

(19)



Евразийское
патентное
ведомство

(21) 202192481 (13) A1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ

(43) Дата публикации заявки
2022.01.24(22) Дата подачи заявки
2020.03.18(51) Int. Cl. H04N 19/31 (2014.01)
H04N 19/33 (2014.01)
H04N 19/107 (2014.01)
H04N 19/109 (2014.01)
H04N 19/70 (2014.01)

(54) ВРЕМЕННАЯ ОБРАБОТКА ДЛЯ ТЕХНОЛОГИИ КОДИРОВАНИЯ ВИДЕО

(31) 1903844.7; 1904014.6; 1904492.4;
1905325.5(32) 2019.03.20; 2019.03.23; 2019.03.29;
2019.04.15

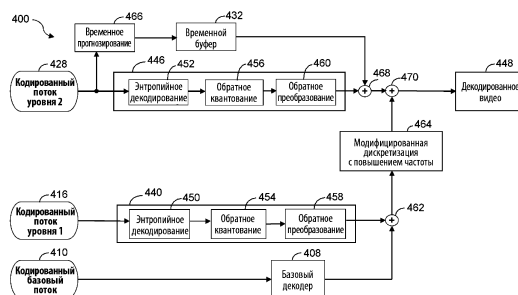
(33) GB

(86) PCT/GB2020/050693

(87) WO 2020/188272 2020.09.24

(71) Заявитель:
В-НОВА ИНТЕРНЭШНЛ ЛТД (GB)(72) Изобретатель:
Меарди Гвидо, Дамнянович Иван
(GB)(74) Представитель:
Нилова М.И. (RU)

(57) Декодер (400), выполненный с возможностью получения первого выходного видео и одного или большего количества дополнительных кодированных потоков (416, 428), декодирования соответствующих кадров одного или большего количества дополнительных кодированных потоков для получения соответствующих наборов остатков и объединения (470) наборов остатков с первым выходным видео, чтобы сгенерировать восстановленное выходное видео (448). Каждый кадр делится на множество мозаичных фрагментов и каждый мозаичный фрагмент делится на множество блоков. Для декодирования соответствующих кадров декодер выполнен с возможностью получения (440, 446) предварительного набора остатков из одного или большего количества дополнительных кодированных потоков, получения набора временных прогнозов с использованием временного буфера (432) и объединения (468) набора временных прогнозов с предварительным набором остатков. Декодер выполнен с возможностью обеспечения обнуления значений набора временных прогнозов на уровне кадра, на уровне мозаичного фрагмента и на уровне блока.



A1

202192481

202192481

A1

ВРЕМЕННАЯ ОБРАБОТКА ДЛЯ ТЕХНОЛОГИИ КОДИРОВАНИЯ ВИДЕО

ОБЛАСТЬ ТЕХНИКИ

Настоящее изобретение относится к способам, аппаратам, компьютерным программам и машиночитаемым носителям для использования в технологии кодирования видео.

УРОВЕНЬ ТЕХНИКИ

Сжатие и восстановление сигналов рассматривается во многих известных системах. Многие типы сигналов, например видео, могут быть сжаты и закодированы для передачи, например, по сети передачи данных. При декодировании такого сигнала может быть желательным повысить уровень качества сигнала и/или извлечь как можно больше информации, содержащейся в исходном сигнале.

Некоторые известные системы используют методы масштабируемого кодирования. Масштабируемое кодирование включает в себя кодирование сигнала вместе с информацией, чтобы обеспечить возможность восстановления сигнала на одном или большем количестве различных уровней качества, например, в зависимости от возможностей декодера и доступной полосы частот.

Есть несколько соображений, касающихся восстановления сигналов в системе масштабируемого кодирования. Одним из таких соображений является способность кодера и/или декодера эффективно обрабатывать информацию. Эффективность, с которой кодер и/или декодер обрабатывает информацию, может быть фактором в уровне производительности кодера и/или декодера.

СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Различные аспекты настоящего изобретения изложены в прилагаемой формуле изобретения.

Дополнительные признаки и преимущества изобретения станут очевидными из следующего описания, приведенного только в качестве примера со ссылкой на прилагаемые графические материалы.

КРАТКОЕ ОПИСАНИЕ ГРАФИЧЕСКИХ МАТЕРИАЛОВ

На Фиг. 1 схематически проиллюстрирован процесс кодирования согласно приведенным в данном документе примерам;

на Фиг. 2 схематически проиллюстрирован процесс декодирования согласно приведенным в данном документе примерам;

на каждой из Фиг. 3А и 3В схематически проиллюстрирован процесс кодирования согласно приведенным в данном документе примерам; и

на каждой из Фиг. 4А и 4В схематически проиллюстрирован процесс декодирования согласно приведенным в данном документе примерам.

ПОДРОБНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

В данном документе описывается гибридная технология кодирования с обратной совместимостью.

Описанные в данном документе примеры предлагают гибкий, адаптируемый, высокоэффективный и недорогой в вычислительном отношении формат кодирования, который сочетает в себе другой формат кодирования видео, базовый кодек (например, AVC, HEVC или любой другой кодек, который существует в настоящее время или появится в будущем) по меньшей мере с двумя уровнями улучшения кодированных данных.

Общая структура схемы кодирования использует сигнал источника с пониженной частотой дискретизации, кодированный с использованием базового кодека, добавляет первый уровень данных коррекции к декодированному выходному сигналу базового кодека для генерации скорректированного изображения, а затем добавляет дополнительный уровень данных улучшения к варианту скорректированного изображения с повышенной частотой дискретизации.

Таким образом, потоки считаются базовым потоком и потоком улучшения. Следует отметить, что обычно ожидается, что базовый поток будет декодироваться аппаратным декодером, а поток улучшения будет подходить для варианта реализации в виде программной обработки с подходящим энергопотреблением.

Такая структура создает множество степеней свободы, которые обеспечивают большую гибкость и адаптируемость ко многим ситуациям, что делает формат кодирования подходящим для многих сценариев использования, включая передачу по технологии OTT, прямую трансляцию, прямую трансляцию сверхвысокой четкости (UHD) и т. д.

Хотя декодированный выходной сигнал базового кодека не предназначен для просмотра, это полностью декодированное видео с более низким разрешением, что делает выходной сигнал совместимым с существующими декодерами и, в случаях, если это

считается подходящим, также может использоваться в качестве выходного сигнала с более низким разрешением.

Формат кодека использует минимальное количество простых инструментов кодирования. При синергетическом сочетании они могут обеспечить улучшение визуального качества по сравнению с изображением с полным разрешением, кодированным с использованием базового кодека, в то же время обеспечивая гибкость в способах их использования.

На Фиг. 1 проиллюстрирован первый пример кодера 100. Проиллюстрированные компоненты также могут быть реализованы как этапы соответствующего процесса кодирования.

В кодере 100 входное видео 102 с полным разрешением обрабатывается для генерации различных кодированных потоков. Первый кодированный поток (кодированный базовый поток 110) создается путем подачи в базовый кодер 106 (например, AVC, HEVC или любой другой кодек) варианта входного видео с пониженной частотой дискретизации, который создается путем понижения 104 частоты дискретизации входного видео 102. Второй кодированный поток (кодированный поток 116 уровня 1) создается путем применения операции 114 кодирования к остаткам, полученным путем нахождения разности 112 между восстановленным видео базового кодека и вариантом входящего видео с пониженной частотой дискретизации. Восстановленное видео с базовым кодеком получают путем декодирования выходного сигнала базового кодера 106 с использованием базового декодера 108. Третий кодированный поток (кодированный поток 128 уровня 2) создается путем обработки 126 остатков, полученных путем нахождения разности 124 между вариантом с повышенной частотой дискретизации скорректированного варианта восстановленного базового кодированного видео и входным видео 102. Скорректированный вариант восстановленного видео с базовым кодеком получают путем объединения 120 восстановленного видео с базовым кодеком и остатков, полученных путем применения операции 118 декодирования к кодированному потоку 116 уровня 1.

Операция 114 кодирования уровня 1 работает с необязательным временным буфером 130 уровня 1, который может использоваться для осуществления временной обработки, как дополнительно описано ниже. Операция кодирования 126 уровня 2 также работает с необязательным временным буфером 132 уровня 2, который может использоваться для осуществления временной обработки, как дополнительно описано ниже. Временной буфер 130 уровня 1 и временной буфер 132 уровня 2 могут работать под управлением компонента 134 временного выбора. Компонент 134 временного выбора

может принимать одно или большее количество из входного видео 102 и выходного сигнала понижения 104 частоты дискретизации для выбора временного режима. Это объясняется более подробно в последующих примерах.

На Фиг. 2 проиллюстрирован первый пример декодера 200. Проиллюстрированные компоненты также могут быть реализованы как этапы соответствующего процесса декодирования. Декодер принимает три потока (кодированный базовый поток 210, кодированный поток 216 уровня 1 и кодированный поток 228 уровня 2), сгенерированные кодером, таким как кодер 100 на Фиг. 1, вместе с заголовками 236, содержащими дополнительную информацию декодирования. Кодированный базовый поток 210 декодируется базовым декодером 208, соответствующим базовому декодеру, используемому в кодере, и его выходной сигнал объединяется 238 с декодированными остатками, полученными посредством декодирования 240 кодированного потока 216 уровня 1. Комбинированное видео подвергается повышению 242 частоты дискретизации и дополнительно комбинируется 244 с декодированными остатками, полученными путем применения операции декодирования 246 к кодированному потоку 228 уровня 2.

На Фиг. 3А и 3В показаны различные варианты второго иллюстративного кодера 300, 380. Второй иллюстративный кодер 300, 380 может содержать реализацию первого иллюстративного кодера 100 на Фиг. 1. В примерах на Фиг. 3А и 3В этапы кодирования потока изображены более подробно, чтобы предоставить пример того, как эти этапы могут быть выполнены. На Фиг. 3А показан первый вариант с временным прогнозированием, обеспечиваемым только на втором уровне процесса улучшения, то есть в отношении кодирования уровня 2. На Фиг. 3В показан второй вариант с временным прогнозированием, выполняемым на обоих уровнях улучшения (т. е. на уровнях 1 и 2).

Базовый поток 310 по существу создается процессом, как объяснено со ссылкой на Фиг. 1. То есть входное видео 302 подвергается понижению 304 частоты дискретизации (то есть операция понижения 304 частоты дискретизации применяется к входному видео 302 для генерации входного видео с пониженной частотой дискретизации). Видео с пониженной частотой дискретизации, полученное путем понижения 304 частоты дискретизации входного видео 302, затем кодируется с использованием первого базового кодера 306 (т. е. к входному видео с пониженной частотой дискретизации применяется операция кодирования, чтобы сгенерировать кодированный базовый поток 310 с использованием первого или базового кодера 306). Предпочтительно первый или базовый кодер 306 является кодеком, подходящим для аппаратного декодирования. Кодированный базовый поток 310 может называться базовым слоем или базовым уровнем.

Как отмечено выше, поток улучшения может содержать два потока. Первый уровень улучшения (описанный в данном документе как «уровень 1») обеспечивает набор данных коррекции, которые могут быть объединены с декодированным вариантом базового потока для генерации скорректированного изображения. Этот первый поток улучшения проиллюстрирован на Фиг. 1 и 3 как кодированный поток 316 уровня 1. Поток улучшения может быть сгенерирован кодером улучшения. Кодер улучшения может отличаться от базового кодера 306, используемого для генерации кодированного базового потока 310.

Чтобы сгенерировать кодированный поток 316 уровня 1, кодированный базовый поток 310 декодируется с использованием базового декодера 308 (то есть операция декодирования применяется к кодированному базовому потоку 310 для генерации декодированного базового потока). Затем определяется разность 312 между декодированным базовым потоком и входным видео с пониженной частотой дискретизации, полученным путем понижения 304 частоты дискретизации входного видео 302 (т. е. операция вычитания 312 применяется к входному видео с пониженной частотой дискретизации и декодированному базовому потоку для генерации первого набора остатков). В данном документе термин «остатки» используется так же, как и термин, известный в данной области техники, то есть как ошибка между опорным кадром и требуемым кадром. В данном документе опорный кадр — это декодированный базовый поток, а требуемый кадр — это входное видео с пониженной частотой дискретизации. Таким образом, остатки, используемые на первом уровне улучшения, можно рассматривать как скорректированное видео, поскольку они «корректируют» декодированный базовый поток до входного видео с пониженной частотой дискретизации, которое использовалось в операции базового кодирования.

Затем разность 312 кодируется для генерации кодированного потока 316 уровня 1 (т. е. операция кодирования применяется к первому набору остатков для генерации первого потока 316 улучшения).

