

(19)



**Евразийское  
патентное  
ведомство**

(11) **034473**

(13) **B1**

**(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ**

**(45)** Дата публикации и выдачи патента  
**2020.02.12**

**(21)** Номер заявки  
**201791476**

**(22)** Дата подачи заявки  
**2016.01.27**

**(51)** Int. Cl. **H04N 19/593** (2014.01)  
**H04N 19/186** (2014.01)  
**H04N 19/105** (2014.01)  
**H04N 19/463** (2014.01)

---

**(54) СИСТЕМА И СПОСОБ КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ ВИДЕОДАНЫХ**

---

**(31)** **62/108,491; 15/006,994**

**(32)** **2015.01.27; 2016.01.26**

**(33)** **US**

**(43)** **2017.11.30**

**(86)** **PCT/US2016/015129**

**(87)** **WO 2016/123219 2016.08.04**

**(71)(73)** Заявитель и патентовладелец:  
**КВЭЛКОММ ИНКОРПОРЕЙТЕД**  
**(US)**

**(72)** Изобретатель:  
**Чэнь Цзяньлэ, Лю Хунбинь, Чэнь**  
**Ин, Чжан Ли, Ли Сян, Чжао Синь,**  
**Карчевич Марта (US)**

**(74)** Представитель:  
**Медведев В.Н. (RU)**

**(56)** **WO-A1-2013067667**

**WO-A1-2014190171**

**ZHANG L. ET AL.:** "Non-SCCE5: On index signaling of cross component prediction", 18. JCTVC MEETING; 30-6-2014 - 9-7-2014; SAPPORO; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ); URL: HTTP://WFPT3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-R0178, 20 June 2014 (2014-06-20), XP030116466, page 1 - page 2

**CHEN J. ET AL.:** "CE6.a.4: Chroma intra prediction by reconstructed luma samples", 20110312, no. JCTVC-E266, 12 March 2011 (2011-03-12) , XP030008772, ISSN: 0000-0007 cited in the application page 1 - page 2

---

**(57)** Система и способ для кодирования и декодирования видеоданных. Предсказанный остаточный сигнал целевой цветовой компоненты определяется в качестве функции одного или более линейных параметров линейной модели и остаточного сигнала цветовой компоненты источника. Остаточный сигнал целевой цветовой компоненты определяется в качестве функции оставшегося остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты.

---

**B1**

**034473**

**034473 B1**

Данная заявка на патент испрашивает приоритет предварительной заявки на патент США № 62/108,491, поданной 27 января 2015 года, полное содержимое которой включено посредством ссылки в этот документ.

### **Область техники, к которой относится изобретение**

Настоящее раскрытие относится к кодированию видео.

#### **Уровень техники**

Возможности цифрового видео могут быть включены в широкий ряд устройств, включая цифровые телевизоры, цифровые системы прямого вещания, беспроводные системы вещания, персональные цифровые помощники (PDA), портативные или настольные компьютеры, планшеты, устройства для чтения электронных книг, цифровые камеры, цифровые записывающие устройства, цифровые медиапроигрыватели, устройства видеоигр, игровые видеоприставки, сотовые или спутниковые радио телефоны, так называемые "смартфоны", устройства для видео телеконференции, устройства потокового видео и тому подобное. Цифровые видеоустройства реализуют технологии кодирования видео, как, например, описанные в стандартах, заданных MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, часть 10, усовершенствованное кодирование видео (AVC) ITU-T H.265, высокоэффективное кодирование видео (HEVC), и в расширениях таких стандартов, как, например, масштабируемое кодирование видео (SVC), многовидовое кодирование видео (MVC), масштабируемое HEVC (SHVC), многовидовое HEVC (MV-HEVC), 3D-HEVC, и расширения ряда HEVC. Видеоустройства могут передавать, принимать, кодировать, декодировать и/или хранить цифровую видеoinформацию более эффективно посредством реализации таких технологий кодирования видео.

Технологии кодирования видео включают в себя пространственное (внутри изображения) предсказание и/или временное (между изображениями) предсказание для сокращения или устранения избыточности, присущей для видеопоследовательностей. Для основанного на блоках кодирования видео, слайс (вырезка) видео (например, видеокадр или участок видеокадра) может быть секционирован на видеоблоки, которые также могут упоминаться как древовидные блоки, единицы дерева кодирования (CTU), единицы кодирования (CU) и/или узлы кодирования. Видеоблоки могут включать в себя блоки яркости и блоки цветности. Во внутренне-кодируемом (I) слайсе изображения блоки кодируются с использованием пространственного предсказания относительно опорных выборок в соседних блоках в идентичном изображении. Видеоблоки во внешне-кодируемом (P или B) слайсе изображения могут использовать пространственное предсказание относительно опорных выборок в соседних блоках в идентичном изображении или временное предсказание относительно опорных выборок в других опорных изображениях. Изображения могут упоминаться как кадры, и опорные изображения могут упоминаться как опорные кадры.

Пространственное или временное предсказание дает в результате предсказанный блок для блока, подлежащего кодированию. Остаточные данные представляют разности пикселей между исходным блоком, который должен быть кодирован, и предсказанным блоком. Внешне-кодированный блок кодируется в соответствии с вектором движения, который указывает на блок опорных выборок, формирующих предсказанный блок, и остаточными данными, показывающими разность между кодируемым блоком и предсказанным блоком. Внутренне-кодированный блок кодируется согласно режиму внутреннего кодирования и остаточным данным. Для дополнительного сжатия, остаточные данные могут быть преобразованы из пиксельной области в область преобразования, что дает в результате остаточные коэффициенты преобразования, которые затем могут быть квантованы. Квантованные коэффициенты преобразования могут быть энтропийно кодированы для достижения еще большего сжатия.

#### **Сущность изобретения**

Данное раскрытие относится к области кодирования видео и, более конкретно, относится к адаптивному кросс-компонентному предсказанию остатка (предсказания). Конкретные технологии этого раскрытия могут сокращать, устранять или иначе управлять межкомпонентной избыточностью. Оно может быть использовано в контексте усовершенствованных видеокодеков, таких как расширения HEVC, или в контексте следующего поколения стандартов кодирования видео.

В одном аспекте это раскрытие описывает способ кодирования видеоданных, причем способ содержит определение, в кодере, предсказанного остаточного сигнала целевой цветовой компоненты текущего блока в качестве функции одного или более параметров линейной модели и остаточного сигнала цветовой компоненты источника текущего блока; определение, в кодере, оставшегося остаточного сигнала для целевой цветовой компоненты на основе остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты; кодирование значений оставшегося остаточного сигнала целевой цветовой компоненты; и вывод кодированных значений оставшегося остаточного сигнала в битовом видеопотоке.

В другом аспекте это раскрытие описывает устройство для кодирования видеоданных, причем устройство содержит: память, сконфигурированную с возможностью хранения видеоданных; и один или более процессоров, сконфигурированных с возможностью: определения предсказанного остаточного сигнала целевой цветовой компоненты текущего блока в качестве функции одного или более параметров линейной модели и остаточного сигнала цветовой компоненты источника текущего блока; определения оставшегося остаточного сигнала для целевой цветовой компоненты на основе остаточного сигнала це-

левой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты; кодирования значений оставшегося остаточного сигнала целевой цветовой компоненты; и вывода кодированных значений оставшегося остаточного сигнала в битовом видеопотоке.

