного обеспечения, аппаратного обеспечения, программно-аппаратного обеспечения или любой их комбинации. При частичной реализации этих методик программными средствами в устройстве могут храниться команды для этих программных средств на подходящем считываемом компьютером носителе длительного хранения, и устройство может выполнять эти команды аппаратными средствами, используя один или несколько процессоров для выполнения раскрытых здесь методик. Видеокодер 20 и декодер 30 могут входить в состав одного или нескольких кодеров либо декодеров, любой из которых может являться составной частью комбинированного кодера/декодера (CODEC) в соответствующем устройстве.

Команда JCT-VC работает в настоящее время над созданием стандарта HEVC. Работы по созданию стандарта HEVC основаны на усовершенствовании модели устройства видеокодирования, называемой тестовой моделью стандарта HEVC (HM). Модель HM предлагает несколько дополнительных функций устройств видеокодирования по сравнению с существующими устройствами, соответствующими, например, стандарту ITU-T H.264/AVC. В то время как стандарт H.264 обеспечивает девять режимов интракодирования, модель HM может обеспечить до тридцати трех режимов интракодирования.

В общем случае рабочая модель HM предполагает возможность разделения видеокадра или видеоизображения на последовательность древовидных блоков или максимальных единиц кодирования (LCU), которые включают в себя как отсчеты яркости, так и отсчеты цветности. Синтаксические данные в битовом потоке могут определить размер для LCU, которая является максимальной единицей кодирования с точки зрения количества пикселей. Слайс включает в себя несколько последовательных древовидных блоков в порядке кодирования. Видеокадр или видеоизображение можно разбить на один или несколько слайсов. Каждый древовидный блок можно разделить на единицы кодирования (CU) согласно квадродереву. В общем случае структура данных квадродерева включает в себя один узел на одну CU с корневым узлом, соответствующим указанному древовидному блоку. Если CU разделена на четыре суб-CU, то узел, соответствующий этой CU, включает в себя четыре концевых узла, каждый

из которых соответствует одной из суб-CU.

Каждый узел структуры данных квадродерева может обеспечить синтаксические данные для соответствующей CU. Например, узел в квадродереве может включать в себя флаг разделения, указывающий, разделена ли CU, соответствующая данному узлу, на несколько суб-CU. Синтаксические элементы для CU могут быть определены рекурсивно и зависеть от того, разделена ли данная CU на суб-CU. Если CU далее не разделяется, то она называется концевой CU. В этом изобретении четыре суб-CU концевой CU также называются концевыми CU, даже в том случае, если в явном виде отсутствует разделение исходной концевой CU. Например, если CU размером 16×16 далее не разделена, то четыре суб-CU размером 8×8 также будут называться концевыми CU, хотя CU размером 16×16 никогда не расщеплялась.

Единица CU имеет такое же назначение, как макроблок в стандарте H.264 за исключением того, что CU не имеет точно установленный размер. Например, древовидный блок может быть разделен на четыре дочерних узла (называемых также суб-CU), а каждый дочерний узел, в свою очередь, может быть родительским узлом и быть разделен на четыре других дочерних узла. Конечный неразделенный дочерний узел, называемый концевым узлом квадродерева, содержит узел кодирования, также называемый концевой CU. Синтаксические данные, связанные с кодированным битовым потоком могут определить максимально возможное количество разделений древовидного блока, называемое максимальной глубиной CU, а также могут определить минимальный размер узлов кодирования. Соответственно, битовый поток также может определить минимальную единицу кодирования (SCU). В настоящем изобретении для ссылки на любую из единиц (CU, PU или TU) в контексте стандарта HEVC используется термин «блок», или на аналогичные структуры данных в контексте других стандартов (например, макроблоки и их субблоки в стандарте H.264/AVC).

Единица CU включает в себя узел кодирования и единицы предсказания (PU), а также единицы преобразования (TU) связанные с узлом кодирования. Размер единицы CU соответствует размеру узла кодирования, причем она должна иметь форму квадрата. Размер CU может находиться в диапазоне от 8×8 пикселей до размера древовидного блока с максимальным размером 64×64 пикселей или более. Каждая CU может содержать одну или несколько PU и одну или несколько TU. Синтаксические данные, связанные с CU, могут, например, описывать разделения CU на одну или несколько PU. Режимы разделения могут отличаться друг от друга в зависимости от того, закодирована ли CU в режиме с интра-предсказанием или в режиме с интер-предсказанием. Единицы PU могут быть разделены на неквадратные части. Синтаксические данные, связанные с CU, могут также описывать, например, разделение CU на одну или несколько PU согласно квадродереву. Единица TU может иметь квадратную или неквадратную (например, прямоугольную) форму.

Стандарт HEVC позволяет выполнять преобразования в соответствии с единицами TU, которые могут отличаться для разных CU. Размеры TU, как правило, устанавливают на основе размера единиц PU в данной CU, определенной для LCU, хотя так может быть не всегда. Единицы TU, как правило, имеют такой же или меньший размер, чем единицы PU. В некоторых примерах остаточные отсчеты, соответствующие CU, могут быть подразделены на меньшие единицы с использованием структуры квадродерева, известной как «остаточное квадродерево» (RQT). Концевые узлы RQT могут называться единицами преобразования (TU). Значения пиксельной разности, связанные с единицами TU, можно преобразовать для создания коэффициентов преобразования, которые можно подвергнуть квантованию.

Концевая CU может включать в себя одну или несколько единиц предсказания (PU). В общем случае PU представляет пространственную область, соответствующую всем или части соответствующих CU и может включать в себя данные для извлечения опорного отсчета для данной PU. Кроме того, PU включает в себя данные, относящиеся к предсказанию. Например, когда PU кодируют с использованием интра-режима, данные для этой PU могут быть включены в остаточное квадродерево (RQT), которое может включать в себя данные, описывающие режим интра-предсказания для TU, соответствующей данной PU. В другом примере, когда при кодировании PU используется интер-режим, PU может содержать данные, определяющие один или несколько векторов движения для этой PU. Данные, определяющие вектор движения для PU, могут, например, описывать горизонтальную компоненту вектора движения, вертикальную компоненту вектора движения, разрешающую способность для вектора движения (например, с точностью в четверть пикселя или с точностью в одну восьмую пикселя), опорное изображение, на которое указывает вектор движения и/или список опорных изображений (например, список 0, список 1 или список C) для вектора движения.

Концевая CU, имеющая одну или несколько PU, также может включать в себя одну или несколько единиц преобразования (TU). Единицы преобразования могут быть заданы с использованием RQT (также называемым структурой квадродерева TU), как было описано выше. Например, флаг разделения может указывать, разделена ли концевая CU на четыре единицы преобразования. Затем каждая единица преобразования может быть дополнительно разделена на суб-TU. Когда TU далее не разделяется, ее можно назвать концевой TU. В общем случае для интра-кодирования все концевые TU, принадлежащие одной концевой CU, совместно используют один и тот же режим интра-предсказания. То есть для вычисления предсказанных значений для всех TU концевой CU обычно применяется один и тот же режим интра-предсказания. Для интра-кодирования видеокодер может вычислить остаточное значение для каждой концевой TU, используя режим интра-предсказания, в виде разности между частью CU, соответствующей TU, и исходным блоком. Размер TU нет необходимости ограничить размером PU. Таким образом, TU могут быть больше и меньше, чем PU. Для интра-кодирования PU может быть совмещена с соответствующей концевой TU для одной и той же CU. В некоторых примерах максимальный размер концевой TU может соответствовать размеру соответствующей концевой CU.

Кроме того, единицы TU концевых CU также могут быть связаны с соответствующими структурами данных квадродерева, называемыми остаточными квадродеревьями (RQT). То есть концевая CU может включать в себя квадродерево, указывающее, каким образом концевая CU разделена на единицы TU. В общем случае корневой узел квадродерева TU соответствует концевой CU, в то время как корневой узел квадродерева CU в общем случае соответствует древовидному блоку (или LCU). Неразделенные единицы TU квадродерева RQT, называются концевыми TU. В этом изобретении термины CU и TU обычно относятся к концевой CU и концевой TU соответственно, если не указано иное.

Видеопоследовательность, как правило, включает в себя ряд кадров или изображений. Группа изображений (GOP) обычно содержит ряд, состоящий из одного или нескольких видеоизображений. Группа GOP может включать в себя синтаксические данные в заголовке GOP, в заголовке одного или нескольких изображений или где-либо еще, которые описывают несколько изображений, включенных в группу GOP. Каждый слайс изображения может включать в себя синтаксические данные слайса, которые описывают режим кодирования для соответствующего слайса. Как правило, видеокодер 20 работает с видеоблоками в отдельных видеослайсах для выполнения кодирования видеоданных.

Видеоблок может соответствовать узлу кодирования в CU. Видеоблоки могут иметь фиксированные или переменные размеры и могут отличаться по размеру в зависимости от конкретного стандарта кодирования.

Например, модель НМ поддерживает предсказание для разных размеров PU.

- 5 Положим, что размер конкретной CU составляет $2N \times 2N$, модель НМ поддерживает интра-предсказание для размеров PU, составляющих $2N \times 2N$ или $N \times N$, и интер-предсказание для симметричных размеров PU, составляющих $2N \times 2N$, $2N \times N$, $N \times 2N$ или $N \times N$. Модель НМ также поддерживает асимметричное разбиение для интер-предсказания для размеров PU, составляющих $2N \times nU$, $2N \times nD$, $nL \times 2N$ и $nR \times 2N$. При асимметричном
- 10 разбиении в одном направлении блока CU разбиение отсутствует, в то время как в другом направлении выполняется разбиение на части в отношении 25% и 75%. Часть CU, соответствующая 25% части, обозначена как «n», после которой следует обозначение «Вверх», «Вниз», «Влево» или «Вправо». Таким образом, например, « $2N \times nU$ » относится к CU размером $2N \times 2N$, которая разбита по горизонтали на PU размером $2N \times 0,5N$ сверху
- 15 и PU размером $2N \times 1,5N$ снизу.

- В данном описании формы « $N \times N$ » и «N на N» могут использоваться как взаимозаменяемые для ссылки на пиксельные размеры видеоблока с точки зрения размеров по вертикали и горизонтали. Например, 16×16 пикселей или 16 на 16 пикселей. В общем случае блок размером 16×16 будет иметь 16 пикселей в вертикальном
- 20 направлении ($y=16$) и 16 пикселей в горизонтальном направлении ($x=16$). Аналогичным образом блок $N \times N$ обычно имеет N пикселей в вертикальном направлении и N пикселей в горизонтальном направлении, где N представляет не отрицательное целое значение. Пиксели в блоке могут быть скомпонованы по строкам и столбцам. Кроме того, блоки не обязательно должны иметь одинаковое количество пикселей в горизонтальном
- 25 направлении и в вертикальном направлении. Например, блоки могут содержать $N \times M$ пикселей, где M не обязательно равно N.

- Вслед за кодированием с интра-предсказанием или интер-предсказанием с использованием единиц PU единицы CU видеокодер 20 может вычислить остаточные данные для единиц TU единицы CU. Единицы PU могут содержать синтаксические
- 30 данные, описывающие способ или режим создания прогнозных пиксельных данных в пространственной области (также называемой областью пикселей), а единицы TU могут содержать коэффициенты в области преобразования после применения преобразования, например, дискретного косинусного преобразования (DCT), целочисленного преобразования, вейвлетного преобразования или концептуально подобного
- 35 преобразования, примененного к остаточным видеоданным. Остаточные данные могут соответствовать пиксельным разностям между пикселями не кодированного изображения и значениями предсказания, соответствующими единицам PU. Видеокодер 20 может образовать единицы TU, включающие в себя остаточные данные для CU, а затем преобразовать единицы TU для создания коэффициентов преобразования для
- 40 данной единицы CU.

- После любого преобразования для создания коэффициентов преобразования видеокодер 20 может выполнить квантование коэффициентов преобразования. Квантование в общем случае относится к процессу, в ходе которого выполняется квантование коэффициентов преобразования с целью возможного сокращения объема
- 45 данных, используемых для представления указанных коэффициентов, что обеспечивает дополнительное сжатие. Процесс квантования может уменьшить глубину в битах, связанную с некоторыми или всеми упомянутыми коэффициентами. Например, n-битовое значение можно округлить в меньшую сторону, получив m-битовое значение во время

квантования, где n больше, чем m .