В иллюстративной реализации на Фиг. 3А и 3В операция кодирования включает несколько этапов, каждый из которых является необязательным, предпочтительным и обеспечивает определенные преимущества.

На Фиг. 3 этапы включают: этап преобразования 336, этап квантования 338 и этап энтропийного кодирования 340.

Хотя это не показано в графических материалах, в некоторых примерах процесс кодирования определяет, выбран ли режим ранжирования остатков. Если выбран режим остатков, может выполняться этап ранжирования остатков (то есть операция

ранжирования остатков может выполняться на первом этапе остатков для генерации ранжированного набора остатков). Ранжированный набор остатков может быть отфильтрован так, чтобы не все остатки кодировались в первый поток 316 улучшения (или поток корректировки).

Первый набор остатков либо ранжированный или отфильтрованный первый набор остатков затем преобразуется 336, квантуется 338 и энтропийно кодируется 340 для создания кодированного потока 316 уровня 1 (т. е. операция преобразования 336 применяется к первому набору остатков или отфильтрованному первому набору остатков в зависимости от того, выбран ли режим ранжирования для генерации преобразованного набора остатков; операция квантования 338 применяется к преобразованному набору остатков, чтобы сгенерировать набор квантованных остатков; и операция энтропийного кодирования 340 применяется к квантованному набору остатков для генерации первого уровня потока 316 улучшения). Предпочтительно операция энтропийного кодирования 340 может быть операцией кодирования Хаффмана или операцией кодирования на основе длин серий, или и тем, и другим. Необязательно операция управления (не показана в графических материалах) может применяться к квантованному набору остатков, чтобы скорректировать эффекты операции ранжирования.

Как отмечено выше, поток улучшения может содержать первый уровень 316 улучшения и второй уровень 328 улучшения. Первый уровень 316 улучшения может рассматриваться как скорректированный поток. Второй уровень 328 улучшения может рассматриваться как дополнительный уровень улучшения, который преобразует скорректированный поток в исходное входное видео.

Дополнительный уровень 328 улучшения создается путем кодирования дополнительного набора остатков, которые представляют собой разность 324 между вариантом с повышенной частотой дискретизации декодированного потока уровня 1 и входным видео 302.

На Фиг. 3 квантованный (или управляемый) набор остатков обратно квантуется 342 и обратно преобразуется 344 перед тем, как деблокирующий фильтр (не показан в графических материалах) необязательно применяется для генерации декодированного первого набора остатков (т. е. операция обратного квантования 342 применяется к квантованному первому набору остатков, чтобы сгенерировать деквантованный первый набор остатков; операция обратного преобразования 344 применяется к деквантованному первому набору остатков, чтобы сгенерировать депреобразованный первый набор остатков; и действие деблокирующего фильтра необязательно применяется к депреобразованному первому набору остатков для генерации декодированного первого

набора остатков). Этап деблокирующего фильтра является необязательным в зависимости от примененного преобразования 336 и включает применение взвешенной маски к каждому блоку депреобразованного 344 первого набора остатков.

Декодированный базовый поток объединяется 320 с декодированным первым набором остатков (то есть операция суммирования 320 выполняется для декодированного базового потока и декодированного первого набора остатков с целью генерации воссозданного первого потока). Как проиллюстрировано на Фиг. 3А и 3В, это объединение затем подвергается дискретизации 322 с повышенной частотой (т. е. операция дискретизации 322 с повышенной частотой применяется к воссозданному первому потоку для генерации воссозданного потока с повышенной частотой дискретизации).

Затем поток с повышенной частотой дискретизации сравнивается с входным видео 302, что создает дополнительный набор остатков (то есть операция разности 324 применяется к воссозданному потоку с повышенной частотой дискретизации, чтобы сгенерировать дополнительный набор остатков). Дополнительный набор остатков затем кодируется как кодированный поток 328 улучшения уровня 2 (т. е. операция кодирования затем применяется к дополнительному набору остатков, чтобы сгенерировать кодированный дополнительный поток 328 улучшения).

Как и в случае кодированного потока 316 уровня 1, кодирование, применяемое к остаткам уровня 2, может включать несколько этапов. На Фиг. 3А показаны такие этапы, как временное прогнозирование (описанное дополнительно ниже), преобразование 348, квантование 350 и энтропийное кодирование 352.

Хотя это не показано в графических материалах, в некоторых примерах процесс кодирования определяет, выбран ли режим ранжирования остатков. Если выбран режим остатков, может выполняться этап ранжирования остатков (то есть операция ранжирования остатков может выполняться над дополнительным набором остатков для генерации дополнительного ранжированного набора остатков). Дополнительный ранжированный набор остатков может быть отфильтрован так, чтобы не все остатки кодировались в дополнительный поток 328 улучшения.

Дополнительный набор остатков или дополнительный ранжированный набор остатков затем преобразуется 348 (то есть операция преобразования 348 выполняется с дополнительным ранжированным набором остатков, чтобы сгенерировать дополнительный преобразованный набор остатков). Как проиллюстрировано, операция преобразования 348 может использовать прогнозируемый коэффициент или прогнозируемое среднее значение, полученное из воссозданного первого потока, перед

дискретизацией 322 с повышенной частотой. Дополнительная информация приведена ниже.

На Фиг. 3А показан вариант второго иллюстративного кодера 300, в котором временное прогнозирование выполняется как часть процесса кодирования уровня 2. Временное прогнозирование выполняется с использованием компонента 334 временного выбора и временного буфера 332 уровня 2. Компонент 334 временного выбора может определять режим временной обработки, как более подробно описано ниже, и, соответственно, управлять использованием временного буфера 332 уровня 2. Например, если временная обработка не должна выполняться, компонент 334 временного выбора может указывать, что содержимое временного буфера 332 уровня 2 должно быть установлено на 0. На Фиг. 3В показан вариант второго иллюстративного кодера 380, в котором временное прогнозирование выполняется как часть процесса кодирования как уровня 1, так и уровня 2. На Фиг. 3В временной буфер 330 уровня 1 предоставляется в дополнение к временному буферу 332 уровня 2. Хотя это и не показано, возможны и другие варианты, в которых временная обработка выполняется на уровне 1, но не на уровне 2.

Если выбрано временное прогнозирование, второй иллюстративный кодер 300, 380 на Фиг. 3А или 3В может дополнительно модифицировать коэффициенты (то есть преобразованные остатки на выходе компонента преобразования) путем вычитания соответствующего набора коэффициентов, полученных из соответствующего временного буфера. Соответствующий набор коэффициентов может содержать набор коэффициентов для той же пространственной области (например, той же единицы кодирования, находящейся в кадре), которые получены из предыдущего кадра (например, коэффициенты для той же области предыдущего кадра). Эти коэффициенты могут быть выведены или иным образом получены из временного буфера. Коэффициенты, полученные из временного буфера, могут называться в данном документе временными коэффициентами. Вычитание может применяться с помощью компонента вычитания, такого как третьи компоненты 354 и 356 вычитания (для соответствующих уровней 2 и 1). Этот этап временного прогнозирования будет дополнительно описан применительно к последующим примерам. Таким образом, в случае применения временного прогнозирования кодированные коэффициенты соответствуют разности между определенным кадром и другим кадром потока. Другой кадр может быть более ранним или более поздним кадром (или блоком в кадре) в потоке. Таким образом, вместо кодирования остатков между воссозданным потоком с повышенной частотой дискретизации и входным видео процесс кодирования может кодировать разность между

преобразованным кадром в потоке и преобразованными остатками кадра. Таким образом может быть уменьшена энтропия. Временное прогнозирование может применяться выборочно для групп единиц кодирования (называемых в данном документе «мозаичными фрагментами») на основе управляющей информации, а применение временного прогнозирования в декодере может применяться путем отправки дополнительной управляющей информации вместе с кодированными потоками (например, в заголовках).

Как показано на Фиг. 3А и 3В, если временное прогнозирование активировано, каждый преобразованный коэффициент может быть следующим:

$$\Delta = F_{current} - F_{buffer}$$

где временной буфер может хранить данные, связанные с предыдущим кадром. Временное прогнозирование может выполняться для одной цветовой плоскости или для нескольких цветовых плоскостей. В целом вычитание может применяться как поэлементное вычитание для «кадра» видео, где элементы кадра представляют преобразованные коэффициенты, причем преобразование применяется в отношении конкретного размера единицы кодирования n на n (например, 2×2 или 4×4). Разность, полученная в результате временного прогнозирования (например, приведенная выше дельта), может быть сохранена в буфере для использования в следующем кадре. Следовательно, в действительности остаток, который приводит к временному прогнозированию, является остатком коэффициента по отношению к буферу. Хотя на Фиг. 3А и 3В показано временное прогнозирование, выполняемое после операции преобразования, оно также может выполняться после операции квантования. Это может устранить необходимость в применении компонента 358 обратного квантования уровня 2 и/или компонента 360 обратного квантования уровня 1. Таким образом, как проиллюстрировано на Фиг. 3А и 3В и описано выше, выходом кодеров 300, 380 второго примера после выполнения процесса кодирования является кодированный базовый поток 310 и один или большее количество потоков улучшения, которые предпочтительно содержат кодированный поток 316 уровня 1 для первого уровня улучшения и кодированный поток 328 уровня 2 для дополнительного или второго уровня улучшения.

Фиг. 4А и 4В иллюстрируют соответствующие варианты второго иллюстративного декодера 400, 480. Варианты второго иллюстративного декодера 400, 480 могут быть соответственно реализованы, чтобы соответствовать первому иллюстративному декодеру 200 на Фиг. 2. Явно идентифицируемые этапы и компоненты декодирования показаны более подробно, чтобы предоставить пример того, как может выполняться декодирование. Как и на Фиг. 3А и 3В, на Фиг. 4А проиллюстрирован вариант, в котором временное

прогнозирование используется только для второго уровня (т. е. уровня 2), а на Фиг. 4В проиллюстрирован вариант, в котором временное прогнозирование используется на обоих уровнях (т. е. на уровнях 1 и 2). Как и раньше, предусмотрены дополнительные варианты (например, уровень 1, но не уровень 2), причем формой конфигурации можно управлять с использованием информации сигнализации.

Как показано в примере на Фиг. 4В, в процессе декодирования декодер 480 может анализировать заголовки 436 (например, содержащие данные глобальной конфигурации, данные конфигурации изображения и другие блоки данных) и конфигурировать декодер на основе этих заголовков 436. Чтобы воссоздать входное видео, декодер 400, 480 может декодировать каждый из базового потока 410, первого потока 416 улучшения и дополнительного потока 428 улучшения. Кадры потока могут быть синхронизированы, а затем объединены для получения декодированного видео 448.

В каждом процессе декодирования потоки улучшения могут подвергаться этапам энтропийного декодирования 450, 452, обратного квантования 454, 456 и обратного преобразования 458, 460 для воссоздания набора остатков.

Процессы декодирования согласно Фиг. 4А и 4В включают извлечение массива энтропийно декодированных квантованных коэффициентов, представляющих первый уровень улучшения, и подачу на выход массива остатков L-1 (уровня 1). Энтропийно декодированные квантованные коэффициенты в этом случае получают путем применения операции энтропийного декодирования 450 к кодированному потоку 416 L-1. Процессы декодирования согласно Фиг. 4А и 4В дополнительно содержат извлечение массива выборок выходного сигнала базового декодера 408. Процессы декодирования на Фиг. 4А и 4В дополнительно включают применение процесса деквантования 454 к массиву энтропийно декодированных квантованных коэффициентов для получения набора деквантованных коэффициентов, применение процесса преобразования 458 к набору деквантованных коэффициентов и, необязательно, применение процесса фильтрации (не показан на Фиг. 4А и 4В) для подачи на выход массива остатков L-1, представляющего первый уровень улучшения, который может называться предварительным набором остатков. В этом случае процесс деквантования 454 применяется к энтропийно декодированным квантованным коэффициентам для соответствующих блоков кадра кодированного потока 416 уровня 1, а процесс преобразования 458 (который может называться операцией обратного преобразования) применяется к выходному сигналу процесса деквантования 454 для соответствующих блоков кадра. Кроме того, процессы декодирования по Фиг. 4А и 4В дополнительно включают воссоздание изображения путем объединения 462 массива остатков L-1 с массивом выборок выходного сигнала

базового декодера 408. Процессы декодирования по Фиг. 4А и 4В включают применение процесса преобразования 458 из набора предварительно определенных процессов преобразования в соответствии с сигнализируемым параметром. Например, процесс преобразования 458 может применяться к единице кодирования 2x2 или единице кодирования 4x4. Единица кодирования может называться в данном документе как блок элементов в массиве, в данном случае массиве остатков L-1.