В другом аспекте это раскрытие описывает устройство для кодирования видеоданных, причем устройство содержит: средство для определения, в кодере, предсказанного остаточного сигнала целевой цветовой компоненты текущего блока в качестве функции одного или более параметров линейной модели и остаточного сигнала цветовой компоненты источника текущего блока; средство для определения, в кодере, оставшегося остаточного сигнала для целевой цветовой компоненты на основе остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты; средство для кодирования значений оставшегося остаточного сигнала целевой цветовой компоненты; и средство для вывода кодированных значений оставшегося остаточного сигнала в битовом видеопотоке.

В другом аспекте это раскрытие описывает компьютерно-читаемый запоминающий носитель, содержащий сохраненные на нем инструкции, которые при исполнении предписывают одному или более процессорам устройства для кодирования видеоданных: определять предсказанный остаточный сигнал целевой цветовой компоненты текущего блока в качестве функции одного или более параметров линейной модели и остаточного сигнала цветовой компоненты источника текущего блока; определять оставшийся остаточный сигнал для целевой цветовой компоненты на основе остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты; кодировать значения оставшегося остаточного сигнала целевой цветовой компоненты; и выводить кодированные значения оставшегося остаточного сигнала в битовом видеопотоке.

В другом аспекте это раскрытие описывает способ декодирования видеоданных, содержащий: определение в качестве функции одного или более линейных параметров линейной модели и остаточного сигнала цветовой компоненты источника предсказанного остаточного сигнала целевой цветовой компоненты; и определение остаточного сигнала целевой цветовой компоненты в качестве функции декодированного оставшегося остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты.

В другом аспекте это раскрытие описывает устройство для декодирования видеоданных, причем устройство содержит: память, сконфигурированную с возможностью хранения видеоданных; и один или более процессоров, сконфигурированных с возможностью определения в качестве функции одного или более линейных параметров линейной модели и остаточного сигнала цветовой компоненты источника предсказанного остаточного сигнала целевой цветовой компоненты; и определения остаточного сигнала целевой цветовой компоненты в качестве функции декодированного оставшегося остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты.

В другом аспекте это раскрытие описывает устройство для декодирования видеоданных, причем устройство содержит: средство для определения в качестве функции одного или более линейных параметров линейной модели и остаточного сигнала цветовой компоненты источника предсказанного остаточного сигнала целевой цветовой компоненты; и средство для определения остаточного сигнала целевой цветовой компоненты в качестве функции декодированного оставшегося остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты.

В другом аспекте это раскрытие описывает компьютерно-читаемый запоминающий носитель, содержащий сохраненные на нем инструкции, которые при исполнении, предписывают одному или более процессорам устройства для декодирования видеоданных: определять в качестве функции одного или более линейных параметров линейной модели и остаточного сигнала цветовой компоненты источника предсказанный остаточный сигнал целевой цветовой компоненты; и определять остаточный сигнал целевой цветовой компоненты в качестве функции декодированного оставшегося остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты.

Подробности одного или более примеров данного раскрытия изложены на прилагаемых чертежах и в нижеприведенном описании. Другие признаки, цели и преимущества будут очевидны из описания, чертежей и формулы изобретения.

#### **Краткое описание чертежей**

Фиг. 1 представляет собой блок-схему, иллюстрирующую примерную систему кодирования и декодирования видео, которая может быть сконфигурирована или иначе приготовлена для реализации или другого использования одной или более технологий, описанных в этом раскрытии.

Фиг. 2 представляет собой блок-схему, иллюстрирующую примерный видеокодер, который может быть сконфигурирован или иначе приготовлен для реализации или другого использования одной или более технологий, описанных в этом раскрытии.

Фиг. 3 представляет собой блок-схему, иллюстрирующую примерный видеокодер, который может быть сконфигурирован или иначе приготовлен для реализации или другого использования одной или более технологий, описанных в этом раскрытии.

Фиг. 4 представляет собой блок-схему последовательности операций, иллюстрирующую пример кодирования видеоданных в соответствии с одним аспектом этого раскрытия.

Фиг. 5 представляет собой блок-схему последовательности операций, иллюстрирующую более под-

робный пример кодирования видеоданных в соответствии с одним аспектом этого раскрытия.

Фиг. 6 представляет собой блок-схему последовательности операций, иллюстрирующую предсказание остаточного сигнала первой цветовой компоненты из остаточного сигнала второй цветовой компоненты в соответствии с одним аспектом этого раскрытия.

Фиг. 7 представляет собой блок-схему последовательности операций, иллюстрирующую предсказание остаточного сигнала цветовой компоненты Cb из остаточного сигнала цветовой компоненты Cr в соответствии с одним аспектом этого раскрытия.

Фиг. 8 представляет собой блок-схему последовательности операций, иллюстрирующую выведение параметров линейной модели на основе внешнего предсказания в соответствии с одним аспектом этого раскрытия.

Фиг. 9 представляет собой блок-схему последовательности операций, иллюстрирующую выведение параметров линейной модели на основе внутреннего предсказания в соответствии с одним аспектом этого раскрытия.

#### Подробное описание

Это раскрытие описывает различные технологии кодирования и/или сжатия видео. В конкретных примерах это раскрытие относится к кросс-компонентному предсказанию, которое используется для сокращения избыточности между компонентами. Оно может быть использовано в контексте усовершенствованных видеокодеков, таких как расширения HEVC, или в контексте следующего поколения стандартов кодирования видео.

Цветное видео играет важную роль в мультимедийных системах, где различные цветовые пространства используются для эффективного представления цвета. Цветовое пространство точно определяет цвет числовыми значениями с использованием нескольких компонент. Популярным цветовым пространством является цветовое пространство RGB, где цвет представлен как комбинация трех основных значений цветовых компонент (то есть, красным, зеленым и синим). Для сжатия цветного видео широко используется цветовое пространство YCbCr. См., например, A. Ford и A. Roberts, "Конверсии цветового пространства", Вестминстерский университет, Лондон, Tech. Rep., август 1998 г. Стандарты кодирования видео включают в себя ITU-T H.261, ISO/IEC MPEG-1 Визуальный, ITU-T H.262 или ISO/IEC MPEG-2 Визуальный, ITU-T H.263, ISO/IEC MPEG-4 Визуальный и ITU-T H.264 (также известный как ISO/IEC MPEG-4 AVC), включая его расширения: масштабируемое кодирование видео (SVC) и многовидовое кодирование видео (MVC).

Новый международный стандарт для кодирования видео, названный высокоэффективным кодированием видео (HEVC), был разработан и завершен объединенной командой по совместной деятельности в области кодирования видео (JCT-VC), группой экспертов по кодированию видео из группы экспертов по кодированию видео ITU-T (VCEG) и из группы экспертов по движущемуся изображению ISO/IEC (MPEG). Первая версия стандарта HEVC была завершена в апреле 2013 года. Вторая версия HEVC, включая расширения RExt, SHVC и MV-HEVC, была завершена в октябре 2014 года. Третья версия HEVC, включая расширение 3D-HEVC, была завершена в феврале 2015 года, в настоящее время действует и доступна по адресу <http://www.itu.int/rec/T-REC-H.265>. Это третье издание стандарта HEVC в дальнейшем в этом документе упоминается как стандарт HEVC. Второе и третье издания стандарта HEVC включают в себя расширение диапазона формата (RExt), которое расширяет кодирование видео для цветовых пространств, отличных от YCbCr 4:2:0, таких как YCbCr 4:2:2, YCbCr 4:4:4, и RGB 4:4:4.