Вслед за квантованием видеокодер может выполнить сканирование коэффициентов преобразования, создавая одномерный вектор из двумерной матрицы, содержащей квантованные коэффициенты преобразования. Сканирование может быть предназначено для перемещения коэффициентов с более высокой энергией (и, следовательно, с более низкой частотой) в переднюю часть матрицы, а коэффициенты с более низкой энергией (и, следовательно, с более высокой частотой) переместить в заднюю часть матрицы. В некоторых примерах видеокодер 20 может использовать заранее определенный порядок «сканирования» для сканирования квантованных коэффициентов преобразования, чтобы получить сериализованный вектор, который можно закодировать с использованием энтропийного кодирования. В других примерах видеокодер 20 может выполнять адаптивное сканирование. После сканирования квантованных коэффициентов преобразования для формирования одномерного вектора видеокодер 20 может выполнить кодирование этого одномерного вектора, используя энтропийное кодирование, например, в соответствии с контекстно-зависимым адаптивным кодированием с переменной длиной слова (CAVLC), контекстно-адаптивным бинарным арифметическим кодированием (CABAC), синтаксически ориентированным контекстно-адаптивным бинарным арифметическим кодированием (SBAC), вероятностным энтропийным кодированием с интервальным разбиением (PIPE), или другим методом энтропийного кодирования. Видеокодер 20 может также, выполнить энтропийное кодирование синтаксических элементов, связанных с кодированными видеоданными, для использования видеодекодером 30 при декодировании видеоданных.

Для выполнения кодирования CABAC видеокодер 20 может присвоить контекст в контекстной модели для символа, подлежащего передаче. Этот контекст может относиться, например, к тому, являются ли соседние значения символа ненулевыми. Для выполнения кодирования CAVLC видеокодер 20 может выбрать код переменной длины для символа, подлежащего передаче. Кодовые слова в VLC могут быть построены так, чтобы относительно более короткие коды соответствовали более вероятным символам, в то время как относительно более длинные коды соответствовали менее вероятным символам. Таким образом, использование кодирования VLC может обеспечить экономное использование бит, например, путем использования кодовых слов одинаковой длины для каждого символа, подлежащего передаче. Определение вероятности может быть основано на контексте, присвоенном данному символу.

Согласно раскрытым здесь методикам видеокодер 20 может выполнить кодирование значения, представляющего позицию последнего значащего коэффициентов блока видеоданных, используя контексты, определенные с использованием одной или нескольких функций бинов указанного значения. Аналогичным образом, видеодекодер 30 может декодировать значение, представляющее последний значащий коэффициент блока видеоданных, используя контексты, определенные с использованием одной или нескольких функций бинов указанного значения. Видеокодер 20 и/или видеодекодер 30 могут быть выполнены с возможностью выполнения любой из функций (1)-(12), более подробно описанных ниже, или концептуально подобных функций для выполнения раскрытых здесь методик. Таким образом, видеокодер 20 и видеодекодер 30 представляют примеры видеокодеров, выполненных с возможностью определения контекста энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, используя функцию индекса бина, и закодировать бин, используя определенный контекст.

Например, «Ctx_i» может обозначать индекс контекста, используемого видеокодером

20 для кодирования i -го бина в двоичной цепочке «последней позиции». Видеокодер 20 может получить Ctx_i , используя следующее уравнение:

$$Ctx_i = f(i)$$

Функция, обозначенная как $f(i)$, может быть линейной или нелинейной. Вдобавок, f
 5 (i) может быть заранее определенной функцией, доступной как видеокодеру 20, так и видеодекодеру 30. В качестве альтернативы, $f(i)$ может быть выбрано пользователем или видеокодером 20 и передано на видеодекодер 30 с использованием синтаксической сигнализации высокого уровня одного или нескольких типов, например, набор параметров последовательности (SPS), набор параметров изображения (PPS), набор
 10 параметров адаптации (APS), заголовок кадра, заголовок слайса, заголовок последовательности либо другая синтаксическая сигнализация. Примером одной из указанных функций, которую может выполнять видеокодер 20, может быть

$$f(i) = (i >> 1), (1)$$

где «>>» обозначает двоичный оператор сдвига вправо. В свою очередь, результат
 15 $f(i)$ может соответствовать Ctx_i . То есть видеокодер 20 может выполнить $f(i)$ для создания выхода, равного значению Ctx_i . В частности, видеокодер 20 может выполнить $f(i)$ для создания контекстного индекса для контекста, подлежащего использованию для энтропийного кодирования i -го бина.

В приведенной ниже Таблице 3 показан пример контекстных индексов, которые
 20 может использовать видеокодер 20 для кодирования бинов с разными индексами бинов для разных размеров блоков (например, TU), используя приведенную выше функцию (1). Хотя Таблица 3 предусмотрена с целью объяснения результатов использования приведенной в качестве примера функции (1), следует понимать, что таблицу, такую как Таблица 3, не обязательно хранить в устройстве видеокodирования, таком как
 25 устройство-источник 12 и/или устройство-адресат 14. Вместо этого видеокодер 20 и/или видеодекодер 30 могут выполнить функцию (1) для создания результатов, показанных в Таблице 3, на основе различных индексов бинов.

Таблица 3										
Индекс бина	0	1	2	3	4	5	6	7	8	9
TU 4×4	0	0	1							
TU 8×8	0	0	1	1	2					
TU 16×16	0	0	1	1	2	2	3			
TU 32×32	0	0	1	1	2	2	3	3	4	

В другом примере видеокодер 20 может выполнять функцию, зависящую как от
 35 индекса (i) бина, так и от размера соответствующего блока (например, TU). Соответствующим блоком может быть блок, включающий в себя коэффициенты, описанные значением последнего значащего коэффициента. Например, контекстный индекс может быть создан функцией, такой как:

$Ctx_i = f(i, TUBlkSize)$, где «TUBlkSize» - значение, указывающее размер блока. В
 40 настоящем изобретении термины «TUBlkSize» и «block_size» могут использоваться как взаимозаменяемые для указания размера блока.

В одном примере указанная функция может быть представлена в виде:

$$f(i, TUBlkSize) = i >> (\log_2(TUBlkSize) - 2) \quad (2)$$

В приведенной ниже Таблице 4 показан пример контекстных индексов, которые
 45 может использовать видеокодер 20 для кодирования бинов с разными индексами бинов для различных размеров блоков (например, TU), используя приведенную в качестве примера функцию (2). Хотя Таблица 4 предусмотрена с целью объяснения результатов использования приведенной в качестве примера функции (2), следует понимать, что

таблицу, такую как Таблица 4, не обязательно хранить в устройстве видеокодирования, таком как устройство-источник 12 и/или устройство-адресат 14. Вместо этого видеокодер 20 и/или видеodeкодер 30 могут выполнить приведенную в качестве примера функцию (2) для создания результатов, показанных в Таблице 4.

Таблица 4

Индекс бина	0	1	2	3	4	5	6	7	8	9
TU 4×4	0	1	2							
TU 8×8	0	0	1	1	2					
TU 16×16	0	0	0	0	1	1	1			
TU 32×32	0	0	0	0	0	0	0	0	1	

В еще одном примере видеокодер 20 может выполнить следующую функцию для получения Ctx_i:

$f(i, \text{TUblkSize}) = i \gg 1 + \text{TUSIZEoffset}$, где

$\text{TUSIZEoffset} = \log_2(\text{TUblkSize}) - 2 * (\log_2(\text{TUblkSize}) + 1) / 2$ (3)

В приведенной ниже Таблице 5 показан пример контекстных индексов, которые может использовать видеокодер 20 для кодирования бинов с разными индексами бинов для разных размеров блоков (например, TU), используя приведенную выше функцию (3). Хотя Таблица 5 предусмотрена с целью объяснения результатов использования приведенной в качестве примера функции (3), следует понимать, что таблицу, такую как Таблица 3, не обязательно хранить в устройстве-источнике 12 и/или устройстве-адресате 14. Вместо этого видеокодер 20 и/или видеodeкодер 30 могут выполнить приведенную в качестве примера функцию (3), описанную выше, для создания результатов, показанных в Таблице 5.

Таблица 5

Индекс бина	0	1	2	3	4	5	6	7	8	9
TU 4×4	0	0	1							
TU 8×8	2	2	3	3	4					
TU 16×16	5	5	6	6	7	7	8			
TU 32×32	9	9	10	10	11	11	12	12	13	

В следующем примере видеокодер 20 может выполнить следующую функцию для получения Ctx_i:

$\text{Ctx_idx} = (i+1) \gg 1 + \text{TUSIZEoffset}$, где

$\text{TUSIZEoffset} = (\log_2(\text{TUblkSize}) - 2) * (\log_2(\text{TUblkSize}) + 1) / 2$. (4)

В приведенной ниже Таблице 6 показан пример контекстных индексов, которые может использовать видеокодер 20 для кодирования бинов с разными индексами бинов для разных размеров блоков (например, TU), используя приведенную выше функцию (4). Хотя Таблица 6 предусмотрена с целью объяснения результатов использования приведенной в качестве примера функции, следует понимать, что таблицу, такую как Таблица 6, не обязательно хранить в устройстве-источнике 12 и/или устройстве-адресате 14. Вместо этого видеокодер 20 и/или видеodeкодер 30 могут выполнить приведенную в качестве примера функцию (4), описанную выше, для создания результатов, показанных в Таблице 6.

Таблица 6

Индекс бина	0	1	2	3	4	5	6	7	8	9
TU 4×4	0	1	1							
TU 8×8	2	3	3	4	4					
TU 16×16	5	6	6	7	7	8	8			

TU 32×32	9	10	10	11	11	12	12	13	13	
----------	---	----	----	----	----	----	----	----	----	--

В еще одном примере упомянутая функция может представлять собой:

$\text{Ctx_idx} = \text{offset} + (i \gg k)$, (5)

где:

$\text{offset} = 3 * n + ((n+1) \gg 2)$, (6)

$k = (n+3) \gg 2$, и (7)

$n = (\log_2(\text{TUblkSize}) - 2)$ (8)

В качестве альтернативы, приведенная в качестве примера функция (8) может быть представлена в виде $n = (\log_2(\text{block_size}) - 2)$, имея в виду цели настоящего изобретения.

В приведенной ниже Таблице 7 показан пример контекстных индексов, которые может использовать видеокодер 20 для кодирования бинов с разными индексами бинов для разных размеров блоков (например, TU), используя приведенные в качестве примера функции (5)-(8). Хотя Таблица 7 предусмотрена с целью объяснения результатов использования указанных функций, следует понимать, что таблицу, такую как Таблица 7, не обязательно хранить в устройстве видеокodирования, таком как устройство-источник 12 и/или устройство-адресат 14. Вместо этого видеокодер 20 и/или видеodeкодер 30 могут выполнить приведенные в качестве примера функции (5)-(8), описанные выше, для создания результатов, показанных в Таблице 7

Таблица 7

Индекс бина	0	1	2	3	4	5	6	7	8	9
TU 4×4	0	1	2							
TU 8×8	3	3	4	4	5					
TU 16×16	6	6	7	7	8	8	9			
TU 32×32	10	10	11	11	12	12	13	13	14	

В таблицах 8 и 9, показанных ниже, приведен другой пример, в котором видеокодер 20 и/или видеodeкодер 30 может применить одну или несколько методик получения контекста на основе формулы для бинов при кодировании «последней позиции» для компонент яркости и цветности унифицированным образом. В частности, в Таблице 8 показаны индексы бинов для TU яркости различных размеров, в то время как в Таблице 9 представлены индексы бинов для TU цветности различных размеров.