Процессы декодирования по Фиг. 4А и 4В содержат извлечение массива энтропийно декодированных квантованных коэффициентов, представляющих дополнительный уровень улучшения, и подачу на выход массива остатков. В процессах декодирования, проиллюстрированных на Фиг. 4А и 4В, дополнительный уровень улучшения является вторым уровнем улучшения, а выходной массив остатков представляет собой массив остатков L-2. Способ по Фиг. 4А и 4В дополнительно включает извлечение массива остатков L-1 первого уровня улучшения, соответствующего массиву энтропийно декодированных квантованных коэффициентов, представляющих дополнительный уровень улучшения. Способ по Фиг. 4А и 4В дополнительно включает применение процесса дискретизации 464 с повышенной частотой к массиву остатков первого уровня улучшения. На Фиг. 4А и 4В процесс дискретизации 464 с повышенной частотой применяется к комбинации массива остатков L-1 первого уровня улучшения и соответствующего массива отсчетов выходного сигнала базового декодера 408.

На Фиг. 4А и 4В процесс дискретизации 464 с повышенной частотой представляет собой модифицированный процесс дискретизации с повышенной частотой, в котором модификатор добавляется к остатку. Этап добавления модификатора может выполняться как часть процесса преобразования 460. В альтернативном варианте реализации изобретения, поскольку процесс преобразования 460 включает линейное преобразование, этап добавления модификатора может выполняться как часть модифицированного процесса дискретизации 464 с повышенной частотой, как показано на Фиг. 4А и 4В. Таким образом, этап добавления модификатора приводит к модификации остатка. Модификация может выполняться на основе местоположения остатка в кадре. Модификация может иметь предварительно определенное значение.

На Фиг. 4А временное прогнозирование применяется во время декодирования уровня 2. В примере на Фиг. 4А временное прогнозирование управляется компонентом временного прогнозирования 466. В этом варианте управляющая информация для временного прогнозирования извлекается из кодированного потока 428 уровня 2, как указано стрелкой от потока к компоненту временного прогнозирования 466. В других вариантах реализации изобретения, таких как показанные на Фиг. 4В, управляющая

информация для временного прогнозирования может направляться отдельно от кодированного потока 428 уровня 2, например, в заголовках 436. Компонент временного прогнозирования 466 управляет использованием временного буфера 432 уровня 2, например, он может определять временной режим и управлять временным обновлением, как описано со ссылкой на приведенные далее примеры. Содержимое временного буфера 432 может обновляться на основе данных для предыдущего кадра остатков. При применении временного буфера 432 содержимое буфера добавляется 468 ко второму набору остатков. На Фиг. 4А содержимое временного буфера 432 добавляется 468 к выходному сигналу компонента 446 декодирования уровня 2 (который на Фиг. 4А реализует энтропийное декодирование 452, обратное квантование 456 и обратное преобразование 460). В других примерах содержимое временного буфера может представлять любой набор промежуточных данных декодирования, и поэтому добавление 468 может быть перемещено соответствующим образом, чтобы применить содержимое временного буфера на соответствующем этапе (например, если временной буфер применяется на этапе деквантованных коэффициентов, добавление 468 может быть расположено перед обратным преобразованием 460). Затем скорректированный по времени второй набор остатков объединяется 470 с выходным сигналом дискретизации 464 с повышенной частотой для генерации декодированного видео 448. Декодированное видео 448 имеет пространственное разрешение уровня 2, которое может быть выше пространственного разрешения уровня 1. Второй набор остатков применяет коррекцию к (просматриваемому) восстановленному видео с повышенной частотой дискретизации, причем коррекция добавляет обратно мелкие детали и улучшает резкость линий и очертаний.

Процессы преобразования 458, 460 могут быть выбраны из набора предварительно определенных процессов преобразования согласно сигнализируемому параметру. Например, процесс преобразования 460 может быть применен к блоку 2x2 элементов в массиве остатков L-2 или к блоку 4x4 элементов в массиве остатков L-2.

На Фиг. 4В показан вариант второго иллюстративного декодера 480. В этом случае данные управления временным прогнозированием принимаются компонентом временного прогнозирования 466 из заголовков 436. Компонент временного прогнозирования 466 управляет временным прогнозированием как на уровне 1, так и на уровне 2, но в других примерах при желании для обоих уровней могут быть предусмотрены отдельные компоненты управления. На Фиг. 4В показано, как восстановленный второй набор остатков, который добавлен 468 к выходному сигналу компонента 446 декодирования уровня 2, может быть возвращен для сохранения во временном буфере 432 уровня 2 для

следующего кадра (обратная связь опущена на Фиг. 4А. для ясности). Также показан временной буфер 430 уровня 1, который функционирует аналогично описанному выше временному буферу 432 уровня 2, и на этом чертеже показан контур обратной связи для буфера. Содержимое временного буфера 430 уровня 1 добавляется в конвейер обработки остатков уровня 1 посредством суммирования 472. Опять же, положение указанного суммирования 472 может изменяться вдоль конвейера обработки остатков уровня 1 в зависимости от того, где применяется временное прогнозирование (например, если оно применяется в преобразованном пространстве коэффициентов, оно может быть расположено перед компонентом обратного преобразования 458 уровня 1).

На Фиг. 4В показаны два способа, которыми временная управляющая информация может быть сигнализирована декодеру. Первый способ — посредством заголовков 436, как описано выше. Второй способ, который можно использовать в качестве альтернативного или дополнительного пути сигнализации, — это данные, закодированные в самих остатках. На Фиг. 4В показан случай, когда данные 474 могут быть закодированы в НН-преобразованных коэффициентах и, таким образом, могут быть извлечены после энтропийного декодирования 452. Эти данные 474 могут быть извлечены из конвейера обработки остатков уровня 2 и переданы компоненту временного прогнозирования 466.

Каждый поток улучшения или оба потока улучшения могут быть инкапсулированы в один или большее количество битовых потоков улучшения с использованием набора единиц уровня абстракции сети (NALU). NALU предназначены для инкапсуляции битового потока улучшения, чтобы применить улучшение к правильному базовому восстановленному кадру. NALU может, например, содержать ссылочный индекс для NALU, содержащей битовый поток восстановленного кадра базового декодера, к которому должно быть применено улучшение. Таким образом, улучшение может быть синхронизировано с базовым потоком, а кадры каждого битового потока объединены для создания декодированного выходного видео (то есть остатки каждого кадра уровня улучшения объединяются с кадром базового декодированного потока). Группа изображений может представлять несколько NALU.

Каждый кадр может состоять из трех разных плоскостей, представляющих разные цветовые компоненты, например, каждый компонент трехканального видео YUV может иметь разные плоскости. Каждая плоскость может затем иметь остаточные данные, которые относятся к заданному уровню улучшения, например, плоскость Y может иметь набор остаточных данных уровня 1 и набор остаточных данных уровня 2. В некоторых случаях, например, для монохромных сигналов, может быть только одна плоскость; в этом случае термины «кадр» и «плоскость» могут использоваться как взаимозаменяемые.

Остаточные данные уровня 1 и остаточные данные уровня 2 могут быть разделены следующим образом. Остаточные данные делятся на блоки, размер которых зависит от размера используемого преобразования. Блоки представляют собой, например, блок элементов 2×2 , если используется преобразование направленной декомпозиции 2×2 , или блок элементов 4×4 , если используется преобразование направленной декомпозиции 4×4 . Мозаичный фрагмент — это группа блоков, которые покрывают область кадра (например, область M на N , которая может быть квадратной областью). Мозаичный фрагмент представляет собой, например, мозаичный фрагмент 32×32 элементов. По существу, каждый кадр в кодированном потоке может быть разделен на множество мозаичных фрагментов, и каждый мозаичный фрагмент из множества мозаичных фрагментов может быть разделен на множество блоков. В случае цветного видео каждый кадр может быть разделен на множество плоскостей, при этом каждая плоскость разделена на множество мозаичных фрагментов, и каждый мозаичный фрагмент из множества мозаичных фрагментов разделен на множество блоков.

Выше было отмечено, как набор компонентов или инструментов обработки может применяться к каждому из потоков улучшения (или входному видео 102, 302) на протяжении всего процесса. Ниже приводится краткое описание каждого из инструментов и их функций в рамках общего процесса, как показано на Фиг. 1–4.

Процесс понижения 104, 304 частоты дискретизации применяется к входному видео 102, 302 для создания видео с пониженной частотой дискретизации, которое должно кодироваться базовым кодером 106, 306. Дискретизация 104, 304 с пониженной частотой может выполняться как в вертикальном, так и в горизонтальном направлениях, или, в альтернативном варианте реализации изобретения, только в горизонтальном направлении.

Входной сигнал для операции кодирования 114 L-1 содержит остатки L-1, полученные путем нахождения разности 112, 312 между декодированным выходным сигналом базового декодера 108, 308 и видео с пониженной частотой дискретизации, полученным путем понижения 104, 304 частоты дискретизации входного видео 102, 302. Затем остатки L-1 преобразуются 336, квантуются 338 и кодируются 340, как дополнительно описано ниже. Преобразование 336 подает на выход коэффициенты преобразования (то есть преобразованные остатки L-1).

В процессе преобразования 336 могут использоваться два типа преобразований. Оба используют небольшие ядра, которые применяются непосредственно к остаткам, остающимся после этапа применения прогнозируемого среднего.

Первое преобразование имеет ядро 2×2 , которое применяется к блоку остатков 2×2 . Полученные коэффициенты следующие:

$$\begin{pmatrix} C_{00} \\ C_{01} \\ C_{10} \\ C_{11} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} R_{00} \\ R_{01} \\ R_{10} \\ R_{11} \end{pmatrix}$$

Второе преобразование имеет ядро 4x4, которое применяется к блоку остатков 4x4. Полученные коэффициенты следующие:

$$\begin{pmatrix} C_{00} \\ C_{01} \\ C_{02} \\ C_{03} \\ C_{10} \\ C_{11} \\ C_{12} \\ C_{13} \\ C_{20} \\ C_{21} \\ C_{22} \\ C_{23} \\ C_{30} \\ C_{31} \\ C_{32} \\ C_{33} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} R_{00} \\ R_{01} \\ R_{02} \\ R_{03} \\ R_{10} \\ R_{11} \\ R_{12} \\ R_{13} \\ R_{20} \\ R_{21} \\ R_{22} \\ R_{23} \\ R_{30} \\ R_{31} \\ R_{32} \\ R_{33} \end{pmatrix}$$

Соответствующим образом адаптированные преобразования также могут применяться, если дискретизация с пониженной и/или повышенной частотой выполняется только в горизонтальном направлении (например, с некоторыми элементами, установленными на 0). Если используется преобразование Адамара, например, как проиллюстрировано в приведенных выше примерах матриц, тогда при декодировании или

обратном преобразовании может использоваться та же матрица, например, матрицы Адамара являются их собственными обратными. В этом случае, например, массив $(M \times N) \times 1$ остатков R , относящийся к блоку $M \times N$, может быть получен в декодере из соответствующего $(M \times N) \times 1$ массива коэффициентов C , используя $R = H * C$, где H равно одной из матрицы Адамара, показанных выше.

Затем коэффициенты квантуются 338 с использованием линейного квантователя. Линейный квантователь может использовать мертвую зону переменного размера. Линейный квантователь может использовать мертвую зону другого размера по сравнению с этапом квантования и нецентрированным смещением деквантования.

Квантованные коэффициенты кодируются с использованием энтропийного кодера 340. Есть две схемы энтропийного кодирования 340. В первой схеме квантованные коэффициенты кодируются с использованием кодера длины серии (RLE). Во второй схеме квантованные коэффициенты сначала кодируются с использованием RLE, а затем кодированный выходной сигнал обрабатывается с использованием кодера Хаффмана. Это может выгодно кодировать длинные потоки нулей, которые обычно обнаруживаются с преобразованными остатками, с помощью RLE, а затем дополнительно выгодно кодировать различные частоты квантованных значений (например, которые часто имеют уменьшенное число по мере увеличения значения из-за распределений остатков и их линейно преобразованных значений) с использованием кодера Хаффмана.

Если был выбран остаточный режим (RM), остатки L-1 дополнительно ранжируются и выбираются, чтобы определить, какие остатки должны быть преобразованы 336 и кодированы. Предпочтительно это выполняется до энтропийного кодирования 340.

Если для кодирования L-1 выбран режим временного выбора, кодер дополнительно изменит коэффициенты путем вычитания соответствующих коэффициентов, полученных из временного буфера 130, 330 уровня 1, то есть временного прогнозирования, описанного ниже.

Входной сигнал операции декодирования 118 L-1 содержит кодированные остатки L-1, которые проходят через энтропийный декодер 450, деквантователь 454 и модуль обратного преобразования 458. Операции, выполняемые этими модулями, являются обратными операциями, выполняемыми модулями, описанными выше.