YCbCr и RGB представляют собой два основных цветовых пространства, используемых для представления цифрового видео. RGB представляет цвет как комбинацию из красного, зеленого и синего, в то время как YCbCr представляет цифровое видео как яркость (Y или яркость), синий минус яркость (Cb, цветной синий или цветовой синий) и красный минус яркость (Cr, цветной красный или цветовой красный). YCbCr можно быть легко конвертировано из цветового пространства RGB посредством линейного преобразования. Избыточность между различными компонентами, а именно кросс-компонентная избыточность, значительно уменьшается в цветовом пространстве YCbCr. В дополнение, одним из преимуществ YCbCr является обратная совместимость с черно-белым телевидением (TV), так как сигнал Y передает информацию о яркости. Кроме того, ширина полосы цветности может быть уменьшена посредством подвыборки компонент Cb и Cr в формате выборки цветности 4:2:0 с менее значительным субъективным влиянием, чем подвыборка в RGB. Из-за этих преимуществ YCbCr является основным цветовым пространством в сжатии видео.

Существуют также другие цветовые пространства, такие как YCoCg, которые могут быть использованы при сжатии видео. YCoCg представляет цифровое видео как яркость (Y или яркость), оранжевый минус яркость (Co, цветной оранжевый или цветовой оранжевый) и зеленый минус яркость (Cg, цветной зеленый или цветовой зеленый). YCbCr можно быть легко конвертировано из цветового пространства RGB посредством линейного преобразования. В этом документе, независимо от фактического используемого цветового пространства, термины Y, Cb и Cr используются для представления трех цветовых компонент в каждой схеме сжатия видео. Следует понимать, что технологии могут применяться и к другим трем цветовым пространствам.

Фиг. 1 представляет собой блок-схему, иллюстрирующую примерную систему кодирования и деко-

дирования видео, которая может быть сконфигурирована или иначе приготовлена для реализации или другого использования одной или более технологий, описанных в этом раскрытии. Как показано на фиг. 1, система 10 включает в себя устройство-источник 12, которое генерирует кодированные видеоданные, подлежащие в дальнейшем декодированию устройством-адресатом 14. Устройство-источник 12 и устройство-адресат 14 могут содержать какие-либо из широкого ряда устройств, в том числе настольные компьютеры, блокнотные (то есть, переносные) компьютеры, планшетные компьютеры, телевизионные приставки, микротелефонные трубки, такие как так называемые "умные" телефоны, так называемые "умные" планшеты, телевизионные приемники, камеры, устройства отображения, цифровые мультимедийные проигрыватели, консоли для видеоигр, устройства потоковой передачи видео или подобное. В некоторых случаях, устройство-источник 12 и устройство-адресат 14 могут быть оборудованы для беспроводной связи.

Устройство-адресат 14 может принимать по линии 16 связи кодированные видеоданные, подлежащие декодированию. Линия 16 связи может содержать какой-либо тип носителя или устройства, способного перемещать кодированные видеоданные из устройства-источника 12 в устройство-адресат 14. В одном примере, линия 16 связи может содержать среду связи для обеспечения возможности устройству-источнику 12 передавать кодированные видеоданные непосредственно устройству-адресату 14 в реальном времени. Кодированные видеоданные могут быть модулированы согласно стандарту связи, такому как протокол беспроводной связи, и переданы устройству-адресату 14. Среда связи может содержать любую проводную или беспроводную среду связи, такую как спектр радиочастот (RF) или одну или более физических линий передач. Среда связи может формировать часть пакетной сети, такой как локальная сеть, широкомасштабная сеть или глобальная сеть, такая как Интернет. Среда связи может включать в себя маршрутизаторы, коммутаторы, базовые станции или другое оборудование, которое может быть пригодным для содействия связи от устройства-источника 12 к устройству-адресату 14.

В качестве альтернативы, кодированные данные могут выводиться из интерфейса 22 вывода в устройство 31 хранения. Аналогично, кодированные данные могут быть доступны из устройства 31 хранения интерфейсом ввода. Устройство 31 хранения может включать в себя любой из многообразия распределяемых или локально доступных носителей хранения данных, таких как жесткий диск, Blu-Ray-диски, DVD, CD-ROM, флэш-память, энергозависимая или энергонезависимая память, либо любые другие подходящие цифровые носители хранения данных для хранения кодированных видеоданных. В дополнительном примере, устройство 31 хранения может соответствовать файловому серверу или другому промежуточному устройству хранения, которое может удерживать кодированное видео, сгенерированное посредством устройства-источника 12. Устройство-адресат 14 может осуществлять доступ к сохраненным видеоданным из устройства 31 хранения через потоковую передачу или загрузку. Файловый сервер может быть любым типом сервера, способным хранить кодированные видеоданные и передавать эти кодированные видеоданные устройству-адресату 14. Примерные файловые серверы включают в себя веб-сервер (например, для веб-сайта), сервер FTP, устройства хранения данных, подключаемые к сети (NAS), или локальные диски. Устройство-адресат 14 может осуществлять доступ к закодированным видеоданным через любое стандартное соединение для передачи данных, включающее в себя Интернет-соединение. Это может включать в себя беспроводный канал (например, Wi-Fi соединение), проводное соединение (например, DSL, кабельный модем и так далее) или их комбинацию, которая является подходящей для осуществления доступа к кодированному видеоданным, хранимым на файловом сервере. Передача кодированных видеоданных от устройства 31 хранения может быть потоковой передачей, передачей загрузки или их комбинацией.

Технологии этого раскрытия не ограничены беспроводными приложениями или установками. Технологии могут быть применены к кодированию видео в поддержку какого-либо из многообразия мультимедийных приложений, таких как телевизионные вещательные передачи по радиointерфейсу, кабельные телевизионные передачи, спутниковые телевизионные передачи, потоковые передачи видео, например, через Интернет, кодирование цифрового видео для хранения на запоминающем носителе, декодирование цифрового видео, сохраненного на запоминающем носителе, или другие приложения. В некоторых примерах, система 10 может быть сконфигурирована с возможностью поддержки односторонней или двухсторонней передачи видео для поддержки приложений, таких как потоковая передача видео, воспроизведение видео, вещание видео и/или видеотелефония.

В примере с фиг. 1, устройство-источник 12 включает в себя источник 18 видео, видеокодер 20 и интерфейс 22 вывода. В некоторых случаях, интерфейс 22 вывода может включать в себя модулятор/демодулятор (модем) и/или передатчик. В устройстве-источнике 12, источник 18 видео может включать в себя источник, такой как устройство видеозахвата, например, видеокамеру, видеоархив, содержащий ранее захваченное видео, интерфейс подачи видео для приема видео от поставщика видеоконтента и/или систему компьютерной графики для генерирования данных компьютерной графики в качестве исходного видео или комбинацию из таких источников. В качестве одного примера, если источником 18 видео является видеокамера, устройство-источник 12 и устройство-адресат 14 могут формировать так называемые телефоны с камерами или видео-телефоны. Однако технологии, описанные в этом раскрытии, могут быть применимы к кодированию видео в общем и могут быть применены к беспроводным

и/или проводным приложениям.