Таблица 8
Яркость

Индекс бина	0	1	2	3	4	5	6	7	8	9
TU 4×4	0	1	2							
TU 8×8	3	3	4	4	5					
TU 16×16	6	6	7	7	8	8	9			
TU 32×32	10	10	11	11	12	12	13	13	14	

Таблица 9
Цветность

Индекс бина	0	1	2	3	4	5	6	7	8	9
TU 4×4	0	1	2							
TU 8×8	0	0	1	1	2					
TU 16×16	0	0	1	1	2	2	3			

В одном примере функция, которую видеокодер 20 и/или видеodeкодер 30 могут использовать для получения контекстов для бинов при кодировании последней позиции единиц TU яркости (Таблица 8) и единиц TU цветности (Таблица 9), представляет собой:

$\text{Ctx_idx} = \text{offset} + (i \gg k)$, (9)

где для яркости и цветности используется одно и то же значение k , $k = (n+3) \gg 2$ при

$n = (\log_2(\text{TUblkSize}) - 2)$.

Видеокодер и/или декодер 30 могут определить значения для переменной «offset» функции (9) на основе того, является ли единица TU единицей TU яркости или единицей TU цветности, с использованием различных функций. Примеры указанных функций включают в себя следующее:

Яркость: $\text{offset} = 3 * n + ((n + 1) >> 2)$ (10)

Цветность: $\text{offset} = 0$ (11)

Таким образом, функция (9) представляет пример функции, которую может выполнить видеокодер 20 и/или декодер 30 для создания контекстного индекса. В свою очередь, контекстный индекс может указывать на контекст для кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных в качестве функции индекса бина (i) и значения, указывающего размер блока (k, которое вычисляют на основе n, представляющего собой $\log_2(\text{TUblkSize}) - 2$). В этом примере видеокодер 20 и/или декодер 30 также могут выполнять приведенную в качестве примера функцию (9) для создания контекстного индекса на основе значения смещения, которое определяют на основе того, является ли данный блок блоком цветности или блоком яркости, например, как показано в функциях (10) и (11).

В другом примере видеокодер 20 может реализовать пошаговую функцию для получения контекстного индекса для контекста, подлежащего использованию для энтропийного кодирования i-го бина. В частности, пошаговая функция может представлять функцию, состоящую из двух или более частей, в зависимости, например, от значения индекса i бина. Таким образом, видеокодер 20 и/или декодер 30 могут разделить бины в значении последней позиции на разные поднаборы, например, поднабор0, поднабор1 и т.д. Вдобавок, видеокодер 20 и/или декодер 30 могут применить разные функции к разным поднаборам, например, F0() для поднабора (), F1() для поднабора 1 и т.д. Например, указанная функция может быть представлена следующим образом:

$$f(i, \text{TUblkSize}) = \begin{cases} i = \text{last_bin}, & 10 \\ \text{иначе} & (i >> 1) + \text{TUSIZEOffset} \end{cases}, \text{ где}$$

$$\text{TUSIZEOffset} = (\log_2(\text{TUblkSize}) - 2) * (\log_2(\text{TUblkSize}) - 1) / 2 \quad (12)$$

В некоторых реализациях указанные поднаборы могут быть определены заранее, причем определение этих поднаборов может быть доступно как видеокодеру 20, так и декодеру 30. В качестве альтернативы, видеокодер 20 (или пользователь устройства-источника 12) может выбрать указанные поднаборы, а выходной интерфейс 22 может передать выбранные поднаборы на декодер 30 устройства-адресата 14, используя одну или несколько методик синтаксической сигнализации высокого уровня, такой как передача SPS, PPS, APS, заголовка кадра, заголовка слайса, заголовка последовательности либо использовать иную синтаксическую сигнализацию. Определение поднаборов также может зависеть от информации различных других типов, такой как: размер блока (например, размер TU); глубина остаточного квадродерева (RQT), соответствующего данному блоку; информация о том, соответствует ли данный блок компоненте яркости или компоненте цветности; размер кадра для кадра, включающего в себя данный блок (например, пиксельном разрешении); размер блока компенсации движения для блока компенсации движения (например, блок предсказания (PU)), соответствующего данному блоку; тип кадра (I/P/B) для кадра, включающего в себя данный блок; направление интер-предсказания для соответствующего блока компенсации движения; амплитуда вектора движения для

соответствующего блока компенсации движения; и/или амплитуда разности векторов движения для вектора движения, соответствующего блоку компенсации движения.

В Таблице 10, представленной ниже, показан пример контекстных индексов, которые может использовать видеокодер 20 для кодирования бинов с разными индексами для разных размеров блока (например, TU), используя приведенную в качестве примера функцию (12). Хотя Таблица 10 приведена здесь с целью объяснения результатов указанной функции, следует понимать, что таблица, такая как Таблица 10, не обязательно хранится в устройстве видеокодирования, таком как устройство-источник 12 и/или устройство-адресат 14. Вместо этого, видеокодер 20 и/или видеodeкодер 30 могут выполнять вышеописанную приведенную в качестве примера функцию (12) для получения результатов, указанных в Таблице 10.

Таблица 10										
Индекс бина	0	1	2	3	4	5	6	7	8	9
TU 4x4	0	0	10							
TU 8x8	1	1	2	2	10					
TU 16x16	3	3	4	4	5	5	10			
TU 32x32	6	6	7	7	8	8	9	9	10	

Приведенные в качестве примера функции (1)-(12), описанные выше, могут зависеть по меньшей мере частично от одного или нескольких элементов дополнительной информации. В одном примере эти функции могут использовать дополнительную информацию в качестве аргументов. В других примерах видеокодер 20 и/или видеodeкодер 30 могут выбрать разные функции на основе соответствующей дополнительной информации. Дополнительная информация может включать в себя любой или все из следующих параметров: размер блока (например, размер TU); глубина остаточного квадродерева (RQT), соответствующего данному блоку; информация о том, соответствует ли данный блок компоненте яркости или компоненте цветности; размер кадра для кадра, включающего в себя данный блок (например, разрешающей способности в пикселях); размер блока компенсации движения для блока компенсации движения (например, блока предсказания (PU)) соответствующего данному блоку; тип кадра (I/P/B) для кадра, включающего в себя данный блок; направление интер-предсказания для соответствующего блока компенсации движения; амплитуда вектора движения для соответствующего блока компенсации движения; и/или амплитуда разности векторов движения для вектора движения, соответствующего блоку компенсации движения. В одном примере видеокодер 20 и/или видеodeкодер 30 могут выбрать разные функции для получения контекстов с целью их применения при кодировании бинов значения, указывающего позицию последнего значащего коэффициента блока яркости по отношению к блоку цветности.

Видеокодер 20 может кроме того, послать синтаксические данные, такие как синтаксические данные на блочной основе, синтаксические данные на покaдровой основе и синтаксические данные на основе GOP на видеodeкодер 30, например, в заголовке кадра, в заголовке блока, в заголовке слайса или заголовке GOP. Синтаксические данные GOP могут описывать количество кадров в соответствующей группе GOP, а синтаксические данные кадра могут указывать режим кодирования/предсказания, используемый для кодирования соответствующего кадра.

Видеокодер 20 и видеodeкодер 30 могут быть реализованы каждый в виде любой из различных подходящих схем кодера или декодера, если они применимы, таких как один или несколько микропроцессоров, цифровых процессоров сигналов (DSP), прикладных специализированных интегральных схем (ASIC), вентильных матриц, программируемых

пользователем (FPGA), дискретной логики, программного обеспечения, аппаратного обеспечения, программно-аппаратного обеспечения или любой их комбинации. Как видеокодер 20, так и декодер 30 могут быть включены в один или несколько кодеров или декодеров, любой из которых может являться составной частью комбинированного видеокодера/декодера (CODEC). Устройство, включающее в себя видеокодер 20 и/или декодер 30, может содержать интегральную схему, микропроцессор и/или устройство беспроводной связи, такое как сотовый телефон.

Таким образом, видеокодер 20 и декодер 30 представляют примеры видеокодера, выполненного с возможностью определения контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса данного бина и кодирования этого бина с использованием определенного контекста.

На фиг. 2 представлена блок-схема, где показан пример видеокодера 20, который может реализовать методики для определения контекста с целью его использования для кодирования значения, представляющего последний значащий коэффициент блока видеоданных. Видеокодер 20 может выполнять интра- и интер-кодирование видеоблоков в видеослайсах. Интра-кодирование основано на пространственном предсказании с целью уменьшения или устранения пространственной избыточности в видео в данном видеокadre или видеоизображении. Интер-кодирование основано на временном предсказании, которое уменьшает или устраняет временную избыточность видео в соседних кадрах или изображениях видеопоследовательности. Интра-режим (I режим) может относиться к любому из нескольких режимов кодирования на пространственной основе. Интер-режимы, такие как однонаправленное предсказание (P режим) или двунаправленное предсказание (B режим) могут относиться к любому из нескольких режимов кодирования на временной основе.

Как показано на фиг. 2, видеокодер 20 принимает текущий видеоблок в видеокadre, подлежащем кодированию. В примере на фиг. 2 видеокодер 20 включает в себя блок 40 выбора режима, память 64 опорных кадров, сумматор 50, блок 52 обработки преобразования, блок 54 квантования и блок 56 энтропийного кодирования. Блок 40 выбора режима, в свою очередь, содержит блок 44 компенсации движения, блок 42 оценки движения, блок 46 интра-предсказания и блок 48 разделения. Для восстановления видеоблока видеокодер 20 также содержит блок 58 обратного квантования, блок 60 обратного преобразования и сумматор 62. Сюда также может быть включен деблокирующий фильтр (на фиг. 2 не показан) для фильтрации границ блоков с целью удаления артефактов «блочности» из восстановленного видео. Если это необходимо, деблокирующий фильтр, как правило, фильтрует выход сумматора 62. Вдобавок к деблокирующему фильтру также могут быть использованы дополнительные фильтры (в контуре или после контура). Указанные фильтры для краткости не показаны, но если это необходимо, они могут фильтровать выход сумматора 50 (как контурные фильтры).

В ходе процесса кодирования видеокодер 20 принимает видеокadre или видеослайс, подлежащий кодированию. Этот кадр или слайс можно разделить на множество видеоблоков. Блок 42 оценки движения и блок 44 компенсации движения выполняет кодирование с интер-предсказанием принятого видеоблока применительно к одному или нескольким блокам в одном или нескольких опорных кадрах для обеспечения временного предсказания. Блок 46 интра-предсказания может попеременно выполнять кодирование с интра-предсказанием принятого видеоблока применительно к одному или нескольким соседним блокам в том же кадре или слайсе в качестве блока,

подлежащего кодированию, для обеспечения пространственного предсказания. Вideoкодер 20 может выполнять множество проходов кодирования, например, для выбора подходящего режима кодирования для каждого блока видеоданных.

Кроме того, блок 48 разделения может разделить блоки видеоданных на субблоки на основе оценки предыдущих схем разделения в предшествующих проходах кодирования. Например, блок 48 разделения вначале может разделить кадр или слайс на несколько LCU и разделить каждую LCU на несколько суб-CU на основе анализа зависимости «скорость - искажение» (например, оптимизация «скорость - искажение»). Блок 40 выбора режима может далее создать структуру данных квадродерева, указывающую разделение LCU на несколько суб-CU. Единицы CU конечного узла квадродерева могут включать в себя одну или несколько PU и одну или несколько TU.

Блок 40 выбора режима может выбрать один из режимов кодирования (интра- или интеркодирование), например, на основе результатов с ошибками, при этом блок 40 подает результирующий интра- или интер-закодированный блок на сумматор 50 для создания остаточных данных блока и на сумматор 62 для восстановления кодированного блока для его использования в качестве опорного кадра. Блок 40 выбора режима также подает синтаксические элементы, такие как векторы движения, индикаторы интра-режима, информацию о разделении и другую синтаксическую информацию в блок 56 энтропийного кодирования.