Если для кодирования L-1 был выбран режим временного выбора, остатки могут быть частично прогнозированы из совмещенных остатков из временного буфера 130, 330 уровня 1. Совмещенные остатки могут называться в данном документе временными прогнозами.

Если используется преобразование 4x4, декодированные остатки могут подаваться в модуль деблокирующего фильтра. Деблокирующий фильтр работает с каждым блоком преобразованных остатков, применяя маску, веса которой могут быть указаны. Общая структура маски следующая:

$$\begin{array}{cccc} \alpha & \beta & \beta & \alpha \\ \beta & 1 & 1 & \beta \\ \beta & 1 & 1 & \beta \\ \alpha & \beta & \beta & \alpha \end{array}$$

где $0 \leq \alpha \leq 1$ и $0 \leq \beta \leq 1$.

Выходной сигнал от объединения 120, 320 декодированных (и деблокированных, если применимо) остатков L-1 и базового декодированного видео подвергается дискретизации с повышенной частотой 122, 322 для генерации восстановленного видео с повышенной частотой дискретизации. Дискретизация с повышенной частотой может быть выбрана и сигнализирована в байтовом потоке.

Входной сигнал для операции кодирования 126 L-2 содержат остатки L-2, полученные путем определения разности 124, 324 между восстановленным видео с повышенной частотой дискретизации и входным видео 102, 302. Затем остатки L-2 преобразуются 348, квантуются 350 и кодируются 352, как дополнительно описано ниже. Преобразование 348, квантование 350 и кодирование 352 выполняются таким же образом, как описано в отношении кодирования 114 L-1. Как объяснено со ссылкой на кодирование 114 L-1, преобразование 348 подает на выход коэффициенты преобразования (то есть преобразованные остатки L-2). Если был выбран RM, остатки L-2 дополнительно ранжируются и выбираются, чтобы определить, какие остатки следует преобразовать и кодировать. Операция кодирования 126 L-2 может дополнительно включать два дополнительных процесса, как описано ниже.

Если выбран режим прогнозируемых коэффициентов, кодер дополнительно модифицирует преобразованный коэффициент C00 (например, «среднее» или коэффициент «A» для преобразования 2x2). Если используется преобразование 2x2, C00 будет модифицирован путем вычитания значения остатка с повышенной частотой дискретизации, из которого прогнозируется преобразованный блок остатков. Если используется преобразование 4x4, C00 будет модифицирован путем вычитания среднего значения четырех остатков с повышенной частотой дискретизации, из которого прогнозируется преобразованный блок остатков.

Если для кодирования L-2 выбран режим временного выбора, кодер дополнительно модифицирует коэффициенты путем вычитания соответствующих коэффициентов, полученных из временного буфера 132, 332 уровня 2, как описано выше.

Входной сигнал для операции декодирования 246, 446 L-2 содержат кодированные остатки L-2. Процесс декодирования остатков L-2 проходит через энтропийный декодер 452, деквантователь 456 и модуль 460 обратного преобразования. Операции, выполняемые этими модулями, являются обратными операциями, выполняемыми модулями, описанными выше. Если для кодирования L-2 был выбран режим временного выбора, остатки могут быть частично прогнозированы из совмещенных остатков из временного буфера 132, 332 уровня 2. Совмещенные остатки могут называться в данном документе временными прогнозами.

Модифицированный процесс дискретизации 242, 464 с повышенной частотой включает два этапа, причем второй зависит от сигнала, полученного декодером. На первом этапе комбинация 238, 462 декодированных (и деблокированных, если применимо) остатков L-1 и базового декодированного видео 208, 408 (восстановленное видео L-1) подвергается дискретизации с повышенной частотой для генерации восстановленного видео с повышенной частотой дискретизации. Если был выбран режим прогнозируемых коэффициентов, то выполняется второй этап. В частности, значение элемента в восстановленном значении L-1, из которого был получен блок 2x2 в восстановленном видео с повышенной частотой дискретизации, добавляется к упомянутому блоку 2x2 в восстановленном видео с повышенной частотой дискретизации.

На протяжении всего вышеизложенного термин «битовый поток» может быть заменен потоком, байтовым потоком или потоком NALU, в зависимости от ситуации.

Как показано на Фиг. 4А и 4В, следующий пример относится к процессу временного прогнозирования, применяемому во время декодирования уровня 2. Однако следует принимать во внимание, что следующий процесс временного прогнозирования может дополнительно или в альтернативном варианте реализации изобретения применяться во время декодирования уровня 1.

В этом примере декодер 400, 480 выполнен с возможностью получения параметра *temporal_enabled*, который указывает, следует ли использовать временное прогнозирование при декодировании изображения. Параметр *temporal_enabled* может упоминаться в данном документе как первый параметр с первым значением, которое указывает, что временная обработка разрешена. В этом случае параметр *temporal_enabled* указывает, должен ли декодер 400, 480 обновлять значения временных прогнозов, полученных из содержимого временного буфера 432. Значение параметра *temporal_enabled* может иметь длину в один бит. В этом примере значение 1 указывает, что временное прогнозирование будет использоваться при декодировании изображения, а значение 0 указывает, что временное прогнозирование не будет использоваться при

декодировании изображения. Параметр *temporal_enabled* может быть принят один раз для группы изображений, связанных с кодированными потоками, описанными выше, причем группа изображений представляет собой набор последовательных изображений в кодированном видеопотоке.

В этом примере декодер 400, 480 выполнен с возможностью получения параметра *temporal_refresh_bit*, который указывает, следует ли обновлять временной буфер 432 для кадра. Если кадр содержит несколько плоскостей, обновление может применяться для всех плоскостей в кадре (то есть для кадра, который содержит плоскости). Обновление временного буфера 432 может включать установку значений из временного буфера 432 на ноль. Таким образом, когда содержимое временного буфера 432 добавляется 468 ко второму набору остатков, второй набор остатков не изменяется таким же образом, как если бы временной буфер 432 не применялся. Значение параметра *temporal_refresh_bit* может иметь длину в один бит. В этом примере значение 1 указывает, что временной буфер 432 должен обновляться для кадра, а значение 0 указывает, что временной буфер 432 не должен обновляться для кадра. Параметр *temporal_refresh_bit* может быть получен один раз для каждого изображения в кодированном видеопотоке. Параметры *temporal_enabled* и *temporal_refresh_bit* могут содержаться во временной управляющей информации, передаваемой декодеру, например, через заголовки 436, как описано выше.

В этом примере, если переменная *temporal_enabled* равна 1 и *temporal_refresh_bit* равна 0, процесс временного прогнозирования запускается, как указано ниже.

Входными данными для вызванного процесса временного прогнозирования являются:

- местоположение ($xTbP$, $yTbP$), определяющее верхнюю левую выборку текущего блока преобразования яркости или цветности относительно верхней левой выборки яркости или цветности текущего изображения. ($xTbP$, $yTbP$) могут быть связаны либо с плоскостью яркости, либо с плоскостью цветности, в зависимости от того, к какой плоскости принадлежат коэффициенты преобразования.
- параметр $nTbS$, указывающий размер текущего блока преобразования. Например, $nTbS$ равно 2, если в описанном выше процессе преобразования должно использоваться преобразование направленной декомпозиции 2x2, и $nTbS$ равно 4, если должен использоваться процесс преобразования направленной декомпозиции 4x4.
- параметр *temporal_tile_intra_signalling_enabled* определяющий, следует ли использовать временное прогнозирование мозаичного фрагмента при декодировании мозаичного фрагмента элементов. Параметр *temporal_tile_intra_signalling_enabled* может называться в данном документе

третьим параметром с третьим значением. Значение параметра *temporal_tile_intra_signalling_enabled* может иметь длину в один бит. В этом примере, если значение параметра *temporal_tile_intra_signalling_enabled* равно 1, будет разрешен процесс временного прогнозирования мозаичного фрагмента. Параметр *temporal_tile_intra_signalling_enabled* может быть принят один раз для группы изображений, связанных с кодированными потоками, описанными выше. Параметр *temporal_tile_intra_signalling_enabled* может содержаться во временной управляющей информации, передаваемой декодеру, например, через заголовки 436. Этот параметр может указывать, может ли предоставляться временная сигнализация, относящаяся к мозаичному фрагменту, если сигнализация инкапсулирована в значениях коэффициентов, как описано ниже (то есть является «внутренней» или находящейся внутри данных). Если этот параметр установлен равным 1, то первый блок в мозаичном фрагменте может нести сигнализацию временного обновления для мозаичного фрагмента (например, посредством значения коэффициента в первом блоке).

- массив *TransCoeffQ* размера $(nTbS) \times (nTbS)$, содержащий блок энтропийно декодированных квантованных коэффициентов с элементами *TransCoeffQ*[x][y].

Выходными данными этого процесса являются массив $(nTbS) \times (nTbS)$ модифицированных коэффициентов *TransCoeffQ* и массив $(nTbS) \times (nTbS)$ *tempPredSamples* с элементами *tempPredSamples*[x][y]. Массив *tempPredSamples* представляет набор временных прогнозов, полученных с использованием временного буфера 432.

В этом примере применяются следующие упорядоченные этапы:

1. Если параметр *temporal_tile_intra_signalling_enabled* равен 1, процесс временного прогнозирования мозаичного фрагмента будет использоваться при декодировании мозаичного фрагмента элементов. В процессе временного прогнозирования мозаичного фрагмента, если $xTbP \gg 5$ равно 0, $yTbP \gg 5$ равно 0 (где « $x \gg y$ » является арифметическим сдвигом вправо целочисленного представления x с двоичным дополнением на y двоичных цифр) и *TransCoeffQ*[nTbs-1][0]&0x1 равно 1 (где «&» представляет побитовый оператор «и»), процесс мозаичного временного обновления вызывается с указанием местоположения ($xTbP$, $yTbP$) в качестве входного сигнала. Сдвиг вправо позволяет пропускать элементы для следующих мозаичных фрагментов, например, мозаичный фрагмент 32x32 имеет длину и ширину 2^5 , в этом случае сдвиг на 5 бит, равный нулю, указывает, что текущее местоположение кратно 32 и, таким образом, относится к первому блоку мозаичного фрагмента. В этом случае первый блок в мозаичном фрагменте используется для сигнализации временного обновления мозаичного

фрагмента. Результатом процесса мозаичного временного обновления является то, что содержимое *temporalBuffer* для мозаичного фрагмента в местоположении (xTbP, yTbP) устанавливается на ноль. В этом примере *temporalBuffer* представляет собой временной буфер 432, в котором хранятся данные, связанные со вторым кадром, который не является текущим кадром, из которого могут быть получены значения временных прогнозов. Таким образом, второй кадр может быть более ранним или более поздним кадром в потоке. $TransCoeffQ[nTbS-1][0] \& 0x1$ является параметром для текущего блока преобразования со значением, указывающим, должны ли обновляться значения набора временных прогнозов, полученные из временного буфера 432 для соответствующего блока. В данном случае параметр «переносится» в пределах значений коэффициентов преобразования. Операция «&» (т. е. логическое «и») со значением бита 1 (например, $0x1$) просто преобразует любое ненулевое значение в выходное значение бита 1 (а любое нулевое значение остается нулевым). В целом параметр $TransCoeffQ[nTbS-1][0] \& 0x1$ может упоминаться в данном документе как второй параметр со вторым значением, который обеспечивает временную сигнализацию, которая в этом случае указывает, должны ли обновляться значения во временном буфере 432, ассоциированном с данным мозаичным фрагментом. В этом случае параметр используется для указания обновления мозаичного фрагмента; в приведенном ниже примере он используется для обозначения обновления блока. Обновление может включать установку значений соответствующего мозаичного фрагмента или блока на ноль. В этом примере значение параметра $TransCoeffQ[nTbS-1][0] \& 0x1$ имеет длину в один бит, и значение 1 указывает, что значения во временном буфере 432 должны быть обновлены. В процессе временного прогнозирования мозаичного фрагмента процесс мозаичного временного обновления выполняется для мозаичного фрагмента временного буфера 432 в ответ на получение параметра $TransCoeffQ[nTbS-1][0] \& 0x1$ для блока внутри мозаичного фрагмента. Другими словами, декодер выполнен с возможностью обновления временного буфера 432 для мозаичного фрагмента в ответ на получение параметра $TransCoeffQ[nTbS-1][0] \& 0x1$ (второй параметр) для указанного блока в мозаичном фрагменте. Таким образом, об обновлении мозаичного фрагмента можно сигнализировать с помощью параметра в одном блоке в мозаичном фрагменте без необходимости отдельно сигнализировать об обновлении для каждого другого блока в мозаичном фрагменте. Следовательно, в этом примере декодер выполнен с возможностью обновления значений временных прогнозов мозаичного фрагмента в ответ на получение как второго параметра ($TransCoeffQ[nTbS-1][0] \& 0x1$) со вторым значением, которое в приведенном ниже случае 2) указывает, что декодер должен обновить значения временных прогнозов для блока, так и третьего

параметра (*temporal_tile_intra_signalling_enabled*) с третьим значением, указывающим, что получение второго параметра для указанного блока со вторым значением указывает, что значения во временном буфере 432, связанном с мозаичным фрагментом, должны быть обновлены. Параметр *TransCoeffQ[nTbS-1][0]&0x1* является примером временной управляющей информации, которая может быть сигнализирована в декодер посредством данных, кодированных внутри самих кодированных остатков, так что эта временная управляющая информация может быть извлечена после энтропийного декодирования 452. Например, этот параметр (второй параметр) может быть получен из одного из набора коэффициентов для блока.