Захваченное, предварительно захваченное или сгенерированное компьютером видео может быть кодировано посредством видеокодера 20. Кодированные видеоданные могут быть переданы непосредственно устройству-адресату 14 через интерфейс 22 вывода устройства-источника 12. Кодированные видеоданные также могут быть (или в качестве альтернативы) сохранены на устройство 31 хранения для более позднего доступа устройством-адресата 14 или другими устройствами, для декодирования и/или воспроизведения.

Устройство-адресат 14 включает в себя интерфейс 28 ввода, видеodeкодер 30 и устройство 32 отображения. В некоторых случаях, интерфейс 28 ввода может включать в себя приемник и/или модем. Интерфейс 28 ввода устройства-адресата 14 принимает кодированные видеоданные по линии 16 связи. Кодированные видеоданные, связанные по линии 16 связи или обеспеченные на устройстве 31 хранения, могут включать в себя многообразие синтаксических элементов, сгенерированных посредством видеокодера 20 для использования видеodeкодером, таким как видеodeкодер 30, в декодировании видеоданных. Такие синтаксические элементы могут быть включены в закодированные видеоданные, передаваемые по среде связи, хранимые на запоминающем носителе, или хранимые на файловом сервере.

Устройство 32 отображения может быть объединено с, или быть внешним относительно устройства-адресата 14. В некоторых примерах устройство-адресат 14 может включать в себя интегрированное устройство отображения и также быть сконфигурировано с возможностью взаимодействия с внешним устройством отображения. В других примерах устройство-адресат 14 может быть устройством отображения. В общем, устройство 32 отображения отображает декодированные видеоданные пользователю и может содержать любое из многообразия устройств отображения, таких как жидкокристаллическое устройство отображения (LCD), плазменное устройство отображения, устройство отображения на органических светодиодах (OLED) или другой тип устройства отображения.

Видеокодер 20 и видеodeкодер 30 могут работать согласно стандарту сжатия видео, такому как упоминаемый выше по тексту стандарт высокоэффективного кодирования видео (HEVC), и могут соответствовать тестовой модели HEVC (HM). В качестве альтернативы, видеокодер 20 и видеodeкодер 30 могут функционировать согласно другим частным или промышленным стандартам, таким как стандарт ITU-T H.264, в качестве альтернативы упоминаемый как MPEG4, часть 10, усовершенствованное кодирование видео (AVC) или расширения таких стандартов. Технологии этого раскрытия, однако, не ограничены каким-либо конкретным стандартом кодирования. Другие примеры стандартов сжатия видео включают в себя MPEG-2 и ITU-T H.263.

Хоть это и не показано на фиг. 1, в некоторых аспектах, видеокодер 20 и видеodeкодер 30, каждый, может быть объединен с аудиокодером и аудиodeкодером, и может включать в себя надлежащие блоки уплотнения-разуплотнения (MUX-DEMUX), или другое аппаратное и программное обеспечение, чтобы управлять кодированием как аудио, так и видео, в общем потоке данных или в отдельных потоках данных. Если применимо, в некоторых примерах, блоки MUX-DEMUX могут соответствовать протоколу мультиплекса ITU H.223 или другим протоколам, таким как протокол пользовательских дейтаграмм (UDP).

Каждый из видеокодера 20 и видеodeкодера 30 может быть реализован как любая из многообразия подходящих схем кодера, таких как схема, имеющая один или более процессоров, микропроцессоров, цифровых сигнальных процессоров (DSP), специализированная интегральная схема (ASIC), программируемая пользователем вентильная матрица (FPGA), дискретная логика, программное обеспечение, аппаратное обеспечение, программно-аппаратное обеспечение или любые их комбинации. Когда технологии реализуются частично в программном обеспечении, устройство может хранить инструкции для программного обеспечения на подходящем, долговременном компьютерно-читаемом носителе и исполнять эти инструкции в аппаратном обеспечении с использованием одного или более процессоров для выполнения технологий этого раскрытия. Каждый из видеокодера 20 и видеodeкодера 30 может быть включен в один или более кодеров или декодеров, любой из которых может быть интегрирован как часть комбинированного кодера/декодера (кодека) в соответствующем устройстве.

В общем, рабочая модель HM описывает, что видеокادر или изображение могут быть разделены на последовательность древовидных блоков или наибольших единиц кодирования (LCU), которые включают в себя как выборки яркости, так и выборки цветности. Древовидный блок имеет аналогичную цель как и макроблок стандарта H.264. Слайс включает в себя некоторое число последовательных древовидных блоков в порядке кодирования. Видеокادر или изображение могут быть секционированы на один или более слайсов. Каждый древовидный блок может быть разбит на единицы кодирования (CU) согласно квадродереву. Например, древовидный блок, в качестве корневого узла квадродерева, может быть разбит на четыре дочерних узла, и каждый дочерний узел может, в свою очередь, быть родительским узлом и быть разбит на другие четыре дочерних узла. Конечный, не разбитый дочерний узел, в качестве конечного узла квадродерева, содержит узел кодирования, то есть, кодированный видеоблок. Синтаксические данные, ассоциированные с кодированным битовым потоком, могут задавать максимальное количество раз, когда древовидный блок может быть разбит, а также могут задавать минимальный размер узлов кодирования.

В одном примерном подходе CU может включать в себя блок кодирования яркости и два блока кодирования цветности. В другом примерном подходе CU может включать в себя блок кодирования яркости, блок кодирования Co и блок кодирования Cg в цветовом пространстве YCoCg. В другом примерном подходе CU может включать в себя блок кодирования красного, блок кодирования зеленого и блок кодирования синего в цветовом пространстве RGB.

В одном примерном подходе CU может иметь ассоциированные единицы предсказания (PU) и единицы преобразования (TU). Каждая из PU может включать в себя один блок предсказания яркости и два блока предсказания цветности, и каждая из TU может включать в себя один блок преобразования яркости и два блока преобразования цветности. Каждый из блоков кодирования может быть секционирован на один или более блоков предсказания, которые содержат блоки для выборок, к которым применяется такое же предсказание. Каждый из блоков кодирования также может быть секционирован на один или более блоков преобразования, которые содержат блоки выборок, к которым применяется такое же преобразование.

Размер CU в общем соответствует размеру узла кодирования и типично должен быть квадратным по форме. Размер CU может находиться в диапазоне от 8×8 пикселей вплоть до размера древовидного блока с максимумом в 64×64 пикселей или больше. Каждая CU может задавать одну или более PU и одну или более TU. Синтаксические данные, включенные в CU, могут описывать, например, секционирование блока кодирования на один или более блоков предсказания. Режимы секционирования могут различаться между тем, закодирована ли CU в прямом режиме или режиме пропуска, закодирована в режиме внутреннего предсказания или закодирована в режиме внешнего предсказания. Блоки предсказания могут быть секционированы, чтобы быть квадратными или не квадратными по форме. Синтаксические данные, включенные в CU, также могут описывать, например, секционирование блока кодирования на один или более блоков преобразования согласно квадродереву. Блоки преобразования могут быть секционированы так, чтобы быть квадратными или не квадратными по форме.

Стандарт HEVC позволяет преобразования согласно TU, которые могут различаться для различных CU. TU типично установлены по размеру на основании размера PU в пределах данной CU, заданной для секционированной LCU, хотя это может не всегда иметь место. TU, как правило, имеют тот же размер или меньше, чем PU. В некоторых примерах, остаточные выборки, соответствующие CU, могут быть подразделены на более маленькие единицы с использованием структуры квадродерева, известной как "остаточное квадродерево" (RQT). Концевые узлы RQT могут представлять TU. Значения пиксельных разностей, ассоциированные с TU, могут быть преобразованы для создания коэффициентов преобразования, которые могут быть квантованы.