Блок 42 оценки движения и блок 44 компенсации движения могут быть интегрированы с высокой степенью интеграции, но они показаны отдельно, исходя из концептуальных целей. Оценка движения, выполняемая блоком 42 оценки движения, представляет собой процесс создания векторов движения, которые оценивают движение для видеоблоков. Вектор движения, например, может указывать смещение PU видеоблока в текущем видеокadre или видеоизображении относительно предсказанного блока в опорном кадре (или в другой кодированной единице). Предсказанный блок - это блок, который, как было обнаружено, близко соответствует блоку, подлежащему кодированию с точки зрения пиксельной разности, которую можно определить как сумму абсолютных разностей (SAD), сумму квадратичных разностей (SSD) или другие метрики разностей. В некоторых примерах видеокодер 20 может вычислить значения для пиксельных позиций опорных изображений (с точностью до долей пикселя), хранящихся в памяти 64 опорных кадров. Например, видеокодер 20 может интерполировать значения четвертных пиксельных позиций, пиксельных позиций с точностью одна восьмая пикселя или других дробных пиксельных позиций опорного изображения. Таким образом, блок 42 оценки движения может выполнить поиск движения относительно полных пиксельных позиций и дробных пиксельных позиций и выдать вектор движения с точностью, выраженной дробным числом пикселей.

Блок 42 оценки движения вычисляет вектор движения для PU видеоблока в интеркодированном слайсе путем сравнения позиции PU с позицией предсказанного блока опорного изображения. Опорное изображение можно выбрать из первого списка опорных изображений (список 0) или второго списка опорных изображений (список 1), каждый из которых идентифицирует одно или несколько опорных изображений, хранящихся в памяти 64 опорных кадров. Блок 41 оценки движения посылает вычисленный вектор движения в блок 56 энтропийного кодирования и блок 44 компенсации движения.

Компенсация движения, выполняемая блоком 44 компенсации движения, может включать в себя извлечение или создание предсказанного блока на основе вектора движения, определенного блоком 42 оценки движения. Опять же, в некоторых примерах

блок 42 оценки движения и блок 44 компенсации движения могут быть функционально интегрированы. После получения вектора движения для PU текущего видеоблока блок 44 компенсации движения может определить местоположение предсказанного блока, на который указывает вектор движения в одном из списков опорных изображений.

5 Сумматор 50 формирует остаточный видеоблок посредством вычитания значений пикселей предсказанного блока из значений пикселей текущего кодируемого видеоблока, получая значения разности пикселей, как описано ниже. В общем случае блок 42 оценки движения выполняет оценку движения применительно к компонентам яркости, а блок 44 компенсации движения использует векторы движения, вычисленные на основе
10 компонент яркости, для обеих компонент цветности и яркости. Блок 40 выбора режима также может создавать синтаксические элементы, связанные с видеоблоками и видеослайсом для использования видеодекодером 30 при декодировании видеоблоков видеослайса.

Блок 46 интра-предсказания может выполнить интра-предсказание для текущего
15 блока в качестве альтернативы интер-предсказанию, выполняемому блоком 42 оценки движения и блоком 44 компенсации движения, как было описано выше. В частности, блок 46 интра-предсказания может определить режим интра-предсказания для использования во время кодирования текущего блока. В некоторых примерах блок 46 интра-предсказания может выполнить кодирование текущего блока, используя разные
20 режимы интра-предсказания, например, во время отдельных проходов кодирования, и блок 46 интра-предсказания (или блок 40 выбора режима в некоторых примерах) может выбрать соответствующий режим интра-предсказания для его использования из числа тестовых режимов.

Например, блок 46 интра-предсказания может вычислить значения «скорость-
25 искажение» с использованием анализа «скорость-искажение» для различных тестовых режимов интра-предсказания и выбирает режим интра-предсказания, имеющий наилучшие характеристики «скорость-искажение» среди тестовых режимов. В ходе анализа «скорость-искажение» обычно определяют величину искажения (или ошибку) между кодированным блоком и исходным, не кодированным блоком, который был
30 кодирован для создания кодированного блока, а также скорость передачи битов (то есть количество бит), используемых для создания кодированного блока. Блок 46 интра-предсказания может вычислить отношения исходя из искажений и скоростей для разных кодированных блоков, чтобы определить, какой из режимов интра-предсказания дает наилучшее соотношение «скорость-искажение» для данного блока.

35 После выбора режима интра-предсказания для блока пикселей блок 46 интра-предсказания может подать информацию, указывающую выбранный режим интра-предсказания для данного блока, в блок 56 энтропийного кодирования. Блок 56 энтропийного кодирования может выполнить кодирование информации, указывающей выбранный режим интра-предсказания. Видеокодер 20 может содержать данные о
40 конфигурации передаваемого битового потока, которые могут включать в себя множество таблиц с индексами режимов интра-предсказания и множество модифицированных таблиц режимов интра-предсказания (также называемых «таблицами отображения кодовых слов»), определения контекстов кодирования для различных блоков и указания наиболее вероятного режима интра-предсказания, таблицу индексов режимов интра-предсказания и модифицированную таблицу индексов режимов интра-предсказания с целью их использования для каждого из упомянутых контекстов.

Видеокодер 20 формирует остаточный видеоблок путем вычитания данных предсказания от блока 40 выбора режима из кодируемого исходного видеоблока.

Сумматор 50 представляет компоненту или компоненты, которые выполняют эту операцию вычитания. Блок 52 обработки преобразования применяет преобразование, такое как дискретное косинусное преобразование (DCT) или концептуально подобное преобразование к остаточному блоку, создавая видеоблок, содержащий остаточные значения коэффициентов преобразования. Блок 52 обработки преобразования может выполнить другие преобразования, концептуально подобные DCT. Также могут быть использованы вейвлетные преобразования, целочисленные преобразования, субполосные преобразования или преобразования других типов. В любом случае блок 52 обработки преобразования применяет преобразование к остаточному блоку, создавая блок остаточных коэффициентов преобразования. Такое преобразование может привести к преобразованию остаточной информации из области значений пикселей в область преобразования, такую как частотная область. Блок 52 обработки преобразования может послать результирующие коэффициенты преобразования в блок 54 квантования. Блок 54 квантования выполняет квантование коэффициентов преобразования для дальнейшего уменьшения битрейта. Процесс квантования может уменьшить глубину в битах, связанную с некоторыми или всеми упомянутыми коэффициентами. Степень квантования можно модифицировать посредством настройки параметра квантования. В некоторых примерах блок 54 квантования может затем выполнить сканирование матрицы, включающей в себя квантованные коэффициенты преобразования. В качестве альтернативы, блок 56 энтропийного кодирования может выполнить сканирование.

Вслед за квантованием блок 56 энтропийного кодирования может выполнить контекстно-зависимое адаптивное кодирование с переменной длиной слова (CAVLC), контекстно-адаптивное бинарное арифметическое кодирование (CABAC), синтаксически ориентированное контекстно-адаптивное бинарное арифметическое кодирование (SBAC), вероятностное энтропийное кодирование с интервальным разбиением (PIPE), или другие методы энтропийного кодирования. В случае контекстно-зависимого энтропийного кодирования контекст может быть основан на соседних блоках. Вслед за энтропийным кодированием, выполняемым блоком 56 энтропийного кодирования, кодированный битовый поток может быть передан на другое устройство (например, видеodeкодер 30) или архивирован для позднейшей передачи или извлечения.

В частности, блок 56 энтропийного кодирования может принять от блока 54 квантования набор квантованных коэффициентов преобразования, связанных с TU. В свою очередь, блок 56 энтропийного кодирования может отсканировать набор квантованных коэффициентов преобразования и определить, содержит ли каждый отсканированный коэффициент значащий коэффициент, то есть имеет ли коэффициент нулевое или ненулевое значение. Ненулевое значение может указывать на то, что конкретный квантованный коэффициент преобразования является «значащим» коэффициентом. В случаях, когда блок 56 энтропийного кодирования обнаруживает значащий коэффициент, он может кодировать данные, представляющие конкретное значение, связанное с этим коэффициентом (например, один, два и т.д.). Указанные данные могут включать в себя, например, указание о знаке коэффициента, информацию о том, превышает ли абсолютное значение данного коэффициента единицу, и, когда абсолютное значение коэффициента превышает единицу, информацию о том, превышает ли абсолютное значение данного коэффициента два. Вдобавок, в тех случаях, когда значащий коэффициент имеет абсолютное значение, превышающее два, блок 56 энтропийного кодирования может вычесть два из абсолютного значения коэффициента, получив тем самым значение, на которое данный коэффициент превышает два, а затем закодировать это значение.

Благодаря сканированию всего набора квантованных коэффициентов преобразования, полученных от блока 54 квантования, блок 56 энтропийного кодирования может также обнаружить и идентифицировать последний значащий коэффициент, связанный с конкретной TU (то есть в порядке сканирования). Вдобавок, блок 56 энтропийного кодирования может определить позицию последнего значащего коэффициента в соответствующей PU. Например, блок 56 энтропийного кодирования может идентифицировать координаты по горизонтали и вертикали x и y последнего значащего коэффициента в данной TU.

Кроме того, блок 56 энтропийного кодирования может быть выполнен с возможностью бинаризации синтаксических элементов, которые уже не имеют двоичного значения. То есть блок 56 энтропийного кодирования может определить бинарную цепочку, представляющую значение синтаксического элемента, когда синтаксический элемент уже не представлен бинарной цепочкой. Бинарная цепочка или бинаризованное значение обычно соответствует матрице бит, каждый из которых может иметь значение «0» или «1». Эта матрица может быть индексирована, начиная с нуля, так что по порядку первый бит матрицы оказывается на позиции 0, второй по порядку бит матрицы оказывается на позиции 1 и т.д. Таким образом, блок 56 энтропийного кодирования может сформировать бинаризованное значение $B[N]$ длиной N бит, где каждый бит появляется на соответствующей позиции $B[i]$, где $0 \leq i \leq N-1$.

В свою очередь, блок 56 энтропийного кодирования может выполнить энтропийное кодирование данных, представляющих координаты x и y последнего значащего коэффициента. Например, блок 56 энтропийного кодирования может быть выполнен с возможностью энтропийного кодирования синтаксических элементов `last_significant_coeff_x_prefix`, `last_significant_coeff_y_prefix`, `last_significant_coeff_x_suffix` и/или `last_significant_coeff_y_suffix`, которые в стандарте HEVC представляют вместе координаты x и y последнего значащего коэффициента в порядке сканирования. Блок 56 энтропийного кодирования может реализовать одну или несколько раскрытых здесь методик для энтропийного кодирования данных, представляющих координаты последнего значащего коэффициента, используя функцию, обозначенную как $f(i)$. Например, блок 56 энтропийного кодирования может выполнить энтропийное кодирование различных синтаксических элементов, таких как синтаксические элементы, предназначенные для квантованных коэффициентов преобразования, полученных из блока 54 квантования, и/или значения, представляющие последний значащий коэффициент TU (например, вышеописанные синтаксические элементы), используя контексты, определенные с использованием одной или нескольких функций бинов значения, представляющего соответствующий синтаксический элемент.

Например, «Ctx_i» может обозначать индекс контекста, используемого блоком 56 энтропийного кодирования для кодирования i -го бина в бинаризованном значении, представляющем позицию последнего значащего коэффициента, как было описано в связи с таблицами 1-2 и 8-9. Контекст, индексированный с помощью Ctx_i обычно указывает наиболее вероятный символ (например, «1» или «0»), а также вероятность наиболее вероятного символа. Блок 56 энтропийного кодирования может получить значение Ctx_i, используя уравнение $Ctx_i = f(i)$, где $f(i)$ может представлять собой заранее определенную функцию, доступную для блока 56 энтропийного кодирования, или функцию, выбранную пользователем. Вдобавок, блок 56 энтропийного кодирования может кодировать данные, представляющие $f(i)$, так что видеodeкодер 30 может декодировать данные для функции $f(i)$ и использовать $f(i)$ для получения значения Ctx_i. Действуя таким образом, блок 56 энтропийного кодирования может определить контекст

для конкретного бина бинаризованного синтаксического элемента, используя функцию индекса бина, то есть позицию бина в бинаризованном значении (то есть в бинарной цепочке), представляющем синтаксический элемент.