2. В случае, когда процесс временного прогнозирования мозаичного фрагмента не происходит, если *TransCoeffQ[nTbS-1][0]&0x1* равен 0, то *tempPredSamples[x][y] = temporalBuffer[xTbP+x][yTbP+y]*, где *x* и *y* находятся в диапазоне $[0, nTbS-1]$. В противном случае все *tempPredSamples[x][y]* установлены на 0. Таким образом, для каждого соответствующего блока значения временных прогнозов для соответствующего блока, полученные из временного буфера 432, обновляются в ответ на получение второго параметра (параметра *TransCoeffQ[nTbS-1][0]&0x1*) из данных из одного или большего количества кодированных потоков (в данном случае кодированного потока L-2) для соответствующего блока. В ответ на определение того, что второй параметр для соответствующего блока имеет второе значение, которое обеспечивает временную сигнализацию для соответствующего блока, значения временных коэффициентов для соответствующего блока обновляются. В этом случае обновление включает установку значений временных прогнозов для соответствующего блока на ноль. В этом случае обновление набора значений временных прогнозов для каждого блока указывается индивидуально для каждого блока. В противном случае, если *TransCoeffQ[nTbS-1][0]&0x1* равен 0, временной буфер 432 не обновляется, и значения набора временных прогнозов для соответствующего блока извлекаются из временного буфера 432 для соответствующего блока.

3. Значение *TransCoeffQ[nTbs-1][0]* установлено равным *TransCoeffQ[nTbs-1][0] >> 1*. Если *TransCoeffQ[nTbs-1][0]* имеет двоичные значения 0 или 1, эта операция устанавливает значение коэффициента преобразования в 0 для последующих операций (например, более позднего обратного преобразования). Это означает, что значения 1, которые используются для сигнализации, не преобразуются в артефакты изображения в блоке (значение 0 остатка просто равнозначно отсутствию изменений).

Массив *tempPredSamples* размера $(nTbS) \times (nTbS)$ добавляется в массив $(nTbS) \times (nTbS)$ *resSamples*, и массив *resSamples* хранится в *temporalBuffer* в

местоположении ($xTbP$, $yTbP$). В этом примере *resSamples* представляет собой второй набор остатков, а *tempPredSamples* представляет собой набор временных прогнозов, которые, как указано выше, могут быть получены из содержимого временного буфера 432, например, в зависимости от временной сигнализации, принятой в декодере. Таким образом, если временная обработка разрешена и сигнализируется о применении ее на уровне блока, содержимое временного буфера 432 добавляется 468 ко второму набору остатков для генерации второго набора остатков с временной коррекцией, который затем сохраняется во временном буфере 432.

В приведенных в данном документе примерах набор временных прогнозов, полученных из временного буфера, объединяется с набором остатков для каждого блока из множества блоков для выполнения временного прогнозирования. Это, например, включает получение соответствующего предварительного остаточного элемента из набора предварительных остатков для соответствующих элементов блока и получение соответствующего временного прогнозирования, связанного с элементом, причем временное прогнозирование может быть установлено равным нулю в некоторых случаях или получено из содержимого временного буфера, которое, в свою очередь, может быть обнулено определенной временной сигнализацией, принимаемой декодером. Соответствующие предварительные остаточные элементы затем могут быть объединены с временными прогнозами для блока, чтобы подать на выход восстановленный набор остаточных значений (например, остаточные значения уровня 2) для комбинации с восстановленным сигналом с дискретизацией с повышенной частотой с более низких уровней (например, уровня 1 и базового уровня).

В описанных в данном документе примерах декодер 400, 480 выполнен с возможностью обеспечения обнуления значений наборов временных прогнозов на трех уровнях: на уровне кадра; на уровне мозаичного фрагмента; и на уровне блока. Например, описанный выше *temporal_refresh_bit* может использоваться для указания обновления временного буфера 432 (например, обнуление значений временного буфера) на уровне кадра, при этом все значения во временном буфере, например, которые относятся к полному кадру, устанавливаются в ноль. Затем это может обеспечить обнуление значений временного прогнозирования, если эти значения позже извлекаются из временного буфера 432. На уровне блока временная сигнализация, которая указывает, должны ли значения временного прогнозирования быть получены из временного буфера 432 или установлены в ноль, может переноситься в значениях коэффициентов преобразования (например, значения коэффициента, такого как НН, для преобразования 4x4 до обратного преобразования). На уровне мозаичного фрагмента такой параметр, как

temporal_tile_intra_signalling_enabled может использоваться, чтобы указать, что сигнализация уровня блока для одного блока в мозаичном фрагменте должна использоваться для указания обновления временного буфера 432 для мозаичного фрагмента, то есть сигнализация уровня блока для одного блока в мозаичном фрагменте становится сигнализацией уровня мозаичного фрагмента. Когда временной буфер 432 обновляется, значения во временном буфере устанавливаются на 0, что затем означает, что для последующих блоков в том же мозаичном фрагменте значения, применяемые из временного буфера, будут равны нулю, независимо от сигнализации уровня блока.

Такой подход обеспечивает гибкость при обновлении временного буфера 432. Например, временной буфер 432 может обновляться на уровне кадра для первого кадра и на уровне мозаичного фрагмента по меньшей мере для одного мозаичного фрагмента второго кадра, и временные прогнозы могут быть обнулены на уровне блока по меньшей мере для одного блока третьего кадра. На уровне блока обнуление временных прогнозов можно рассматривать как операцию, эквивалентную обнулению временного буфера 432; обе операции приводят к элементам с нулевыми значениями, применяемыми при сложении 468. Например, декодер 400, 480 может быть выполнен с возможностью обновления значений временного буфера 432 для первого мозаичного фрагмента кадра и применения нулей для временных прогнозов для первого блока второго мозаичного фрагмента того же кадра, применяя ненулевые значения временных прогнозов для второго блока второго мозаичного фрагмента.

В описанном в данном документе процессе декодирования генерация декодированного видео может выполняться поблочно. Таким образом, создание блока элементов в кадре декодированного видео может выполняться без использования другого блока элементов в том же кадре декодированного видео, который был сгенерирован ранее. По этой причине процесс временного прогнозирования может выполняться параллельно для всех блоков элементов в кадре, в отличие от последовательного выполнения процесса временного прогнозирования для каждого блока элементов в кадре.

Дополнительные примеры описаны ниже.

Некоторые дополнительные примеры относятся к способу кодирования входного видео во множество кодированных потоков, так что кодированные потоки могут быть объединены для восстановления входного видео, причем способ включает: получение входного видео с полным разрешением; дискретизация с пониженной частотой входного видео с полным разрешением для создания видео с пониженной частотой дискретизации; кодирование видео с пониженной частотой дискретизации с использованием первого кодека для создания базового кодированного потока; восстановление видео из

кодированного видео для создания восстановленного видео; сравнение восстановленного видео с входным видео; и создание одного или большего количества дополнительных кодированных потоков на основе сравнения. Входное видео по сравнению с восстановленным видео может быть видео с пониженной частотой дискретизации.

Согласно иллюстративному способу сравнение восстановленного видео с входным видео включает: сравнение восстановленного видео с видео с пониженной частотой дискретизации для создания первого набора остатков, и при этом создание одного или большего количества дополнительных кодированных потоков включает кодирование первого набора остатков для создания кодированного потока первого уровня.

Входное видео по сравнению с восстановленным видео может быть входным видео с полным разрешением, а восстановленное видео может быть подвергнуто дискретизации с повышенной частотой.

Согласно иллюстративному способу сравнение восстановленного видео с входным видео включает: дискретизация с повышенной частотой восстановленного видео для создания восстановленного видео с повышенной частотой дискретизации; и сравнение восстановленного видео с повышенной частотой дискретизации с входным видео с полным разрешением для создания второго набора остатков, и при этом создание одного или большего количества дополнительных кодированных потоков включает кодирование второй разности для создания кодированного потока второго уровня.

Соответственно, в примере способ может генерировать базовый кодированный поток, кодированный поток первого уровня и кодированный поток второго уровня согласно приведенным выше иллюстративным способам. Каждый из кодированного потока первого уровня и кодированного потока второго уровня может содержать данные улучшения, используемые декодером для улучшения кодированного базового потока.

Согласно иллюстративному способу этап кодирования первого набора остатков включает: применение преобразования к набору остатков для создания набора коэффициентов; применение операции квантования к коэффициентам для создания набора квантованных коэффициентов; и применение операции кодирования к квантованным коэффициентам.

Согласно иллюстративному способу этап кодирования второго набора остатков включает: применение преобразования ко второму набору остатков для создания набора коэффициентов; применение операции квантования к коэффициентам для создания набора квантованных коэффициентов; и применение операции кодирования к квантованным коэффициентам.

Преобразование для кодирования первого и/или второго набора остатков может быть, например, дискретным косинусным преобразованием или вейвлет-преобразованием. В альтернативном примере преобразование может быть небольшим преобразованием (например, с использованием ядра 2×2 или ядра 4×4), которое разбивает блок элементов на направленные компоненты. Например, ядро 2×2 может быть преобразованием Адамара. Более подробную информацию о преобразовании можно найти, например, в патентных заявках PCT/EP2013/059847 или PCT/GB2017/052632, которые включены в настоящий документ посредством ссылки. В дополнительном примере кодер может выбирать между различными преобразованиями, которые будут использоваться, например, между ядром 2×2 и ядром 4×4 . Это обеспечивает дополнительную гибкость в способе кодирования остатков. Выбор преобразования может быть основан на анализе данных, которые нужно преобразовать.

К первому набору остатков и второму набору остатков могут применяться разные преобразования, и выбор может быть предварительно определен или осуществлен в ходе процесса. Об используемом преобразовании может сигнализироваться в заголовке.

Квантование для кодирования первого и/или второго набора остатков может быть, например, линейным квантованием. Линейный квантователь может использовать мертвую зону переменного размера. Операция кодирования может быть, например, энтропийным кодером и может включать кодирование длин серий и/или кодирование Хаффмана.

Остатки могут быть разностью между двумя видео или кадрами.

Этап кодирования первого набора остатков может включать: ранжирование первого набора остатков на основе предварительного анализа первого набора остатков; и выбор подмножества остатков для преобразования и кодирования.

В одном примере способ включает анализ первого набора остатков и, на основе анализа, выполнение следующих этапов или их отсутствие: ранжирование первого набора остатков; и выбор подмножества остатков для преобразования и кодирования.

В одном примере способ включает анализ первого набора остатков и: ранжирование первого набора остатков; и выбор подмножества остатков для преобразования и кодирования, так что этапы ранжирования и/или выбора выполняются по-разному на основе анализа.

Согласно иллюстративному способу этап применения преобразования выполняется на выбранном подмножестве остатков из первого набора остатков.

Этап кодирования второго набора остатков может включать: ранжирование второго набора остатков на основе предварительного анализа второго набора остатков; и выбор подмножества остатков для преобразования и кодирования.

В одном примере способ включает анализ второго набора остатков и, на основе анализа, выполнение следующих этапов или их отсутствие: ранжирование второго набора остатков; и/или выбор подмножества остатков для преобразования и кодирования.

В одном примере способ включает анализ второго набора остатков и: ранжирование второго набора остатков; и выбор подмножества остатков для преобразования и кодирования, так что этапы ранжирования и/или выбора выполняются по-разному на основе анализа.

Согласно иллюстративному способу этап применения преобразования выполняется на выбранном подмножестве остатков из второго набора остатков.

Кодированные потоки могут сопровождаться одним или большим количеством заголовков, которые содержат параметры, указывающие аспекты процесса кодирования для облегчения декодирования. Например, заголовки могут содержать используемый кодек, примененное преобразование, примененное квантование и/или другие параметры декодирования.