В общем, PU включает в себя данные, связанные с процессом предсказания. Например, когда PU закодирована во внутреннем режиме, PU может включать в себя данные, описывающие режим внутреннего предсказания для PU. В качестве другого примера, когда PU закодирована во внешнем режиме, PU может включать в себя данные, задающие вектор движения для PU. Данные, задающие вектор движения для PU могут описывать, например, горизонтальную компоненту вектора движения, вертикальную компоненту вектора движения, разрешение для вектора движения (например, одна четвертая пиксельной точности или одна восьмая пиксельной точности), опорное изображение, на которое указывает вектор движения и/или список опорных изображений (например, список 0, список 1 или список C) для вектора движения.

В общем, TU используется для процессов преобразования и квантования. Данная CU, имеющая одну или более PU, также может включать в себя одну или более TU. После предсказания видеокодер 20 может вычислять остаточные значения из видеоблока, идентифицированного узлом кодирования, в соответствии с PU. Затем узел кодирования обновляется для обращения к остаточным значениям, а не к исходному видеоблоку. Остаточные значения содержат значения пиксельных разностей, которые могут быть преобразованы в коэффициенты преобразования, квантованы и сканированы с использованием преобразований и другой информации преобразования, точно определяемой в TU, для создания упорядоченных коэффициентов преобразования для энтропийного кодирования. Узел кодирования может быть снова обновлен, чтобы обращаться к этим упорядоченным коэффициентам преобразования. Это раскрытие типично использует термин "видеоблок" для обращения к узлу кодирования CU. В некоторых особых случаях, это раскрытие также может использовать термин "видеоблок" для обращения к древовидному блоку, то есть, LCU или CU, который включает в себя узел кодирования, а также PU и TU.

Видеопоследовательность типично включает в себя ряд видеок кадров или изображений. Группа изображений (GOP), в общем, содержит ряд из одного или более изображений видео. GOP может включать в себя синтаксические данные в заголовке GOP, заголовке одного или более из изображений, или в другом месте, которые описывают число изображений, включенных в GOP. Каждый слайс изображения может включать в себя синтаксические данные слайса, которые описывают режим кодирования для соответственного слайса. Видеокодер 20 типично работает над видеоблоками, в пределах индивидуальных слайсов видео, для того, чтобы кодировать видеоданные. Видеоблок может соответствовать узлу кодирования в пределах CU. Видеоблоки могут иметь фиксированные или варьирующиеся размеры, и могут

отличаться по размеру в соответствии с точно определенным стандартом кодирования.

В качестве примера, НМ поддерживает предсказание в различных размерах PU. Предположим, что размер конкретной CU равен  $2N \times 2N$ , тогда НМ поддерживает внутреннее предсказание в размерах PU  $2N \times 2N$  или  $N \times N$ , и внешнее предсказание в симметричных размерах PU  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$  или  $N \times N$ . НМ также поддерживает асимметричное секционирование для внешнего предсказания в размерах PU  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  и  $nR \times 2N$ . При асимметричном секционировании, одно направление CU не секционируется, в то время как другое направление секционируется на 25 и 75%. Участок CU, соответствующий 25%-ому участку, обозначается "n", за чем следует обозначение "Up" (вверх), "Down" (вниз), "Left" (налево) или "Right" (направо). Таким образом, например, " $2N \times nU$ " относится к CU  $2N \times 2N$ , которая секционирована по горизонтали с PU  $2N \times 0,5N$  сверху и PU  $2N \times 1,5N$  внизу.

В настоящем раскрытии, " $N \times N$ " и " $N$  на  $N$ " может использоваться взаимозаменяемо для обращения к размерностям пикселей видеоблока в значениях вертикальной и горизонтальной размерностей, например,  $16 \times 16$  пикселей или 16 на 16 пикселей. В целом,  $16 \times 16$  блок будет иметь 16 пикселей в вертикальном направлении ( $y=16$ ) и 16 пикселей в горизонтальном направлении ( $x=16$ ). Подобным образом, блок  $N \times N$ , в общем, имеет  $N$  пикселей в вертикальном направлении и  $N$  пикселей в горизонтальном направлении, где  $N$  представляет собой неотрицательное целое значение. Пиксели в блоке могут быть расположены по строкам и столбцам. Более того, блоки не обязательно должны иметь такое же количество пикселей в горизонтальном направлении, как и в вертикальном направлении. Например, блоки могут содержать  $N \times M$  пикселей, где  $M$  не обязательно равно  $N$ .

Вслед за кодированием с внутренним предсказанием или внешним предсказанием с использованием PU из CU, видеокодер 20 может вычислять остаточные данные, к которым применяются преобразования, точно определенные TU из CU. Остаточные данные могут соответствовать пиксельным разностям между пикселями

незакодированного изображения и значениями предсказания, соответствующими CU. Видеокодер 20 может формировать остаточные данные для CU, а затем преобразовывать остаточные данные, чтобы создавать коэффициенты преобразования.

Следом за любыми преобразованиями для создания коэффициентов преобразования, видеокодер 20 может выполнять квантование коэффициентов преобразования. Под квантованием, в общем, имеется в виду процесс, в котором коэффициенты преобразования квантуются, чтобы можно было уменьшить объем данных, используемых для представления коэффициентов, обеспечивая дополнительное сжатие. Процесс квантования может уменьшать битовую глубину, ассоциированную с некоторыми или всеми коэффициентами. Например,  $n$ -битовое значение может быть округлено в меньшую сторону до  $m$ -битового значения в ходе квантования, при этом  $n$  больше  $m$ .

В некоторых примерах, видеокодер 20 может использовать предварительно заданный порядок сканирования для сканирования квантованных коэффициентов преобразования для создания упорядоченного вектора, который может быть энтропийно кодирован. В других примерах, видеокодер 20 может выполнять адаптивное сканирование. После сканирования квантованных коэффициентов преобразования для формирования одномерного вектора, видеокодер 20 может энтропийно кодировать одномерный вектор, например, согласно контекстно-адаптивному кодированию с переменной длиной кодового слова (CAVLC), контекстно-адаптивному двоичному арифметическому кодированию (CABAC), синтаксическому контекстно-адаптивному двоичному арифметическому кодированию (SBAC), энтропийному кодированию на основе секционирования на интервалы вероятности (PIPE) или другой методологии энтропийного кодирования. Видеокодер 20 также может энтропийно кодировать синтаксические элементы, ассоциированные с закодированными видеоданными, для использования видеодекодером 30 при декодировании видеоданных.

Для выполнения CABAC, видеокодер 20 может назначать контекст в пределах контекстной модели символу, который должен быть передан. Контекст может относиться к, например, тому, являются ли соседние значения символа ненулевыми или нет. Для выполнения CAVLC видеокодер 20 может выбирать код с переменной длиной для символа, который должен быть передан. Кодовые слова в VLC могут быть построены так, что относительно более короткие коды соответствуют наиболее вероятным символам, в то время как более длинные коды соответствуют менее вероятным символам. Таким образом, использование VLC может достичь сохранений битов за счет, например, использования равно-длинных кодовых слов для каждого символа, который должен быть передан. Определение вероятности может быть основано на контексте, назначенном символу.