В некоторых примерах блок 56 энтропийного кодирования выполнен с возможностью
5 определения контекстов для кодирования бинов данных, представляющих позицию последнего значащего коэффициента, с использованием формул (5)-(8), описанных выше. То есть блок 56 энтропийного кодирования может вычислить $f(i)$ следующим образом: $\text{Ctx_idx} = \text{offset} + (i > k)$. Кроме того, блок 56 энтропийного кодирования может получить значения offset (смещения) и k , используемые в $f(i)$, пользуясь следующими
10 уравнениями:

$$\text{offset} = 3 * n + ((n + 1) >> 2),$$

$$k = (n + 3) >> 2 \text{ и}$$

$$n = (\log_2(\text{block_size}) - 2).$$

В других реализациях блок 56 энтропийного кодирования может использовать одну
15 или несколько приведенных в качестве примера функций (1)-(4) и (9)-(12) вдобавок или в качестве альтернативы формулам (5)-(8) при определении контекста для энтропийного кодирования бина данных, представляющих позицию последнего значащего коэффициента TU. Таким путем видеокодер 20 и его компоненты, такие как блок 56 энтропийного кодирования, могут реализовать раскрытые здесь методики для
20 кодирования данных, представляющих последний значащий коэффициент, с использованием одной или нескольких функций. Указанные функции выгоднее хранить в памяти видеокодера 20 и видеodeкодера 30, чем в виде таблиц. Таким образом, раскрытые здесь методики могут обеспечить видеокодеры и видеodeкодеры, которые используют память более эффективно, например, благодаря распределению памяти,
25 которая, в противном случае, была бы предоставлена таблице для других данных, или благодаря уменьшению требуемого объема памяти для видеокодера или видеodeкодера.

Блок 58 обратного квантования и блок 60 обратного преобразования выполняют соответственно обратное квантование и обратное преобразование для восстановления
остаточного блока пиксельной области, например, для использования в дальнейшем
30 в качестве опорного блока. Блок 44 компенсации движения может вычислить опорный блок путем добавления остаточного блока к предсказанному блоку одного из кадров, хранящихся в памяти 64 опорных кадров. Блок 44 компенсации движения также может применить один или несколько интерполяционных фильтров к восстановленному остаточному блоку для вычисления дробных значений пикселей для их использования
35 при оценке движения. Сумматор 62 добавляет восстановленный остаточный блок к блоку предсказания, созданному блоком 44 компенсации движения, для создания восстановленного видеоблока для его хранения в памяти 64 опорных кадров. Восстановленный видеоблок может быть использован блоком 42 оценки движения и блоком 44 компенсации движения в качестве опорного бока для интер-кодирования
40 блока в последующем видеокадре.

Таким путем видеокодер 20 по фиг. 2 представляет пример видеокодера, выполненного с возможностью определения контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, используя функцию индекса данного бина, и для кодирования этого бина с
45 использованием определенного контекста. Кроме того, видеокодер 20 также представляет пример видеокодера, в котором упомянутая функция создает контекстный индекс для контекста посредством сдвига вправо индекса бина на величину k и добавления этого сдвинутого вправо значения к значению смещения, где значение

смещения определяют в соответствии с формулой

$\text{offset} = 3 * n + ((n + 1) \gg 2)$, где значение k определяют согласно формуле $k = (n + 3) \gg 2$ и где значение n определяют согласно формуле $n = (\log_2(\text{block_size}) - 2)$.

5 На фиг. 3 представлена блок-схема, иллюстрирующая примерный видеodeкодер 30, который может реализовать методики для определения контекста с целью их использования для кодирования значения, представляющего последний значащий коэффициент блока видеоданных. В примере на фиг. 3 видеodeкодер 30 включает в себя блок 70 энтропийного декодирования, блок 72 компенсации движения, блок 74 интра-
10 предсказания, блок 76 обратного квантования, блок 78 обратного преобразования, память 82 опорных кадров и сумматор 80. В некоторых примерах видеodeкодер 30 может выполнять прогон декодирования, который является в общем случае обратным по отношению к проходу кодирования, описанному применительно к видеodeкодеру 20 (фиг. 2).

15 Блок 72 компенсации движения может создать данные предсказания на основе векторов движения, полученных от блока 70 энтропийного декодирования, в то время как блок 74 интра-предсказания может создать данные предсказания на основе индикаторов режимов интра-предсказания, полученных от блока 70 энтропийного декодирования.

20 В ходе процесса декодирования видеodeкодер 30 принимает кодированный битовый видеопоток, который представляет видеоблоки кодированного видеослайса и связанные с ними синтаксические элементы от видеodeкодера 20. Блок 70 энтропийного декодирования в составе видеodeкодера 30 выполняет энтропийное декодирование битового потока для создания квантованных коэффициентов, векторов движения или
25 индикаторов режимов интра-предсказания и других синтаксических элементов. Блок 70 энтропийного декодирования направляет векторы движения и другие синтаксические элементы в блок 72 компенсации движения. Видеodeкодер 30 может получить эти синтаксические элементы на уровне видеослайса и/или на уровне видеоблока.

Блок 70 энтропийного декодирования может создать блок (например, TU)
30 квантованных коэффициентов путем энтропийного декодирования кодированного битового видеопотока и распределения декодированных квантованных коэффициентов в блоке в порядке сканирования. Например, блок 70 энтропийного декодирования может выполнить энтропийное декодирование синтаксических элементов кодированного видеопотока для определения местоположений значащих коэффициентов в создаваемом
35 блоке. Если местоположение блока соответствует коэффициенту, не являющемуся значащим коэффициентом, блок 70 энтропийного декодирования может установить значение коэффициента для этого местоположения в блоке равным нулю. С другой стороны, если блок 70 энтропийного декодирования определяет, что конкретный квантованный коэффициент является значащим коэффициентом, то блок 70
40 энтропийного декодирования может установить значение значащего коэффициента на основе данных, обеспеченных в кодированном битовом потоке видео видеodeкодером 20.

Кроме того, как объясняется ниже, блок 70 энтропийного декодирования может
45 определить позицию последнего значащего коэффициента в блоке на основе синтаксических элементов, указывающих координаты x и y последнего значащего коэффициента. Согласно раскрытым здесь методикам, как более подробно объясняется ниже, блок 70 энтропийного декодирования может использовать функцию для определения контекста, необходимого для энтропийного декодирования бинов значений,

представляющих координаты x и y последнего значащего коэффициента. Видеодекодер 30 может использовать индикацию о позиции последнего значащего коэффициента для определения того, когда данные битового потока представляют последовательные синтаксические элементы, то есть синтаксические элементы, которые не представляют

данные восстанавливаемого блока.

Блок 70 энтропийного декодирования может определить на основе данных, обеспеченных в кодированном битовом потоке видео, знак для каждого значащего коэффициента и данные, представляющие значение уровня каждого значащего коэффициента. Например, блок 70 энтропийного декодирования может определить знак для конкретного коэффициента посредством энтропийного декодирования синтаксического элемента, представляющего знак, например, `coeff_sign_flag`. Вдобавок, блок 70 энтропийного декодирования может декодировать один или несколько синтаксических элементов, представляющих значение уровня каждого значащего коэффициента, например, `coeff_abs_level_greater1_flag`, `coeff_abs_level_greater2_flag` и `coeff_abs_level_remaining`. В общем случае `coeff_abs_level_greater1_flag` указывает, превышает ли 1 абсолютное значение значащего коэффициента, `coeff_abs_level_greater2_flag` указывает, превышает ли 2 абсолютное значение значащего коэффициента, а `coeff_abs_level_remaining` указывает абсолютное значение значащего коэффициента минус 2.

Блок 70 энтропийного декодирования может также определить позицию последнего значащего коэффициента восстанавливаемого блока (например, TU). В частности, блок 70 энтропийного декодирования может идентифицировать позицию (например, на основе кодированных синтаксических элементов, представляющих координаты x и y) последнего значащего коэффициента в единице TU, связанной с кодированным битовым потоком видео. На основе идентификации позиции последнего значащего коэффициента блок 70 энтропийного декодирования может установить равными нулю значения остальных коэффициентов в TU в порядке сканирования. То есть видеодекодеру 30 нет необходимости получать какие-либо синтаксические элементы для коэффициентов, следующих за последним значащим коэффициентом, и может затем положить значения 0 для этих коэффициентов.

Вдобавок, блок 70 энтропийного декодирования может реализовать одну или несколько раскрытых здесь методик для декодирования бинов бинаризованного значения, представляющего координаты x и y позиции последнего значащего коэффициента с использованием функции, обычно обозначаемой как $f(i)$, где i соответствует позиции данного бина в бинаризованном значении. В некоторых примерах блок 70 энтропийного декодирования может декодировать кодированные данные, используя определенный контекст для воспроизведения значения для данного бина, например, «0» или «1». Хотя описанные здесь методики относятся к методикам для определения позиции последнего значащего коэффициента, их также можно применить к другим синтаксическим элементам энтропийного декодирования. Например, блок 70 энтропийного декодирования может выполнять энтропийное декодирование разных синтаксических элементов, таких как синтаксические элементы квантованных коэффициентов, посылаемых в блок 72 компенсации движения и/или блок 74 интра-предсказания, синтаксических элементов, представляющих квантованные коэффициенты преобразования, и/или значения, представляющие последний значащий коэффициент единицы TU, связанной с кодированным битовым потоком видео, используя контексты, определенные с использованием одной или нескольких функций индексов бинов для значения, представляющего соответствующий синтаксический элемент.

Например, «Ctx_i» может обозначать индекс контекста, используемого блоком 70 энтропийного декодирования для декодирования i-го бина в бинаризованном значении, представляющем позицию последнего значащего коэффициента, как было описано в связи с таблицами 1-2 и 8-9. В этом примере блок 70 энтропийного декодирования
 5 может получить значение Ctx_i, используя уравнение $Ctx_i = f(i)$, где $f(i)$ может представлять собой заранее определенную функцию, доступную для блока 70 энтропийного декодирования (например, переданную устройством-источником 12) или функцию, выбранную пользователем. Вдобавок, блок 70 энтропийного декодирования может декодировать данные, с тем чтобы использовать данные, представляющие $f(i)$,
 10 для получения значения Ctx_i.

В некоторых примерах блок 70 энтропийного декодирования выполнен с возможностью определения контекстов для декодирования бинов данных, представляющих позицию последнего значащего коэффициента, с использованием формул (5)-(8), описанных выше. То есть блок 70 энтропийного декодирования может
 15 вычислить $f(i)$ следующим образом: $Ctx_idx = offset + (i >> k)$. Кроме того, блок 70 энтропийного декодирования может получить значения offset (смещения) и k, используемые в $f(i)$, пользуясь следующими уравнениями:

$$offset = 3 * n + ((n + 1) >> 2),$$

$$k = (n + 3) >> 2 \text{ и}$$

$$n = (\log_2(block_size) - 2).$$

В других реализациях блок 70 энтропийного декодирования может установить $f(i)$ согласно одному или нескольким из приведенных в качестве примера уравнений (1)-(4) и (9)-(12) при декодировании последнего значащего коэффициента TU, представленного
 25 кодированным битовым потоком видео. Таким путем видеodeкодер 30 и его компоненты, такие как блок 70 энтропийного декодирования, могут реализовать раскрытые здесь методики для декодирования последнего значащего коэффициента с использованием одной или нескольких функций. Указанные функции выгоднее хранить в памяти видеodeкодера 20 и видеodeкодера 30, чем соответствующие таблицы. Таким образом, раскрытые здесь методики могут обеспечить видеodeкодеры и видеodeкодеры, которые
 30 используют память более эффективно, например, благодаря распределению памяти, которая, в противном случае, была бы предоставлена таблице для других данных, или благодаря уменьшению требуемого объема памяти для видеodeкодера или видеodeкодера.

Когда видеослайс закодирован с использованием интра-предсказания (I), блок 74 интра-предсказания может создать данные предсказания для видеоблока текущего
 35 видеослайса на основе сигнализации в режиме интра-предсказания, а также на основе данных из ранее декодированных блоков текущего кадра или изображения. Когда видеокادر закодирован в виде слайса с интер-предсказанием (то есть B, P или GPB) блок 72 компенсации движения создает предсказанные блоки для видеоблока текущего видеослайса на основе векторов движения и других синтаксических элементов,
 40 полученных от блока 70 энтропийного декодирования. Предсказанные блоки можно создать из одного из опорных изображений в одном из списков опорных изображений. Видеodeкодер 30 может построить списки опорных кадров, список 0 и список 1, используя стандартные методики построения на основе опорных изображений, хранящихся в памяти 82 опорных изображений.