В некоторых примерах этап квантования может включать адаптацию квантования на основе анализа коэффициентов и/или данных, которые должны быть преобразованы, например, остаточных данных. В некоторых примерах распределение, используемое на этапе квантования, может быть адаптировано.

Этап кодирования первого набора остатков может включать: получение набора временных коэффициентов из временного буфера; и вычитание набора временных коэффициентов из набора коэффициентов.

Этап кодирования второго набора остатков может включать: получение набора временных коэффициентов из временного буфера; и вычитание набора временных коэффициентов из набора коэффициентов.

Выше было описано, как этап ранжирования и выбора может быть применен к остаточным данным, может быть выполнен этап вычитания временных коэффициентов и может быть адаптировано квантование. Каждый из этих этапов может быть предварительно определен и выборочно применен или может применяться на основе анализа входного видео, видео с пониженной частотой дискретизации, восстановленного видео, видео с повышенной частотой дискретизации или любой комбинации вышеперечисленного для улучшения общих рабочих характеристик кодера. Этапы могут применяться выборочно на основе предварительно определенного набора правил или определительно применяться на основе анализа или обратной связи касательно рабочих характеристик.

Согласно иллюстративному способу первый кодек является аппаратным кодеком, предпочтительно первый кодек представляет собой AVC, HEVC, AV1, VP8 или VP9.

Иллюстративный способ дополнительно включает отправку базового кодированного потока.

Иллюстративный способ дополнительно включает отправку кодированного потока первого уровня.

Иллюстративный способ дополнительно включает отправку кодированного потока второго уровня.

Некоторые дополнительные примеры относятся к способу декодирования множества кодированных потоков с получением восстановленного выходного видео, причем способ включает: получение первого базового кодированного потока; декодирование первого базового кодированного потока согласно первому кодеку для генерации первого выходного видео; получение одного или большего количества дополнительных кодированных потоков; декодирование одного или большего количества дополнительных кодированных потоков для генерации набора остатков; и объединение набора остатков с первым видео для генерации декодированного видео.

В одном примере способ включает извлечение множества параметров декодирования из заголовка. Параметры декодирования могут указывать, какие процедурные этапы были включены в процесс кодирования.

В одном примере этап декодирования одного или большего количества дополнительных кодированных потоков для генерации набора остатков включает: применение операции энтропийного декодирования; применение операции деквантования; и применение операции обратного преобразования для генерации набора остатков.

В одном примере этап декодирования одного или большего количества дополнительных кодированных потоков для генерации набора остатков включает: прогнозирование подмножества остатков на основе совмещенных остатков из временного буфера.

В одном из примеров способ может включать получение кодированного потока первого уровня и получение кодированного потока второго уровня. В этом примере этап декодирования одного или большего количества дополнительных кодированных потоков для генерации набора остатков включает: декодирование кодированного потока первого уровня для получения первого набора остатков; при этом этап объединения набора остатков с первым видео для генерации декодированного видео включает: объединение первого набора остатков с первым выходным видео для генерации второго выходного

видео; дискретизация с повышенной частотой второго выходного видео для генерации второго выходного видео с повышенной частотой дискретизации; декодирование кодированного потока второго уровня для получения второго набора остатков; и объединение второго набора остатков со вторым выходным видео для генерации восстановленного выходного видео.

В одном примере этап дискретизации с повышенной частотой второго выходного видео для генерации второго выходного видео с повышенной частотой дискретизации включает: добавление значения элемента в первом наборе остатков, из которого был получен блок во втором выходном видео с повышенной частотой дискретизации, в соответствующий блок во втором выходном видео с повышенной частотой дискретизации. Блок может быть блоком 2×2 . Данный этап добавления может выполняться выборочно на основе предварительно определенного значения или сигнала, включенного в заголовок.

В одном примере этап декодирования кодированного потока первого уровня для получения первого набора остатков включает: применение операции энтропийного декодирования; применение операции деквантования; и применение операции обратного преобразования для генерации первого набора остатков.

В этом примере этап декодирования кодированного потока первого уровня для получения первого набора остатков включает: применение деблокирующего фильтра, выполненного с возможностью применения маски к блоку остатков. Маска может быть взвешена в соответствии с набором предварительно определенных весов.

В одном примере этап декодирования кодированного потока второго уровня для получения второго набора остатков включает: применение операции энтропийного декодирования; применение операции деквантования; и применение операции обратного преобразования для генерации второго набора остатков.

Операция обратного преобразования может быть операцией, обратной операциям, определенным выше, или может быть по существу зеркальной операцией. То есть может выборочно применяться блочное преобразование 2×2 или 4×4 . Преобразование может быть обнаружено методом декодирования или сигнализировано в заголовке.

Если используется преобразование 2×2 , коэффициенты могут быть модифицированы путем добавления значения остатка, из которого прогнозируется преобразованный блок остатков. Если используется преобразование 4×4 , коэффициенты будут изменены путем добавления среднего значения четырех остатков.

Способ может дополнительно включать отображение или подачу на выход восстановленного выходного сигнала.

В одном примере способ декодирования множества кодированных потоков в восстановленное выходное видео включает: получение кодированного потока улучшения; декодирование кодированного потока улучшения для получения набора остатков, при этом декодирование включает: получение набора предварительного набора остатков из кодированного потока улучшения; получение набора временных прогнозов с использованием временного буфера; и добавление набора временных прогнозов к набору предварительных остатков для подачи на выход обработанного набора остатков (например, остатков уровня 2). В этом примере способ декодирования множества кодированных потоков в восстановленное выходное видео также включает: получение первого выходного видео, причем первое выходное видео содержит выходной сигнал базового декодера, примененный к кодированному потоку базового уровня; получение дополнительного кодированного потока улучшения, относящегося к более низкому уровню; декодирование дополнительного кодированного потока улучшения для получения дополнительного набора остатков, комбинирующего дополнительный набор остатков с первым выходным видео для генерации второго выходного видео; дискретизация с повышенной частотой второго выходного видео для генерации второго выходного видео с повышенной частотой дискретизации; и объединение обработанного набора остатков со вторым выходным видео с повышенной частотой дискретизации для генерации восстановленного выходного видео.

В одном примере декодирование кодированного потока улучшения для получения набора предварительных остатков включает: применение операции энтропийного декодирования; применение операции деквантования; применение операции преобразования для генерации набора предварительных остатков.

В примере декодирование дополнительного кодированного потока улучшения для получения дополнительного набора остатков включает: получение дополнительного набора предварительных остатков из дополнительного кодированного потока улучшения; получение дополнительного набора временных прогнозов с использованием второго временного буфера; и добавление дополнительного набора временных прогнозов к дополнительному набору предварительных остатков для подачи на выход дополнительного набора остатков (уровень 1).

В одном примере декодирование дополнительного кодированного потока улучшения (т. е. потока уровня 1) для получения дополнительного набора остатков (т. е. остатков уровня 1) включает: применение операции энтропийного декодирования; применение операции деквантования; применение операции преобразования для генерации дополнительного набора остатков. Временная обработка, описанная в этих

примерах, может применяться подобным образом и выборочно к каждому из уровней улучшения, описанных в данном документе.

Некоторые описанные в данном документе примеры относятся к временной обработке в связи с одним или большим количеством кодированных потоков. Декодер может быть выполнен с возможностью осуществления временной обработки с использованием временного буфера.

Декодер может быть выполнен с возможностью декодирования множества кодированных потоков в восстановленное выходное видео. Декодер может быть выполнен с возможностью получения первого выходного видео, причем первое выходное видео содержит выходной сигнал базового декодера, примененный к кодированному потоку базового уровня. Декодер может быть выполнен с возможностью получения одного или большего количества дополнительных кодированных потоков. Декодер может быть выполнен с возможностью декодирования соответствующих кадров одного или большего количества дополнительных кодированных потоков для получения соответствующих наборов остатков.

В некоторых случаях каждый кадр соответствующих кадров может быть разделен на множество мозаичных фрагментов. Каждый мозаичный фрагмент может быть разделен на множество блоков.

В других случаях каждый кадр соответствующих кадров может быть разделен на множество плоскостей. Каждая плоскость может быть разделена на множество мозаичных фрагментов. Каждый мозаичный фрагмент из множества мозаичных фрагментов может быть разделен на множество блоков.

Декодер также может быть выполнен с возможностью объединения наборов остатков с первым выходным видео для генерации восстановленного выходного видео. Для декодирования соответствующих кадров декодер может быть выполнен с возможностью получения для каждого блока из множества блоков предварительного набора остатков из одного или большего количества дополнительных кодированных потоков. Чтобы декодировать соответствующие кадры, декодер может быть выполнен с возможностью получения набора временных прогнозов с использованием временного буфера. Чтобы декодировать соответствующие кадры, декодер может быть выполнен с возможностью объединения набора временных прогнозов с предварительным набором остатков для подачи на выход данных для объединения с первым выходным видео.

Декодер может быть выполнен с возможностью обеспечения выборочного обнуления значений набора временных прогнозов. Декодер может быть выполнен с возможностью обеспечения обнуления значений набора временных прогнозов на уровне

кадра по меньшей мере для одного из соответствующих кадров. Декодер может быть выполнен с возможностью обеспечения обнуления значений набора временных прогнозов на уровне мозаичного фрагмента по меньшей мере для одного из множества мозаичных фрагментов. Декодер может быть выполнен с возможностью обеспечения обнуления значений на уровне блока по меньшей мере для одного из множества блоков.

Декодер может быть выполнен с возможностью получения соответствующих остаточных элементов блока из множества блоков. Декодер может быть выполнен с возможностью получения соответствующего временного прогнозирования набора временных прогнозов из временного буфера для каждого из соответствующих остаточных элементов.

Декодер может быть выполнен с возможностью обнуления значений набора временных прогнозов путем обновления по меньшей мере части временного буфера.

Некоторые дополнительные примеры относятся к аппарату для кодирования набора данных в набор кодированных данных, содержащий заголовок и полезную нагрузку. Аппарат выполнен с возможностью кодирования входного видео в соответствии с вышеуказанными этапами. Аппарат может содержать процессор, выполненный с возможностью осуществления способа согласно любому из вышеупомянутых аспектов.

Некоторые дополнительные примеры относятся к аппарату для декодирования набора данных в восстановленное видео из набора данных, содержащего заголовок и полезную нагрузку. Аппарат выполнен с возможностью декодирования выходного видео в соответствии с вышеуказанными этапами. Аппарат может содержать процессор, выполненный с возможностью осуществления способа согласно любому из вышеупомянутых аспектов.

Также могут быть предусмотрены кодер и декодер.

Некоторые дополнительные примеры относятся к машиночитаемым носителям, которые при исполнении процессором побуждают процессор выполнять любой из способов согласно вышеупомянутым аспектам.

Вышеупомянутые варианты реализации изобретения следует понимать как иллюстративные примеры. Предусмотрены другие варианты реализации изобретения. Следует понимать, что любой признак, описанный в отношении любого одного варианта реализации изобретения, может использоваться отдельно или в комбинации с другими описанными признаками, а также может использоваться в сочетании с одним или большим количеством признаков любого другого варианта реализации изобретения или любой комбинацией любого другого варианта реализации изобретения. Кроме того,

эквиваленты и модификации, не описанные выше, также могут быть использованы в пределах объема прилагаемой формулы изобретения.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Декодер, выполненный с возможностью декодирования множества кодированных потоков в восстановленное выходное видео, причем декодер выполнен с возможностью:

получения первого выходного видео, причем первое выходное видео содержит выходной сигнал базового декодера, примененный к кодированному потоку базового уровня;

получения одного или большего количества дополнительных кодированных потоков;

декодирования соответствующих кадров одного или большего количества дополнительных кодированных потоков для выведения соответствующих наборов остатков, причем каждый кадр соответствующих кадров делится на множество мозаичных фрагментов, и каждый мозаичный фрагмент из множества мозаичных фрагментов делится на множество блоков; и

объединения наборов остатков с первым выходным видео, чтобы сгенерировать восстановленное выходное видео,

при этом для декодирования соответствующих кадров декодер выполнен с возможностью:

получения для каждого блока из множества блоков предварительного набора остатков из одного или большего количества дополнительных кодированных потоков;

получения набора временных прогнозов с использованием временного буфера; и

объединения набора временных прогнозов с предварительным набором остатков для подачи на выход данных для объединения с первым выходным видео;

и

при этом декодер выполнен с возможностью обнуления значений набора временных прогнозов: на уровне кадра для по меньшей мере одного из соответствующих кадров; на уровне мозаичного фрагмента для по меньшей мере одной из множества мозаичных фрагментов; и на уровне блока для по меньшей мере одного из множества блоков.