Фиг. 2 представляет собой блок-схему, иллюстрирующую примерный видеокодер, который может быть сконфигурирован или иначе подготовлен для реализации или другого использования одной или более технологий, описанных в этом раскрытии. Видеокодер 20 может выполнять внутренне и внешнее кодирование видеоблоков внутри слайсов видео. Внутреннее кодирование полагается на пространственное предсказание для сокращения или удаления пространственной избыточности видео в пределах данного видеокadra или изображения. Внешнее кодирование полагается на временное предсказание, чтобы сокращать или удалять временную избыточность в видео в пределах смежных кадров или изображений

видеопоследовательности. Внутренний режим (I режим) может относиться к любому или нескольким режимам кодирования на основании пространственного предсказания. Внешние режимы, как, например, однонаправленное предсказание (P режим) или двунаправленное предсказание (B режим), могут относиться к любому из нескольких режимов кодирования на основании временного предсказания.

Как показано на фиг. 2, видеокодер 20 принимает текущий видеоблок в видеокадре, который должен быть кодирован. На примере с фиг. 2 видеокодер 20 включает в себя память 38 видеоданных, блок 40 обработки предсказания, память 64 опорных изображений, сумматор 50, блок 52 обработки преобразования, блок 54 квантования и блок 56 энтропийного кодирования. Блок 40 обработки предсказания, в свою очередь, включает в себя блок 44 компенсации движения, блок 42 оценки движения и блок 46 внутреннего предсказания, и блок 48 секционирования. Для восстановления видеоблока, видеокодер 20 также включает в себя блок 58 обратного квантования, блок 60 обратного преобразования и сумматор 62. Деблокирующий фильтр (не показан на фиг. 2) также может быть включен в границы блока фильтров для удаления артефактов блочности из восстановленного видео. При необходимости, деблокирующий фильтр, типично, будет фильтровать выходные данные сумматора 62. Дополнительные фильтры (в цикле или пост-цикле) также могут быть использованы в дополнение к деблокирующему фильтру. Такие фильтры не изображены для краткости, но, если желательно, то могут фильтровать выходные данные сумматора 62 (как внутри-циклический фильтр).

Во время процесса кодирования видеокодер 20 принимает видеокадр или слайс, подлежащий кодированию, и сохраняет его в памяти 38 видеоданных. Кадр или слайс может быть разделен на несколько видеоблоков посредством блока 40 обработки предсказания. Блок 42 оценки движения и блок 44 компенсации движения выполняют кодирование с внешним предсказанием принятого видеоблока по отношению к одному или нескольким блокам в одном или нескольких опорных кадрах для обеспечения временного предсказания. Блок 46 внутреннего предсказания может альтернативно выполнять кодирование с внутренним предсказанием принятого видеоблока по отношению к одному или нескольким соседним блокам в том же кадре или слайсе, что и блок, который должен быть кодирован, для обеспечения пространственного предсказания. Видеокодер 20 может выполнять несколько проходов кодирования, например, для выбора надлежащего режима кодирования для каждого блока видеоданных.

Более того, блок 48 секционирования может секционировать блоки видеоданных на субблоки, основанные на оценке предыдущих схем секционирования в предыдущих проходах кодирования. Например, блок 48 секционирования может исходно секционировать кадр или слайс на LCU, и секционировать каждую из LCU на суб-CU на основании анализа отношения скорости к искажению (например, оптимизации отношения скорости к искажению). Блок 40 обработки предсказания может дополнительно создавать структуру данных квадродерева, указывающую секционирование LCU на суб-CU. Концевой узел CU квадродерева может включать в себя одну или более PU и одну или более TU.

Блок 40 обработки предсказания может выбирать один из режимов кодирования, внутренний или внешний, например, на основании результатов ошибки, и обеспечивать результирующий блок с внутренним или внешним кодированием сумматору 50 для генерирования остаточного блока данных и сумматору 62 для восстановления закодированного блока для использования в качестве опорного кадра. Блок 40 обработки предсказания также обеспечивает синтаксические элементы, такие как векторы движения, указатели внутреннего режима, информацию секционирования и другую такую синтаксическую информацию, блоку 56 энтропийного кодирования. Блок 40 обработки предсказания может выбирать один или несколько внешних режимов с использованием анализа скорость-искажение.

Блок 42 оценки движения и блок 44 компенсации движения могут быть высоко интегрированы, но проиллюстрированы по отдельности для концептуальных целей. Оценка движения, выполняемая блоком 42 оценки движения, является процессом генерирования векторов движения, которые оценивают движение для видеоблоков. Вектор движения, например, может указывать перемещение PU видеоблока в текущем видеокадре или изображении по отношению к предсказанному блоку в опорном кадре (или другой кодированной единице) относительно текущего блока, который кодируется в текущем кадре (или другой кодированной единице). Предсказанный блок является блоком, в отношении которого установлено, что он близко подходит к блоку, который должен быть закодирован, в плане пиксельной разности, которая может быть определена посредством суммы абсолютных разностей (SAD), суммы квадратичных разностей (SSD), или другой метрики разности. В некоторых примерах, видеокодер 20 может вычислять значения для субцелочисленных пиксельных позиций опорных изображений, сохраненных в памяти 64 опорных изображений. Например, видеокодер 20 может интерполировать значения одной четвертой пиксельных позиций, одной восьмой пиксельных позиций, или других дробных пиксельных позиций опорного изображения. А значит, блок 42 оценки движения может выполнять поиск движения относительно полных пиксельных позиций и дробных пиксельных позиций и выводить вектор движения с точностью до дробного пикселя.

Блок 42 оценки движения вычисляет вектор движения для PU видеоблока во внешне-кодированном слайсе, сравнивая позицию PU с позицией предсказанного блока опорного изображения. Опорное изображение может быть выбрано из первого списка (список 0) опорных изображений или второго списка (список 1) опорных изображений, каждый из которых идентифицирует одно или более опорных изобра-

жений, сохраненных в памяти 64 опорных изображений. Блок 42 оценки движения отправляет вычисленный вектор движения блоку 56 энтропийного кодирования и блоку 44 компенсации движения.

Компенсация движения, выполненная посредством блока 44 компенсации движения, может вовлечь в себя выборку или генерирование предсказанного блока на основе вектора движения, определенно посредством блока 42 оценки движения. Опять же, в некоторых примерах блок 42 оценки движения и блок 44 компенсации движения могут быть функционально интегрированы. После приема вектора движения для PU текущего видеоблока, блок 44 компенсации движения может располагать предсказанный блок, на который указывает вектор движения, в одном из списков опорных изображений. Сумматор 50 формирует остаточный видеоблок посредством вычитания значений пикселя предсказанного блока из значений пикселя текущего видеоблока, который кодируется, формируя значения пиксельной разности, как рассматривается ниже по тексту. В общем, блок 42 оценки движения выполняет оценку движения относительно блоков кодирования яркости, а блок 44 компенсации движения использует векторы движения, вычисленные на основании блоков кодирования яркости, для блоков кодирования цветности и блоков кодирования яркости. Блок 40 обработки предсказания может также генерировать синтаксические элементы, ассоциированные с видеоблоками и слайсом видео для использования видеодекодером 30 при декодировании видеоблоков слайса видео.