Блок 72 компенсации движения определяет информацию предсказания для видеоблока
 45 текущего видеослайса путем синтаксического анализа векторов движения и других синтаксических элементов и использует информацию предсказания для создания предсказанных блоков для текущего декодируемого видеоблока. Например, блок 72

компенсации движения использует некоторые из полученных синтаксических элементов для определения режима предсказания (например, интра- или интер-предсказание), используемого для кодирования видеоблоков видеослайса, типа слайса с интер-предсказанием (например, В слайс, Р слайс или GPB слайс), информации о структуре для одного или нескольких списков опорных изображений для данного слайса, векторов движения для каждого видеоблока слайса с интер-кодированием, статуса интер-предсказания для каждого видеоблока слайса, закодированного с интер-предсказанием, и другой информации для декодирования видеоблоков в текущем видеослайсе.

Блок 72 компенсации движения также может выполнить интерполяцию на основе интерполяционных фильтров. Блок 72 компенсации движения может использовать интерполяционные фильтры, как это делается видеокодером 20 в процессе кодирования видеоблоков, для вычисления интерполированных значений для дробных пикселей опорных блоков. В этом случае блок 72 компенсации движения может определить интерполяционные фильтры, используемые видеокодером 20, исходя из полученных синтаксических элементов, и использовать эти интерполяционные фильтры для создания предсказанных блоков.

Блок 76 обратного квантования выполняет обратное квантование, то есть деквантование квантованных коэффициентов преобразования, предусмотренных в битовом потоке и декодируемых блоком 70 энтропийного декодирования. Процесс обратного квантования может включать в себя использование параметра квантования QP_Y , вычисленного видеодекодером 30 для каждого видеоблока в видеослайсе, для определения степени квантования по аналогии со степенью обратного квантования, которую и следует применить.

Блок 78 обратного преобразования применяет обратное преобразование, например, обратное DCT преобразование, обратное целочисленное преобразование или концептуально подобный процесс обратного преобразования к коэффициентам преобразования, чтобы обеспечить остаточные блоки в пиксельной области.

После того, как блок 72 компенсации движения создал предсказанный блок для текущего видеоблока на основе векторов движения и других синтаксических элементов, видеодекодер 30 формирует декодированный видеоблок посредством суммирования остаточных блоков, полученных от блока 78 обратного преобразования, с соответствующими предсказанными блоками, созданными блоком 72 компенсации движения. Сумматор 80 представляет компоненту или компоненты, которые выполняют операцию суммирования. Если это необходимо, деблокирующий фильтр также может быть применен для фильтрации декодированных блоков с целью удаления артефактов блочности. Для сглаживания переходов от пикселя к пикселю или иного улучшения качества видео также можно использовать другие контурные фильтры (либо в контуре кодирования, либо после контура кодирования). Затем декодированные видеоблоки в данном кадре или изображении запоминаются в памяти 82 опорных изображений, где хранятся опорные изображения, используемые для последующей компенсации движения. В памяти 82 опорных изображений также запоминается декодированное видео для его представления в дальнейшем на устройстве отображения, таком как устройство 32 отображения по фиг. 1.

Таким образом видеодекодер 30 по фиг. 3 представляет пример видеодекодера, выполненного с возможностью определения контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, используя функцию индекса данного бина, и для кодирования этого бина с использованием определенного контекста. Кроме того, видеодекодер 30 также

представляет пример видеodeкодера, в котором упомянутая функция создает контекстный индекс для контекста посредством сдвига вправо индекса бина на величину k и добавления этого сдвинутого вправо значения к значению смещения, где значение смещения определяют в соответствии с формулой $\text{offset}=3*n+((n+1)>>2)$, где значение k определяют согласно формуле $k=(n+3)>>2$ и где значение n определяют согласно формуле $n=(\log_2(\text{block_size})-2)$.

На фиг. 4 представлена блок-схема, иллюстрирующая примерный способ для кодирования текущего блока. Текущий блок может содержать текущую CU или часть текущей CU. Хотя данное описание относится к видеокодеру 20 (фиг. 1 и 2) следует понимать, что для выполнения способа, аналогичного способу по фиг. 4, можно сконфигурировать другие устройства. Кроме того, хотя в примерном способе по фиг. 4 конкретно описывается кодирование синтаксических элементов, относящихся к позиции последнего значащего коэффициента видеоблока с использованием указанных методик, следует понимать, что эти методики можно использовать также для кодирования других синтаксических элементов.

В этом примере видеокодер 20 сначала выполняет предсказание текущего блока (шаг 150). Например, видеокодер 20 может вычислить одну или несколько единиц предсказания (PU) для текущего блока. Затем видеокодер 20 может вычислить остаточный блок для текущего блока, например, для создания единицы преобразования (TU) (этап 152). Для вычисления остаточного блока видеокодер 20 может вычислить разность между исходным, не кодированным блоком и предсказанным блоком для текущего блока. Затем видеокодер 20 может выполнить преобразование и квантование коэффициентов остаточного блока (этап 154). Далее видеокодер 20 может выполнить сканирование квантованных коэффициентов преобразования остаточного блока (этап 156). Во время сканирования или вслед за сканированием видеокодер 20 может выполнить энтропийное кодирование упомянутых коэффициентов (этап 158). Например, видеокодер 20 может кодировать коэффициенты, используя CAVLC или CABAC.

Видеокодер 20 может также определить значение для позиции последнего значащего коэффициента в TU (этап 160). Это значение может, например, содержать бинаризованное значение, представляющее позицию последнего значащего коэффициента, например, как это описано выше в связи с Таблицей 1. Максимальное количество бинов упомянутого значения можно закодировать, используя CABAC, в то время как оставшиеся бины можно закодировать, используя обходное кодирование, как это было описано в связи с Таблицей 1. В частности, согласно раскрытым здесь методикам, видеокодер 20 может определить контексты для бинов упомянутого значения, используя функцию (этап 162). Как объяснялось выше, указанные контексты могут описывать вероятности бинов, имеющих конкретное значение, например, «0» или «1». Указанная функция может соответствовать одной из вышеописанных функций (1)-(12) или концептуально подобных функций.

В связи с примерами функций (5)-(8) видеокодер 20 может определить контекст ctx_idx для бина на позиции i бинаризованного значения, представляющего позицию последнего значащего бина, используя формулу $\text{offset}+(i>k)$, где $\text{offset}=3*n+((n+1)>>2)$, $k=(n+3)>>2$ и $n=(\log_2(\text{block_size})-2)$, то есть видеокодер 20 может выполнять итерации по каждому бину, подлежащему энтропийному кодированию, и выполнять функции, указанные выше, для определения контекста, необходимого для кодирования бина текущей итерации. Затем видеокодер 20 может выполнить кодирование бинов для данного значения (например, бинов, не выходящих за указанное максимальное количество бинов), используя определенные контексты (этап 164). Аналогичным образом, видеокодер 20

может выполнить обходное кодирование остальных бинов указанного значения (этап 166).

Таким образом, способ по фиг. 4 представляет пример способа, включающего в себя определение контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса данного бина, и кодирования данного бина с использованием определенного контента. Кроме того, указанная функция может создать индекс для данного контекста путем сдвига вправо индекса бина на величину k и добавления сдвинутого вправо значения к значению смещения, где значение смещения определяют согласно формуле $offset = 3 * n + ((n + 1) >> 2)$, где значение k определяют согласно формуле $k = (n + 3) >> 2$, и где значение n определяют согласно формуле $n = (\log_2(block_size) - 2)$.

На фиг. 5 представлена блок-схема, иллюстрирующая примерный способ для декодирования текущего блока видеоданных. Текущий блок может содержать текущую CU или часть текущей CU. Хотя данное описание относится к видеодекодеру 30 (фиг. 1 и 3) следует понимать, что для выполнения способа, аналогичного способу по фиг. 5, можно сконфигурировать другие устройства. Кроме того, хотя в примерном способе по фиг. 4 конкретно описывается кодирование синтаксических элементов, относящихся к позиции последнего значащего коэффициента видеоблока с использованием указанных методик, следует понимать, что эти методики можно использовать также для кодирования других синтаксических элементов.

Видеодекодер 30 может выполнить предсказание текущего блока (200), например, используя режим интра- или интер-предсказания с целью вычисления предсказанного блока для текущего блока. Видеодекодер 30 также может принимать данные для текущего блока после энтропийного кодирования, такие как данные (после энтропийного кодирования) для коэффициентов остаточного блока, соответствующего текущему блоку (этап 202). Видеодекодер 30 может выполнить энтропийное декодирование кодированных данных для воспроизведения коэффициентов остаточного блока (этап 204).

Согласно раскрытым здесь методикам видеодекодер 30 может получить кодированное значение, указывающее позицию последнего значащего коэффициента в TU (этап 206). Максимальное количество бинов упомянутого значения можно декодировать, используя CABAC, в то время как оставшиеся бины можно декодировать, используя обходное декодирование, как это было описано в связи с Таблицей 1. В частности, согласно раскрытым здесь методикам, видеодекодер 30 может определить контексты для бинов упомянутого значения, используя функцию (этап 208). Как объяснялось выше, указанные контексты могут описывать вероятности бинов, имеющих конкретное значение, например, «0» или «1». Указанная функция может соответствовать одной из вышеописанных функций (1)-(12) или концептуально подобной функции.

В связи с примерами функций (5)-(8) видеодекодер 30 может определить контекст ctx_idx для бина на позиции i декодируемого бинаризованного значения, где бинаризованное значение представляет позицию последнего значащего коэффициента, используя формулу $offset + (i >> k)$, где $offset = 3 * n + ((n + 1) >> 2)$, $k = (n + 3) >> 2$ и $n = (\log_2(block_size) - 2)$. То есть видеодекодер 30 может итеративно декодировать каждый бин, подлежащий энтропийному декодированию, и выполнять функции, указанные выше, для определения контекста, необходимого для кодирования бина текущей итерации. Затем видеодекодер 30 может декодировать бины значения (например, бины, вошедшие в указанное максимальное количество бин), используя определенные контексты (этап 210). Например, видеодекодер 30 может декодировать кодированные данные,

полученные от видеокодера 20, используя определенные контексты для воспроизведения или иного получения бинов указанного значения. Аналогичным образом, видеокодер 30 может выполнить обходное декодирование оставшихся бинов указанного значения (этап 212).

5 Затем видеодекодер 30 может выполнить обратное сканирование воспроизведенных коэффициентов на основе позиции последнего значащего коэффициента (этап 214) для создания блока квантованных коэффициентов преобразования. То есть видеодекодер 30 может разместить декодированные коэффициенты в TU, начиная с позиции последнего значащего коэффициента и продолжая размещение коэффициентов в порядке
10 сканирования, который в общем случае соответствует порядку сканирования, использованного кодером. Затем декодер 30 может выполнить обратное квантование и обратное преобразование указанных коэффициентов для создания остаточного блока (этап 216). Видеодекодер 30 может, наконец, может декодировать текущий блок путем комбинирования предсказанного блока и остаточного блока (этап 218).

15 Таким образом, способ по фиг. 5 представляет пример способа, включающего в себя определение контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса данного бина, и кодирования данного бина с использованием определенного контента. Кроме того, указанная функция может создать индекс для данного контекста
20 путем сдвига вправо индекса бина на величину k и добавления сдвинутого вправо значения к значению смещения, где значение смещения определяют согласно формуле $\text{offset} = 3 * n + ((n + 1) \gg 2)$, где значение k определяют согласно формуле $k = (n + 3) \gg 2$, и где значение n определяют согласно формуле $n = (\log_2(\text{block_size}) - 2)$.

25 Должно быть ясно, что в зависимости от того или иного примера некоторые действия или события, связанные с раскрытыми здесь методиками, могут выполняться в разной последовательности, с возможностью добавления новых действий или событий, объединение нескольких имеющихся действий или событий, а также исключения тех или иных действий или событий (например, когда не все описанные действия или события
30 необходимы для практической реализации предложенных методик). Кроме того, в некоторых примерах действия или события могут выполняться одновременно, например, посредством обработки с прерыванием или посредством обработки множеством процессоров, а не последовательно.