2. Декодер по п. 1, отличающийся тем, что выполнен с возможностью генерации первого блока элементов из кадра в восстановленном выходном видео без использования второго блока элементов из кадра в восстановленном выходном видео.
3. Декодер по п. 1 или 2, отличающийся тем, что выполнен с возможностью:
получения соответствующих остаточных элементов блока из множества блоков; и
получения соответствующего временного прогнозирования набора временных прогнозов из временного буфера для каждого из соответствующих остаточных элементов.
4. Декодер по любому из пп. 1–3, дополнительно выполненный с возможностью применения операции энтропийного декодирования к одному или большему количеству дополнительных кодированных потоков и получения предварительного набора остатков путем применения операции деквантования к соответствующему блоку и применения операции обратного преобразования к соответствующему блоку.
5. Декодер по любому из пп. 1–4, отличающийся тем, что выполнен с возможностью обнуления значений набора временных прогнозов путем обновления по меньшей мере части временного буфера.
6. Декодер по любому из пп. 1–5, отличающийся тем, что выполнен с возможностью обнуления значений временных прогнозов в ответ на получение первого параметра с первым значением, которое указывает, что временная обработка разрешена.
7. Декодер по п. 6, отличающийся тем, что длина в битах первого значения первого параметра составляет один бит.
8. Декодер по п. 6 или п. 7, отличающийся тем, что первый параметр принимается один раз для группы изображений, связанных с множеством кодированных потоков.
9. Декодер по любому из пп. 1–8, отличающийся тем, что выполнен с возможностью:
получения второго параметра из данных из одного или большего количества дополнительных кодированных потоков для каждого соответствующего блока из множества блоков; и
установки значений временных прогнозов для соответствующего блока из множества блоков равными нулю в ответ на определение того, что второй параметр для

соответствующего блока имеет второе значение, которое обеспечивает временную сигнализацию для соответствующего блока.

10. Декодер по любому из пп. 1–9, отличающийся тем, что выполнен с возможностью обновления временного буфера для мозаичного фрагмента из множества мозаичных фрагментов в ответ на получение второго параметра для указанного блока в мозаичном фрагменте.

11. Декодер по п. 10, отличающийся тем, что выполнен с возможностью обновления значений временного буфера, связанного с мозаичным фрагментом, в ответ на получение как второго параметра со вторым значением, так и третьего параметра с третьим значением, указывающим, что получение второго параметра для указанного блока со вторым значением указывает, что значения во временном буфере, связанном с мозаичный фрагментом, должны быть обновлены.

12. Декодер по п. 11, отличающийся тем, что длина в битах третьего значения третьего параметра составляет один бит.

13. Декодер по п. 11 или п. 12, отличающийся тем, что третий параметр принимается один раз для группы изображений, связанных с множеством кодированных потоков.

14. Декодер по любому из пп. 1–13, отличающийся тем, что выполнен с возможностью обновления значений временного буфера для первого мозаичного фрагмента кадра соответствующих кадров и обнуления значений для набора временных прогнозов для первого блока второго мозаичного фрагмента кадра без обнуления значений набора временных прогнозов для второго блока второго мозаичного фрагмента.

15. Декодер по любому из пп. 1–14, отличающийся тем, что содержит декодер улучшения, выполненный с возможностью получения остаточных данных, сгенерированных в результате сравнения данных, полученных из входного видео, и данных, полученных из кодированного потока базового уровня, при этом один или большее количество дополнительных кодированных потоков содержат кодированные остаточные данные, которые являются декодируемыми для восстановления одного или большего количества дополнительных наборов остаточных данных для применения к первому выходному видео.

16. Декодер по любому из пп. 1–15, отличающийся тем, что декодер улучшения отличается от базового декодера.

17. Декодер по любому из пп. 1–16, отличающийся тем, что набор коэффициентов преобразования для декодирования набора остатков для первого кадра одного или большего количества дополнительных кодированных потоков представляет собой разность между набором коэффициентов преобразования для первого кадра и дополнительным набором коэффициентов преобразования для второго кадра одного или большего количества дополнительных кодированных потоков, отличных от первого кадра.

18. Декодер по любому из пп. 1–17, отличающийся тем, что каждый кадр соответствующих кадров разделен на множество плоскостей, причем каждая плоскость во множестве плоскостей делится на множество мозаичных фрагментов, и каждый мозаичный фрагмент из множества мозаичных фрагментов разделяется на множество блоков.

19. Способ декодирования множества кодированных потоков с получением восстановленного выходного видео, включающий:

получение первого выходного видео, причем первое выходное видео содержит выходной сигнал базового декодера, примененный к кодированному потоку базового уровня;

получение одного или большего количества дополнительных кодированных потоков;

декодирование соответствующих кадров одного или большего количества дополнительных кодированных потоков для выведения соответствующих наборов остатков, причем каждый кадр соответствующих кадров делится на множество мозаичных фрагментов, и каждый мозаичный фрагмент из множества мозаичных фрагментов делится на множество блоков; и

объединение набора остатков с первым выходным видео для генерации восстановленного выходного видео,

при этом декодирование дополнительно включает:

получение для каждого блока из множества блоков набора предварительных остатков из одного или большего количества дополнительных кодированных потоков;

получение набора временных прогнозов с использованием временного буфера; и

объединение набора временных прогнозов с набором предварительных остатков; и

при этом декодирование включает обнуление значений набора временных прогнозов: на уровне кадра для по меньшей мере одного из соответствующих кадров; на уровне мозаичного фрагмента для по меньшей мере одной из множества мозаичных фрагментов; и на уровне блока для по меньшей мере одного из множества блоков.

20. Способ по п. 19, включающий создание первого блока элементов из кадра в восстановленном выходном видео без использования второго блока элементов из кадра в восстановленном выходном видео.

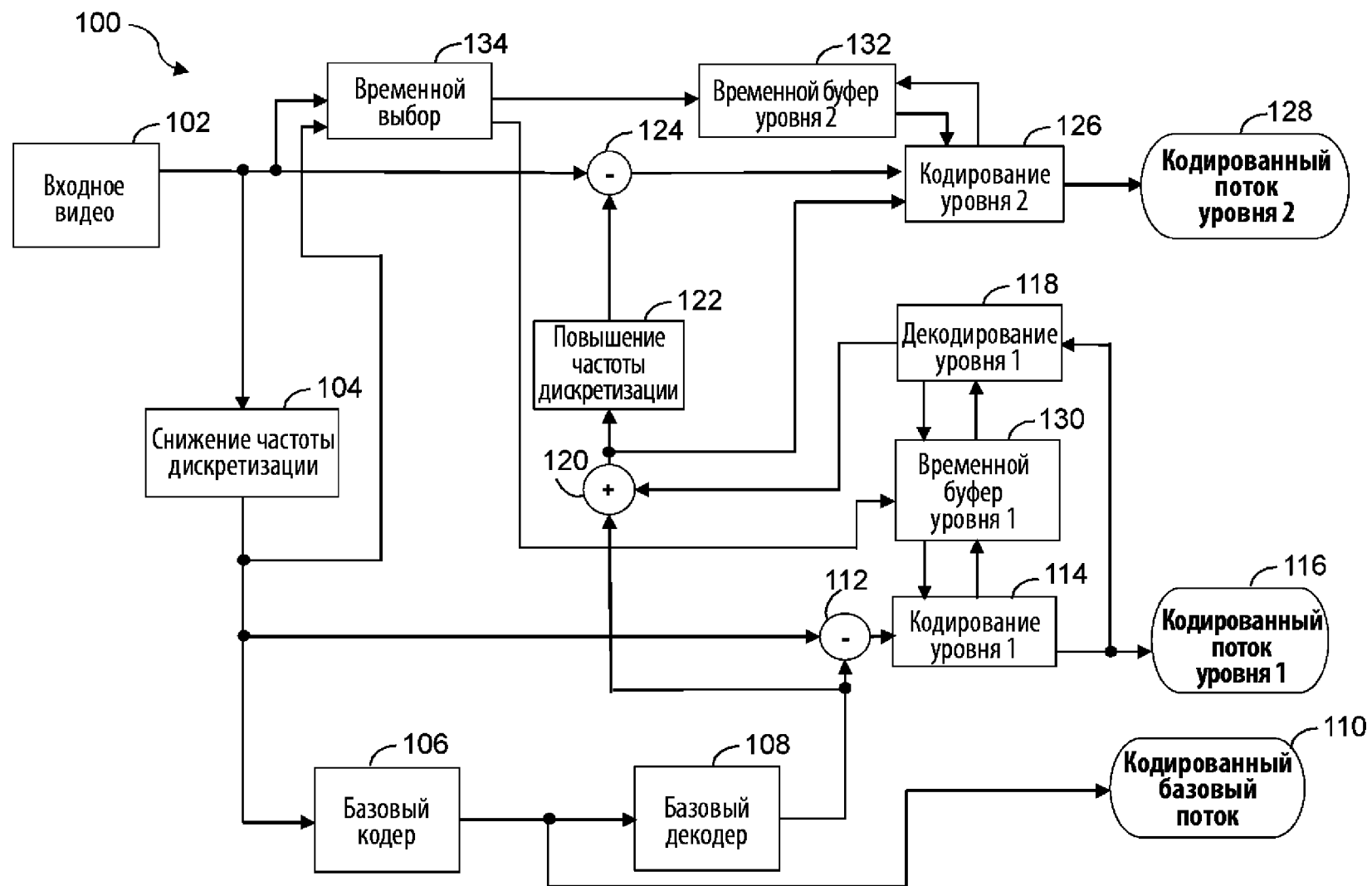
21. Способ по п. 19 или 20, отличающийся тем, что декодирование включает:

получение первого параметра на уровне кадра, указывающего, разрешена ли временная обработка;

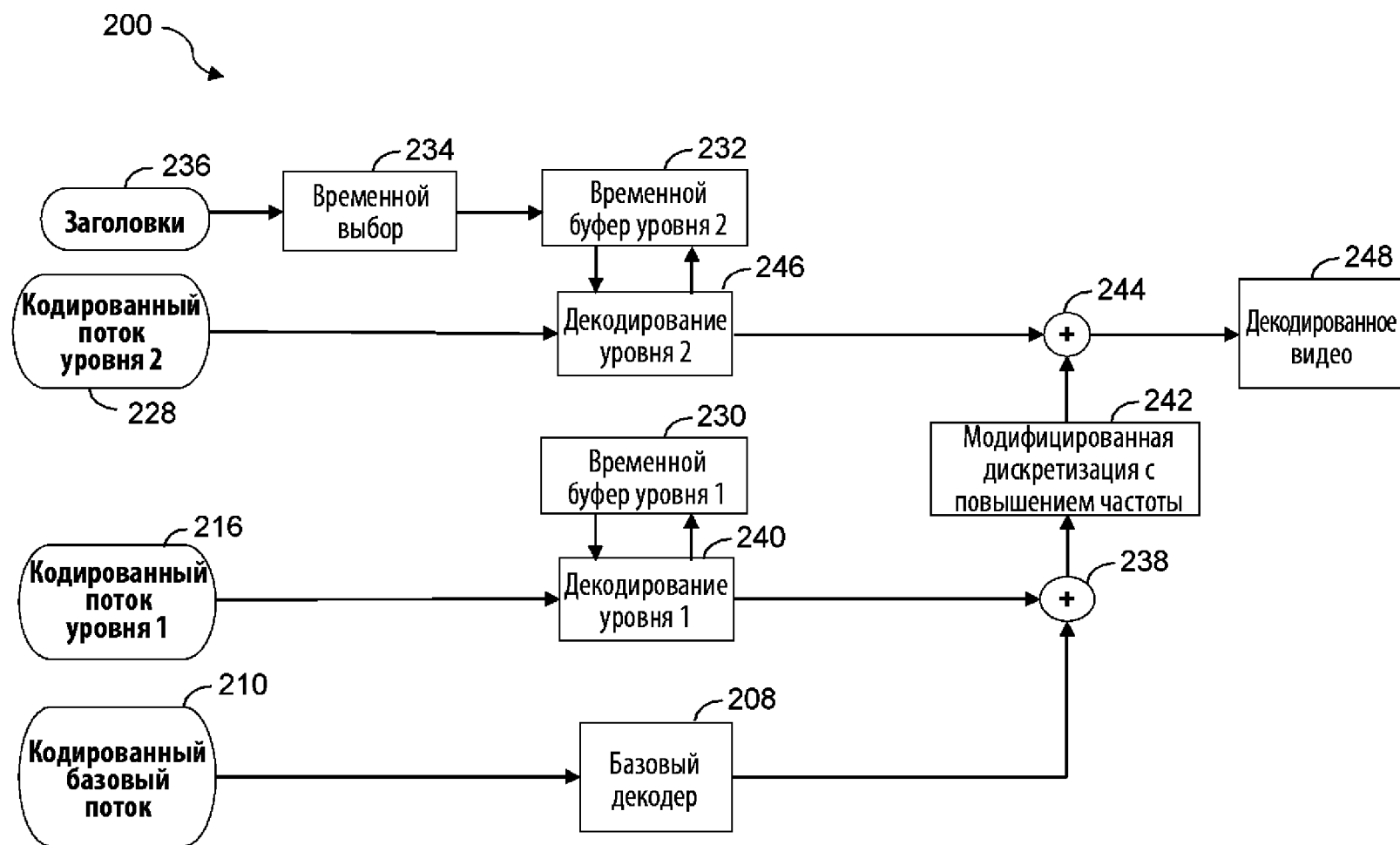
получение второго параметра на уровне блока, указывающего, должен ли использоваться временный буфер; и