Блок 46 внутреннего предсказания может выполнять внутреннее предсказание текущего блока, в качестве альтернативы внешнему предсказанию, которое выполняется блоком 42 оценки движения и блоком 44 компенсации движения, как описано выше по тексту. В частности, блок 46 внутреннего предсказания может определять режим внутреннего предсказания для использования при кодировании текущего блока. В некоторых примерах, блок 46 внутреннего предсказания может кодировать текущий блок с использованием различных режимов внутреннего предсказания, например, во время отдельных проходов кодирования, и блок 46 внутреннего предсказания (или блок 40 обработки предсказания, в некоторых примерах) может выбирать подходящий режим внутреннего предсказания для использования из протестированных режимов.

Например, блок 46 внутреннего предсказания может вычислять значения скорость-искажение с использованием анализа скорость-искажение для различных тестируемых режимов внутреннего предсказания, и выбирать режим внутреннего предсказания, имеющий лучшие характеристики скорость-искажение среди тестируемых режимов. Анализ скорость-искажение в общем определяет объем искажения (или ошибку) между закодированным блоком и исходным, незакодированным блоком, который был закодирован для создания закодированного блока, а также скорость прохождения битов (то есть, количество битов), используемую для создания закодированного блока. Блок 46 внутреннего предсказания может вычислять отношения из искажений и скоростей для различных закодированных блоков, чтобы определить, какой режим внутреннего предсказания показывает наилучшее значение скорость-искажение для блока.

После выбора режима внутреннего предсказания для блока, блок 46 внутреннего предсказания может обеспечивать информацию, указывающую выбранный режим внутреннего предсказания для блока, блоку 56 энтропийного кодирования. Блок 56 энтропийного кодирования может кодировать информацию, указывающую выбранный режим внутреннего предсказания. Видеокодер 20 может включать в передаваемый битовый поток данные конфигурации, которые могут включать в себя множество индексных таблиц режима внутреннего предсказания и множество модифицированных индексных таблиц режима внутреннего предсказания (также называемых таблицами отображения кодовых слов), определения контекстов кодирования для различных блоков и указания наиболее вероятного режима внутреннего предсказания, индексную таблицу режима внутреннего предсказания и модифицированную индексную таблицу режима внутреннего предсказания для использования для каждого из контекстов.

Видеокодер 20 формирует остаточный видеоблок посредством вычитания данных предсказания от блока 40 выбора режима из исходного видеоблока, который кодируется. Сумматор 50 представляет компонент или компоненты, которые выполняют эту операцию вычитания. Блок 52 обработки преобразования применяет преобразование, такое как дискретное косинусное преобразование (DCT) или концептуально подобное преобразование, к остаточному блоку, создавая видеоблок, содержащий остаточные значения коэффициентов преобразования. Блок 52 обработки преобразования может выполнять другие преобразования, которые концептуально подобны DCT. Также могут быть использованы вейвлет-преобразования, целочисленные преобразования, суб-полосные преобразования или другие типы преобразований. В любом случае, блок 52 обработки преобразования применяет преобразование к остаточному блоку, создавая блок из остаточных коэффициентов преобразования. Преобразование может конвертировать остаточную информацию из области пиксельных значений в область преобразования, такую как частотная область. Блок 52 обработки преобразования может отправлять получившиеся в результате коэффициенты преобразования блоку 54 квантования. Блок 54 квантования квантует коэффициенты преобразования, чтобы дополнительно уменьшить скорость передачи битов. Процесс квантования может уменьшать битовую глубину, ассоциированную с некоторыми или всеми коэффициентами. Степень квантования может быть модифицирована посредством корректировки параметра квантования. В некоторых примерах блок 54 квантования затем может выполнять сканирование матрицы, включающей в

себя квантованные коэффициенты преобразования. В качестве альтернативы сканирование может выполнять блок 56 энтропийного кодирования.

Вслед за квантованием блок 56 энтропийного кодирования энтропийно кодирует квантованные коэффициенты преобразования. Например, блок 56 энтропийного кодирования может выполнять контекстно-адаптивное кодирование с переменной длиной кодового слова (CAVLC), контекстно-адаптивное двоичное арифметическое кодирование (CABAC), основанное на синтаксисе контекстно-адаптивное двоичное арифметическое кодирование (SBAC), энтропийное кодирование с секционированием интервала вероятности (PIPE) или другую технологию энтропийного кодирования. В случае с основанным на контексте энтропийным кодированием контекст может быть основан на соседних блоках. Вслед за энтропийным кодированием блоком 56 энтропийного кодирования закодированный битовый поток может быть передан другому устройству (например, видеодекодеру 30) или архивирован для дальнейшей передачи или извлечения.

Блок 58 обратного квантования и блок 60 обратного преобразования применяют обратное квантование и обратное преобразование, соответственно, для восстановления остаточного блока в пиксельной области, например, для последующего использования в качестве опорного блока. Блок 44 компенсации движения может вычислять опорный блок посредством добавления остаточного блока к предсказанному блоку одного из кадров из памяти 64 опорных изображений. Блок 44 компенсации движения также может применять один или более интерполяционных фильтров к восстановленному остаточному блоку для вычисления субцелочисленных пиксельных значений для использования в оценке движения. Сумматор 62 добавляет восстановленный остаточный блок к блоку предсказания после компенсации движения, созданному блоком 44 компенсации движения, для создания восстановленного видеоблока для хранения в памяти 54 опорных изображений. Восстановленный видеоблок может быть использован блоком 42 оценки движения и блоком 44 компенсации движения в качестве опорного блока для внешнего кодирования блока в последующем видеокадре.

Фиг. 3 представляет собой блок-схему, иллюстрирующую примерный видеодекодер, который может быть сконфигурирован или иначе подготовлен для реализации или другого использования одной или более технологий, описанных в этом раскрытии. В примере с фиг. 3, видеодекодер 30 включает в себя память 68 видеоданных, блок 70 энтропийного декодирования, блок 72 компенсации движения, блок 74 внутреннего предсказания, блок 76 обратного квантования, блок 78 обратного преобразования, сумматор 80 и память 82 опорных изображений. В примере с фиг. 3, видеодекодер 30 включает в себя блок 71 предсказания, который, в свою очередь, включает в себя блок 72 компенсации движения и блок 74 внутреннего предсказания. Видеодекодер 30 может, в некоторых примерах, выполнять проход декодирования в общем обратный проход кодирования, описанному относительно видеокодера 20 (Фиг. 2). Блок 72 компенсации движения может генерировать данные предсказания на основании векторов движения, принятых от блока 70 энтропийного декодирования, в то время как блок 74 внутреннего предсказания может генерировать данные предсказания на основании указателей режима внутреннего предсказания, принятых от блока 70 энтропийного декодирования.

Во время процесса декодирования память 68 видеоданных видеодекодера 30 принимает закодированный битовый видеопоток и сохраняет закодированный битовый видеопоток в памяти 68 видеоданных. Закодированный битовый видеопоток представляет собой видеоблоки кодированного слайса видео и ассоциированные синтаксические элементы, переданные из видеокодера 20. Блок 70 энтропийного декодирования энтропийно декодирует битовый поток, чтобы сгенерировать квантованные коэффициенты, векторы движения или указатели режима внутреннего предсказания, и другие синтаксические элементы. Блок 70 энтропийного декодирования пересылает векторы движения и другие синтаксические элементы блоку 72 компенсации движения. Видеодекодер 30 может принимать синтаксические элементы на уровне слайса видео и/или уровне видеоблока.