В одном или нескольких примерах описанные функции могут быть реализованы аппаратными средствами, программными средствами, программно-аппаратными
35 средствами или любой их комбинацией. При реализации программными средствами эти функции могут храниться или передаваться в виде одной или нескольких команд или кода на считываемом компьютером носителе и выполняться обрабатывающим аппаратным блоком. Считываемая компьютером среда может включать в себя считываемую компьютером запоминающую среду, которая соответствует
40 материальному носителю, такому как среда хранения данных или среда передачи, включая любой носитель, который позволяет обеспечить пересылку компьютерной программы из одного места в другое, например, согласно какому-либо протоколу связи. В этом аспекте считываемая компьютером среда в общем случае может соответствовать (1) материальной считываемой компьютером запоминающей среде,
45 которая обеспечивает долговременное хранение, или (2) среде связи, такой как сигнал или несущая волна. Средой для запоминания данных может служить любая доступная среда, доступ к которой может осуществляться одним или несколькими компьютерами или одним или несколькими процессорами для извлечения команд, кода и/или со стороны

структур данных для реализации раскрытых в этом изобретении методик. Компьютерный программный продукт может включать в себя считываемый компьютером носитель.

В качестве примера, но не как ограничение, указанный считываемый компьютером носитель данных может содержать RAM, ROM, EEPROM, CD-ROM или другое запоминающее устройство на оптических дисках, запоминающее устройство на магнитных дисках или другие магнитные запоминающие устройства, флэш-память, или любой другой носитель, который можно использовать для запоминания требуемого программного кода в виде команд или структур данных, и к которым может иметь доступ компьютер. Также считываемой компьютером средой будет правильно назвать любое соединение. Например, если команды передаются с Web-сайта, сервера или другого удаленного источника с использованием коаксиального кабеля, оптоволоконного кабеля, скрученной пары, цифровой абонентской линии (DSL) или беспроводных технологий, таких как инфракрасная связь, радиосвязь и микроволновая связь, то тогда коаксиальный кабель, оптоволоконный кабель, витая пара, линии DSL или такие беспроводные технологии, как инфракрасная связь, радиосвязь и микроволновая связь, входят в определение среды. Однако следует понимать, что термины: «считываемый компьютером носитель данных» и «среда для хранения данных» не включают в себя соединения, несущие волны, сигналы или другие временные носители, а относятся к материальной запоминающей среде долговременного хранения. Используемый здесь термин оптический диск относится к компакт-диску (CD), лазерному диску, оптическому диску, цифровому универсальному диску (DVD), флоппи-диску и диску Blu-ray, воспроизводящих данные оптическим путем с использованием лазеров, в то время как термин магнитный диск обычно относится к дискам, воспроизводящим данные магнитным путем. В понятие считываемой компьютером среды также следует отнести комбинации из вышеперечисленного.

Команды могут выполняться одним или несколькими процессорами, такими как один или несколько цифровых процессоров сигналов (DSP), микропроцессоров общего назначения, прикладных специализированных интегральных схем (ASIC), вентильных матриц, программируемых пользователем (FPGA), или других эквивалентных интегральных или дискретных логических схем. Соответственно, используемый здесь термин «процессор» может относиться к любой из вышеописанных структур или любой иной структуре, подходящей для реализации вышеописанных методик. Вдобавок, в некоторых аспектах описанные здесь функциональные возможности могут быть обеспечены в рамках специализированных аппаратных и/или программных модулей, выполненных с возможностью кодирования и декодирования, или включенных в состав комбинированного кодека. Также раскрытые здесь методики можно полностью реализовать в одной или нескольких схемах или логических элементах.

Раскрытые здесь методики могут быть реализованы в самых разных устройствах или установках, включая беспроводную телефонную трубку, интегральную схему (IC) или набор интегральных схем (например, микропроцессорный набор). Различные компоненты, модули или блоки описаны здесь таким образом, чтобы подчеркнуть функциональные аспекты устройств, выполненных с возможностью выполнения раскрытых здесь методик, причем для этого не требуется их реализация с помощью различных аппаратных блоков. Скорее наоборот, как обсуждалось выше, различные блоки могут быть скомбинированы в аппаратном блоке-кодеке или обеспечены в виде набора взаимодействующих аппаратных блоков, включая один или несколько процессоров, как было описано выше, в сочетании с подходящим программным и/или

программно-аппаратным обеспечением.

Здесь были описаны различные примеры. Эти и другие примеры входят в объем нижеследующей формулы изобретения.

(57) Формула изобретения

1. Способ кодирования видеоданных, причем способ содержит:
определение контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных, с использованием функции индекса упомянутого бина,

в котором функция создает контекстный индекс для контекста путем сдвига вправо индекса бина на величину k и добавления сдвинутого вправо значения к значению смещения,

причем значение смещения определяют согласно формуле:

$$\text{offset} = 3 * n + ((n + 1) \gg 2),$$

причем значение k определяют согласно формуле:

$$k = (n + 3) \gg 2, \text{ и}$$

причем значение n определяют согласно формуле:

$$n = (\log_2(\text{block_size}) - 2), \text{ и}$$

причем значение block_size содержит значение, указывающее размер блока, и кодирование упомянутого бина с использованием определенного контекста.

2. Способ по п. 1, в котором определение контекста содержит выполнение упомянутой функции.

3. Способ по п. 1, в котором функция содержит линейную функцию.

4. Способ по п. 1, в котором функция содержит нелинейную функцию.

5. Способ по п. 1, в котором функция создает контекстный индекс для контекста путем сдвига вправо индекса бина на единицу.

6. Способ по п. 1, дополнительно содержащий прием упомянутой функции от пользователя.

7. Способ по п. 1, дополнительно содержащий прием синтаксических данных, определяющих упомянутую функцию.

8. Способ по п. 1, в котором кодирование бина содержит энтропийное декодирование кодированных данных с использованием определенного контекста для воспроизведения значения для бина.

9. Способ по п. 1, в котором кодирование бина содержит энтропийное кодирование бина с использованием определенного контекста.

10. Устройство для кодирования видеоданных, причем устройство содержит видеокодер, выполненный с возможностью:

определения контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных с использованием функции индекса упомянутого бина,

в котором функция создает контекстный индекс для контекста путем сдвига вправо индекса бина на величину k и добавления сдвинутого вправо значения к значению смещения,

причем значение смещения определяют согласно формуле:

$$\text{offset} = 3 * n + ((n + 1) \gg 2),$$

причем значение k определяют согласно формуле:

$$k = (n + 3) \gg 2, \text{ и}$$

причем значение n определяют согласно формуле:

$n = (\log_2(\text{block_size}) - 2)$,

причем значение `block_size` содержит значение, указывающее размер блока, и кодирования упомянутого бина с использованием определенного контекста.

11. Устройство по п. 10, в котором видеокодер выполнен с возможностью определения контекста по меньшей мере частично путем выполнения упомянутой функции.

12. Устройство по п. 10, в котором видеокодер дополнительно выполнен с возможностью приема синтаксических данных, определяющих упомянутую функцию.

13. Устройство по п. 10, в котором видеокодер выполнен с возможностью кодирования бина по меньшей мере частично путем энтропийного декодирования кодированных данных с использованием определенного контекста для воспроизведения значения для бина.

14. Устройство по п. 10, в котором видеокодер выполнен с возможностью кодирования бина по меньшей мере частично путем энтропийного кодирования бина с использованием определенного контекста.

15. Устройство по п. 10, в котором устройство содержит по меньшей мере одно из: интегральную схему;

микропроцессор; и

устройство беспроводной связи, содержащее упомянутый видеокодер.

16. Устройство для кодирования видеоданных, причем устройство содержит: средство для определения контекста для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных с использованием функции индекса упомянутого бина,

в котором функция создает контекстный индекс для контекста путем сдвига вправо индекса бина на величину k и добавления сдвинутого вправо значения k к значению смещения,

причем значение смещения определяют согласно формуле:

$\text{offset} = 3 * n + ((n + 1) \gg 2)$,

причем значение k определяют согласно формуле:

$k = (n + 3) \gg 2$,

причем значение n определяют согласно формуле:

$n = (\log_2(\text{block_size}) - 2)$, и

причем значение `block_size` содержит значение, указывающее размер блока, и средство для кодирования упомянутого бина с использованием определенного контекста.

17. Устройство по п. 16, в котором средство для определения контекста содержит средство для выполнения упомянутой функции.

18. Устройство по п. 16, дополнительно содержащее средство для приема синтаксических данных, определяющих упомянутую функцию.

19. Считываемый компьютером носитель данных, кодированный командами, которые при их выполнении предписывают программируемому процессору вычислительного устройства:

определять контекст для энтропийного кодирования бина значения, указывающего последний значащий коэффициент блока видеоданных с использованием функции индекса упомянутого бина,

причем функция создает контекстный индекс для контекста путем сдвига вправо индекса бина на величину k и добавления сдвинутого вправо значения k к значению смещения,

причем значение смещения определяют согласно формуле:

$\text{offset}=3*n+((n+1)>>2),$

причем значение k определяют согласно формуле:

$k=(n+3)>>2,$

причем значение n определяют согласно формуле:

5 $n=(\log_2(\text{block_size})-2),$ и

причем значение block_size содержит значение, указывающее размер блока, и кодировать упомянутый бин с использованием определенного контекста.

20. Считываемый компьютером носитель данных по п. 19, в котором команды, предписывающие программируемому процессору определять упомянутый контекст,
10 дополнительно включают в себя команды, предписывающие программируемому процессору выполнять упомянутую функцию.

21. Считываемый компьютером носитель данных по п. 19, дополнительно кодированный командами, которые при их выполнении предписывают программируемому процессору принимать синтаксические данные, определяющие
15 упомянутую функцию.