получение третьего параметра на уровне мозаичного фрагмента, указывающего, следует ли использовать второй параметр в качестве временной сигнализации для мозаичного фрагмента,

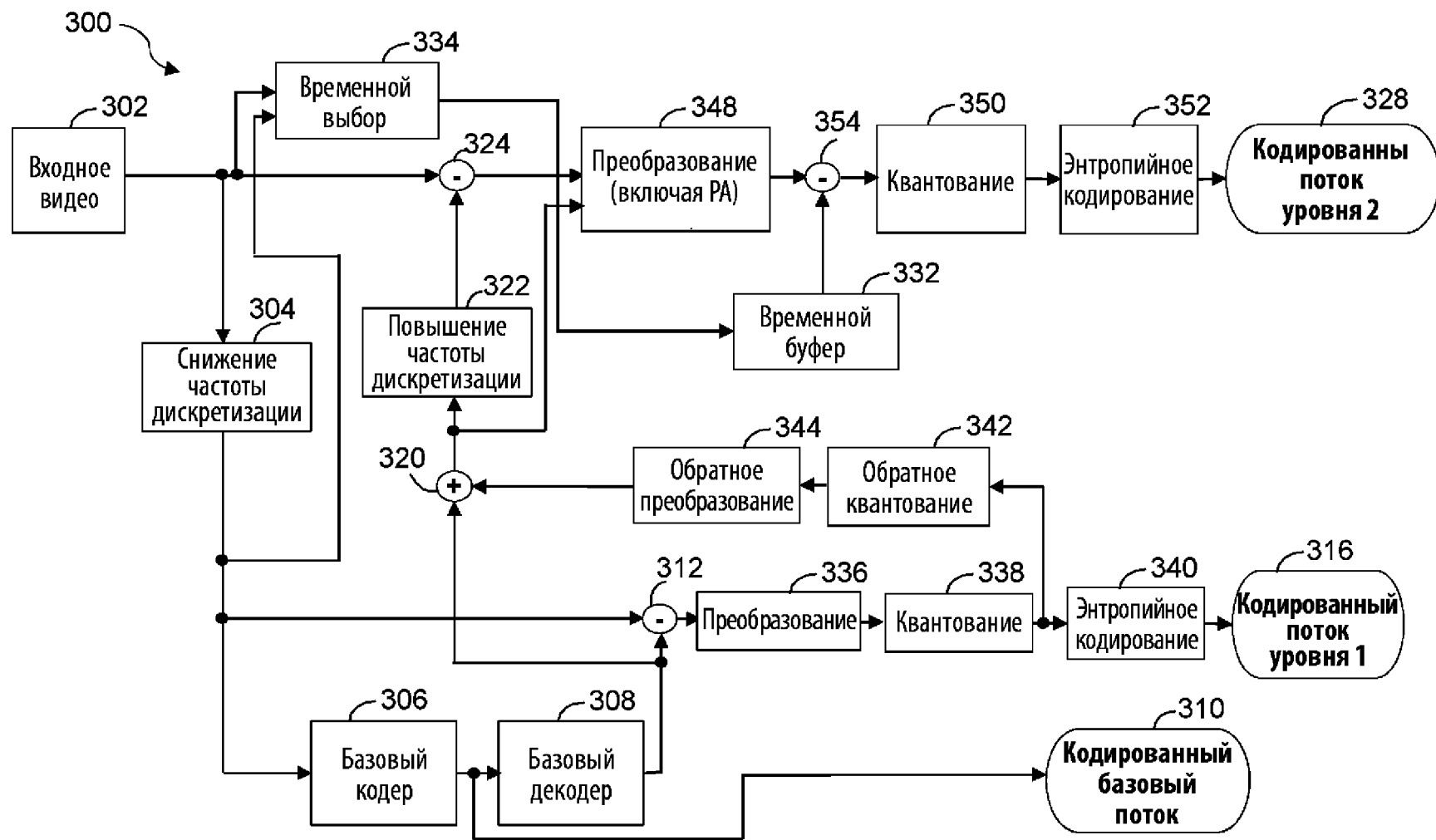
при этом инструкции касательно указанного получения подаются первым, вторым и третьим параметрами.



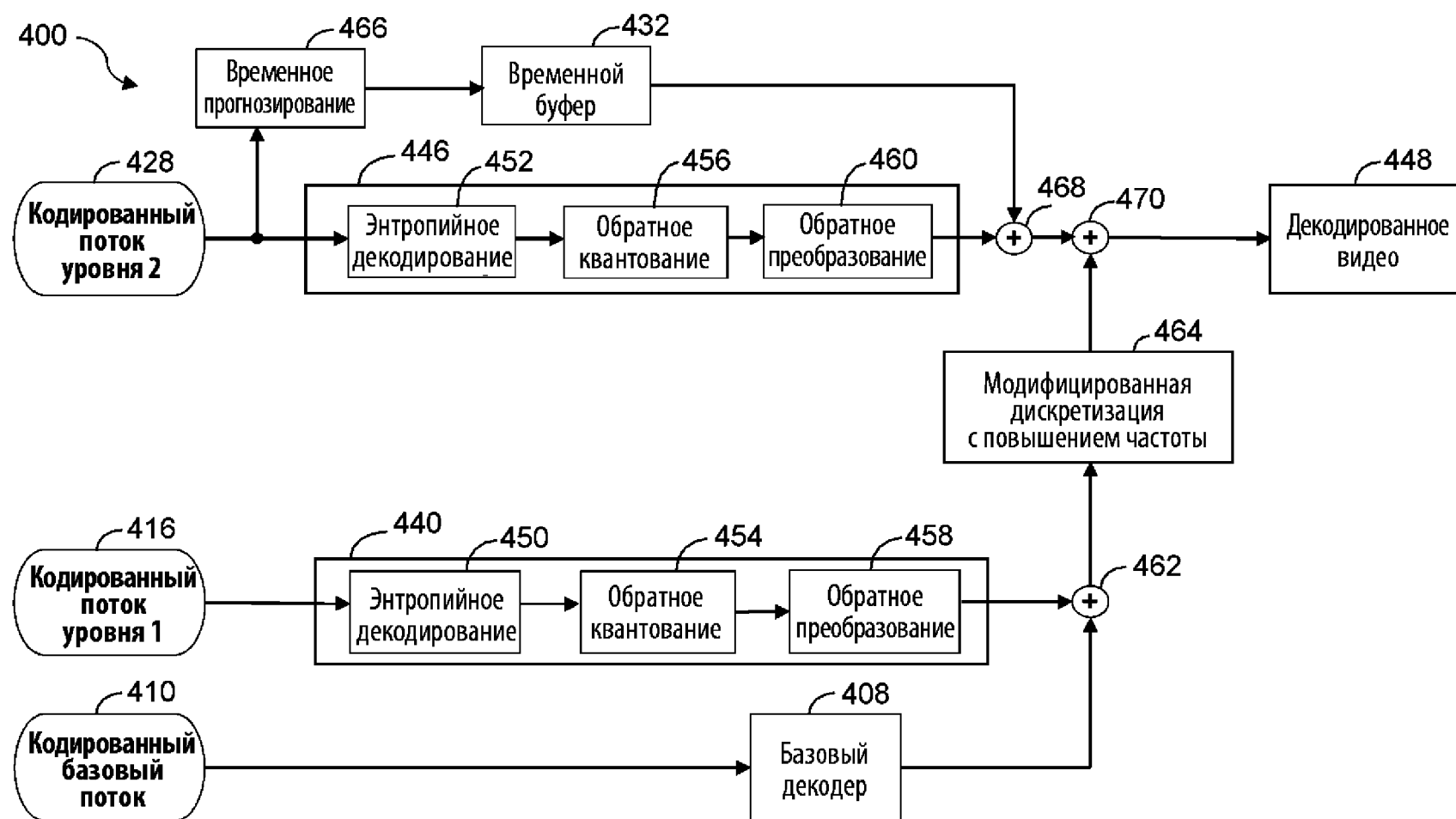
Фиг. 1



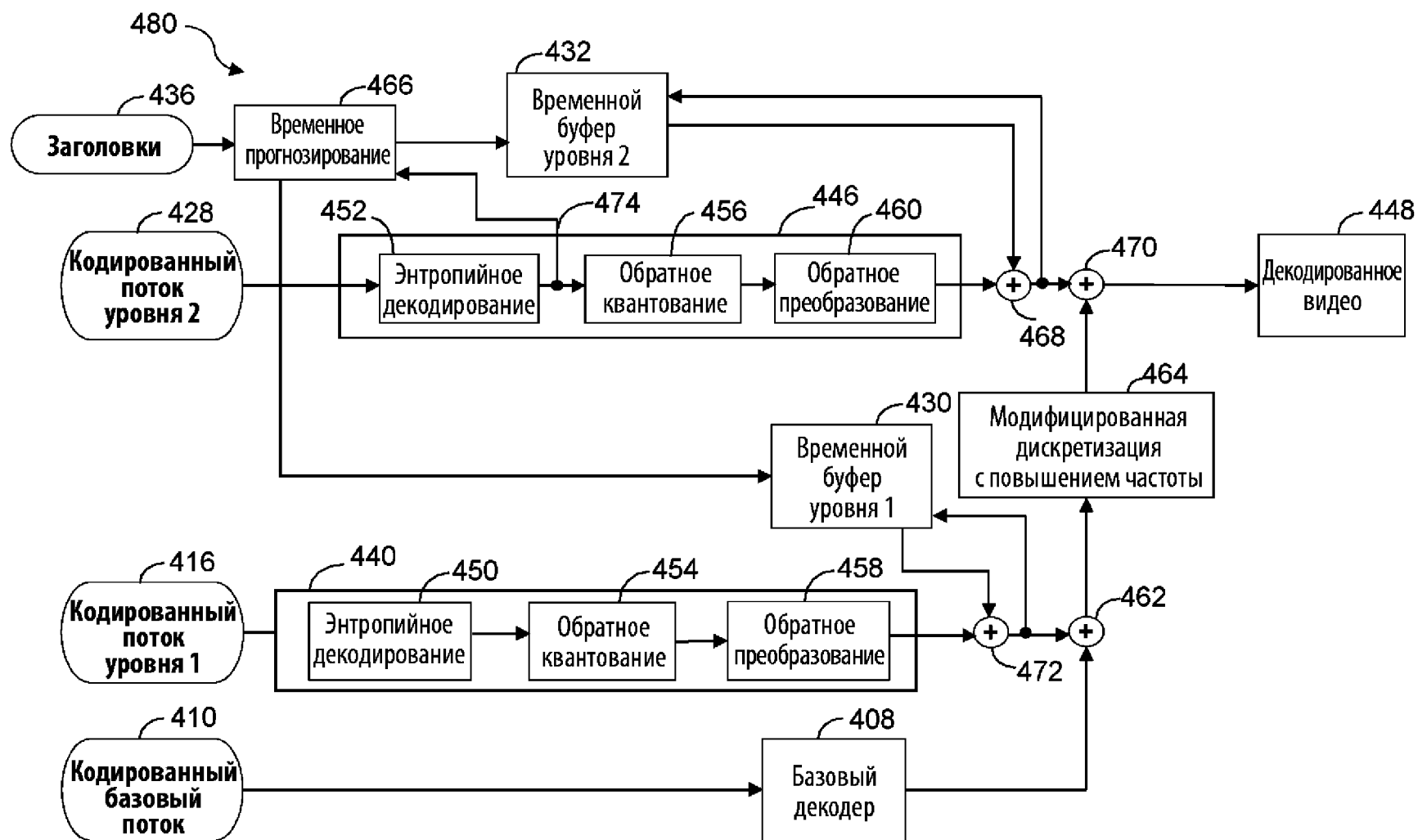
Фиг. 2



Фиг. 3А



Фиг. 4А



Фиг. 4В