20

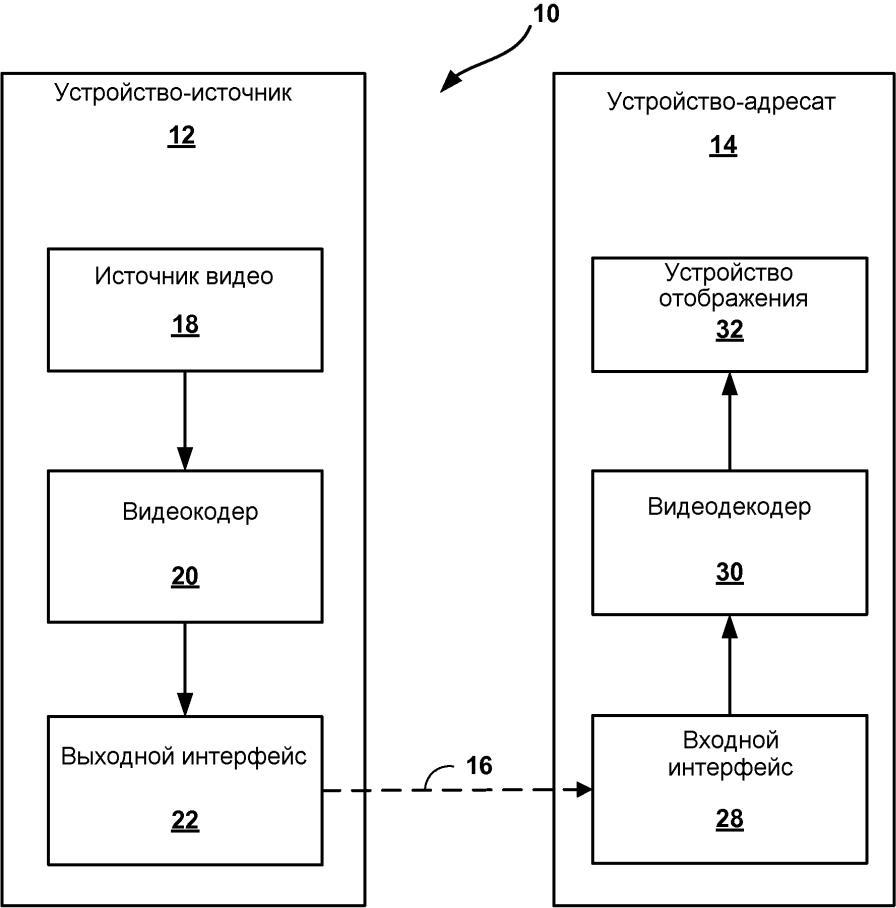
25

30

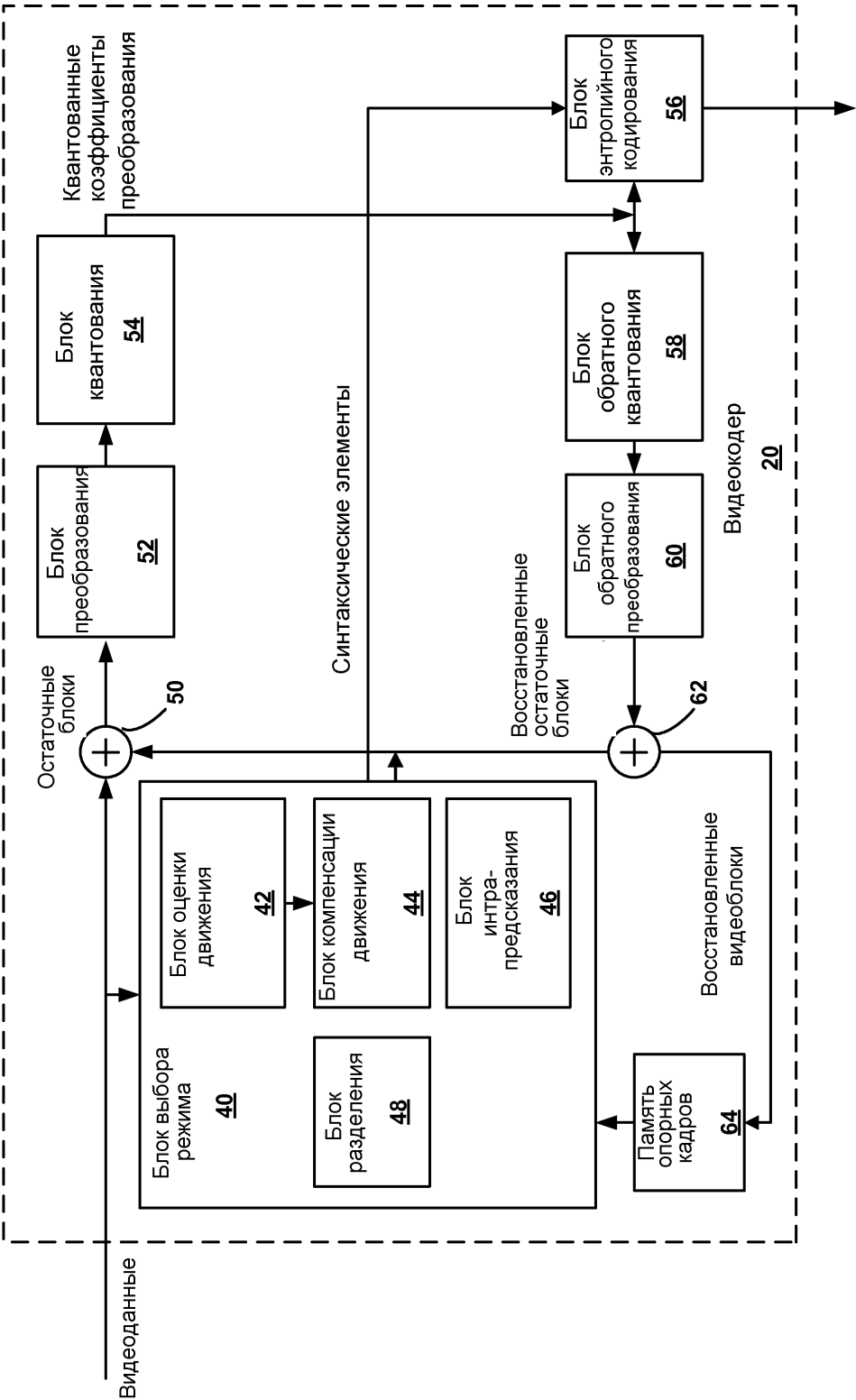
35

40

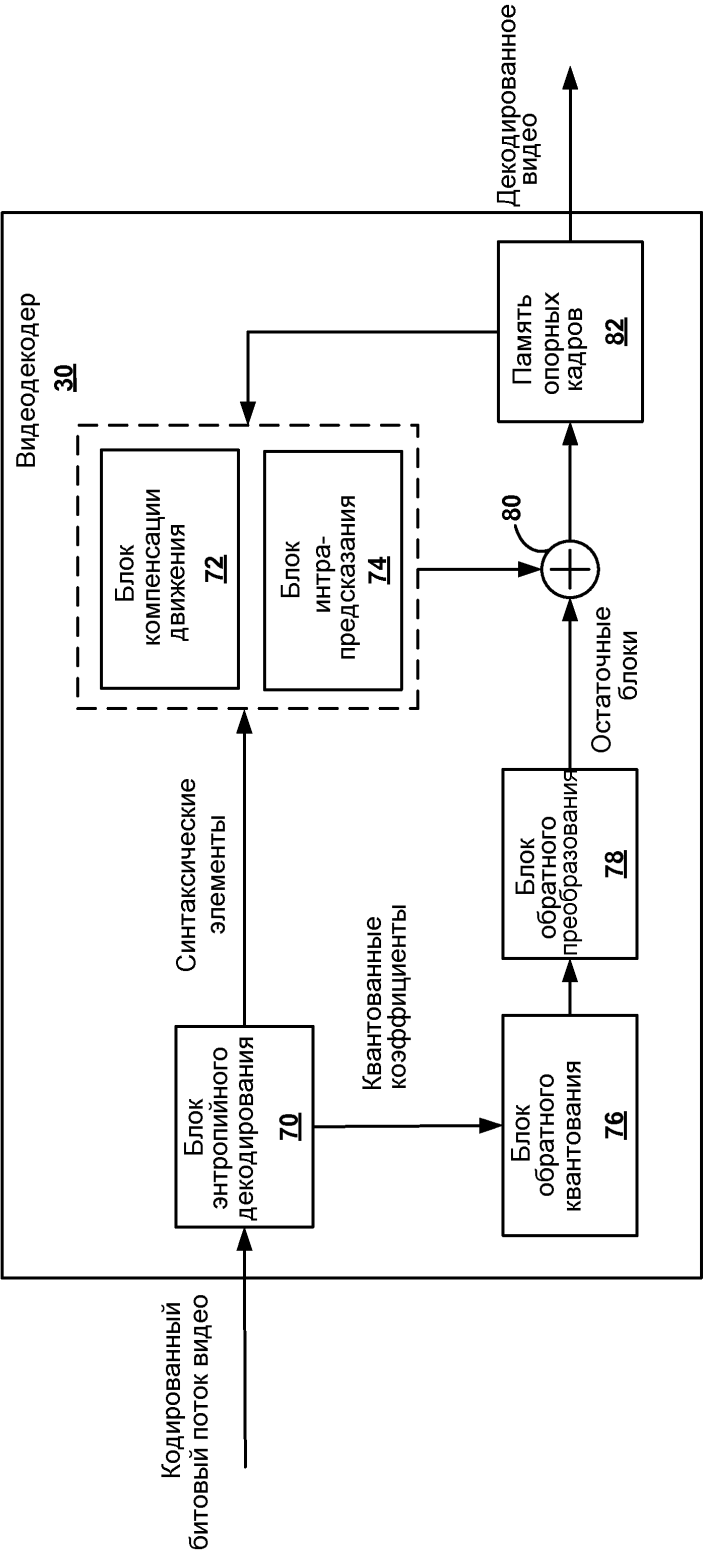
45



Фиг. 1

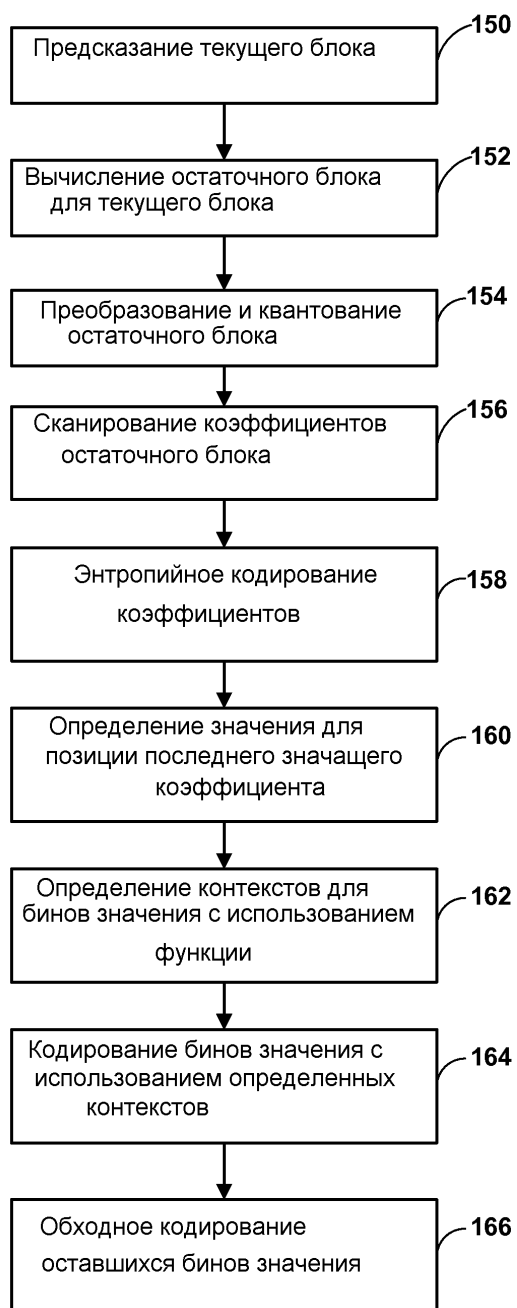


Фиг. 2



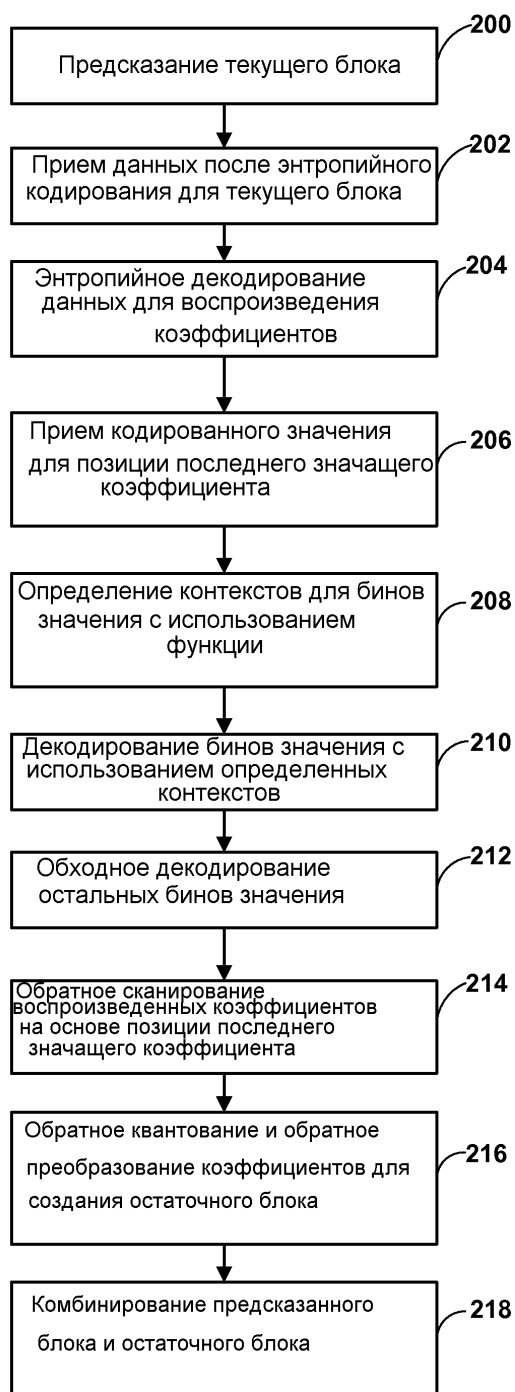
фиг. 3

4 / 5



Фиг. 4

5 / 5



фиг. 5