Когда слайс видео закодирован как внутренне-кодированный (I) слайс, то блок 74 внутреннего предсказания может генерировать данные предсказания для видеоблока текущего слайса видео на основании сигнализированного режима внутреннего предсказания и данных от ранее декодированных блоков текущего кадра или изображения. Когда видеокадр закодирован как внешне-кодированный (то есть, кадры B, P или GPB) слайс, то блок 72 компенсации движения создает предсказанные блоки для видеоблока текущего слайса видео на основании векторов движения и других синтаксических элементов, принятых от блока 70 энтропийного декодирования. Предсказанные слайсы (слайсы P) и обобщенные слайсы P и B (слайсы GPB) могут использовать данные из предыдущих кадров, чтобы задавать слайс, тогда как двусторонний предсказанный слайс (слайс B) может использовать как предыдущий, так и передний кадр, чтобы задавать слайс. В некоторых примерных подходах предсказанные блоки могут быть созданы из одного из опорных изображений в пределах одного из списков опорных изображений. Видеодекодер 30 может построить списки опорных кадров, список 0 и список 1, с использованием технологий построения по умолчанию на основании опорных изображений, сохраненных в памяти 82 опорных изображений.

Блок 72 компенсации движения определяет информацию предсказания для видеоблока текущего слайса видео посредством синтаксического анализа векторов движения и других синтаксических элементов, и использует информацию предсказания для создания предсказанных блоков для текущего видео-

блока, который кодируется. Например, блок 72 компенсации движения использует некоторые из принятых синтаксических элементов для определения режима предсказания (например, внутреннего или внешнего предсказания), используемого для кодирования видеоблоков слайса видео, типа слайса с внешним предсказанием (например, слайса В, слайса Р или слайса GРВ), информации построения для одного или нескольких списков опорных изображений для слайса, векторов движения для каждого внешне кодированного видеоблока слайса, статуса внешнего предсказания для каждого внешне кодированного видеоблока слайса, и другой информации для декодирования видеоблоков в текущем видеослайсе.

Блок 72 компенсации движения также может выполнять интерполяцию на основе интерполяционных фильтров. Блок 72 компенсации движения может использовать интерполяционные фильтры так же, как и видеокодер 20 во время кодирования видеоблоков, для вычисления интерполированных значений для субцелочисленных пикселей опорных блоков. В таком случае, блок 72 компенсации движения может определять интерполяционные фильтры, использованные видеокодером 20, из принятых синтаксических элементов, и использовать интерполяционные фильтры для создания предсказанных блоков.

Блок 76 обратного квантования обратное квантует, то есть, деквантует, квантованные коэффициенты преобразования, обеспеченные в битовом потоке и декодированные блоком 70 энтропийного декодирования. Процесс обратного квантования может включать в себя использование параметра QPУ квантования, вычисленного посредством видеодекодера 30 для каждого видеоблока в слайсе видео для определения степени квантования и, аналогично, степени обратного квантования, которые должны быть применены.

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

После того как блок 72 компенсации движения генерирует предсказанный блок для текущего видеоблока на основе векторов движения и других синтаксических элементов, видеодекодер 30 формирует декодированный видеоблок посредством суммирования остаточных блоков из блока 78 обратного преобразования с соответствующими предсказанными блоками, сгенерированными блоком 72 компенсации движения. Сумматор 80 представляет компонент или компоненты, которые выполняют эту операцию суммирования. Если желательно, деблокирующий фильтр также может быть применен для фильтрации декодированных блоков для того, чтобы удалять артефакты блочности. Другие циклические фильтры (либо в цикле кодирования, либо после цикла кодирования) также могут быть использованы для сглаживания пиксельных переходов, или, иначе, для улучшения качества видео. Декодированные видеоблоки в данном кадре или изображении затем сохраняются в памяти 82 опорных изображений, которая хранит опорные изображения, используемые для последующей компенсации движения. Память 82 опорных изображений также хранит декодированное видео для более позднего представления на устройстве отображения, таком как устройство 32 отображения с фиг. 1.

Ниже будет рассмотрено адаптивное кросс-компонентное предсказание остатка.

Хотя кросс-компонентная избыточность значительно уменьшается в цветовом пространстве YCbCr, корреляция между тремя цветовыми компонентами может все еще существовать. Различные способы были изучены для улучшения производительности кодирования видео посредством дополнительного уменьшения корреляции, особенно для кодирования видео в формате цветности 4:4:4. В одном подходе масштабный коэффициент и смещение используются для каждого блока для предсказания сигнала цветности из восстановленного сигнала яркости. (См. В. С. Song, Y. G. Lee и N. H. Kim, "Алгоритм межцветовой компенсации с адаптацией по блокам для кодирования видео RGB 4:4:4", "IEEE протоколы в отношении схем и систем для видео технологий", том 18, № 10, стр. 1447-1451, октябрь 2008 г. предложение LM.).

В другом подходе в остаточной области выполняется технология, называемая кросс-компонентным предсказанием (ССР). (См. W. Pu, W.-S. Kim, J. Chen, J. Sole, M. Karchewicz, "RCE1: описания и результаты экспериментов 1, 2, 3 и 4", Объединенная команда по совместной деятельности в области кодирования видео (JCT-VC) ITU-T SG16 WP3 и ISO/IEC JTC1/SC29/WG11, JCTVC-O0202, Женева, ноябрь 2013 г.).

В ССР остаточный сигнал цветности предсказывается с использованием восстановленного остаточного сигнала яркости на стороне кодера как

$$\Delta r_c(x, y) = r_c(x, y) - (\alpha \times r'_L(x, y)) \gg 3 \quad (1)$$

и это компенсируется на стороне декодера как

$$r'_c(x, y) = \Delta r'_c(x, y) + (\alpha \times r'_L(x, y)) \gg 3 \quad (2)$$

где  $r_c(x, y)$  и  $r'_c(x, y)$  обозначают исходные и восстановленные остаточные выборки цветности в позиции  $(x, y)$ . В ССР,  $\Delta r_c(x, y)$  и  $\Delta r'_c(x, y)$  обозначают конечные остаточные выборки цветности после кросс-компонентного предсказания, в то время как  $r'_L(x, y)$  представляет значение восстановленных остаточных выборок яркости. В некоторых примерах весовой коэффициент  $\alpha$  явно сигнализируется в битовом

вый поток для каждой единицы преобразования цветности в HEVC. Этот способ кросс-компонентного предсказания остатка был принят в формате и расширении диапазона стандарта HEVC для кодирования видео в формате выборки цветности 4:4:4, выполняемый как для внутренне предсказанного остатка, так и для внешне предсказанного остатка. Способ ССР также был предложен, но еще не был принят для кодирования видео формата 4:2:0.

В дополнение к использованию яркости для предсказания компонент цветности можно использовать одну из компонент цветности для предсказания другой компоненты цветности. То есть, в некоторых примерных подходах, восстановленный ранее кодированный остаток  $C_b$  может быть использован для предсказания остатка  $C_r$ . (См. А. Khairat, Т. Nguyen, М. Siekmann, D. Marpe, "Non-RCE1: расширенное адаптивное меж-компонентное предсказание", Объединенная команда по совместной деятельности в области кодирования видео (JCT-VC) ITU-T SG16 WP3 и ISO/IEC JTC1/SC29/WG11, JCTVC-O0150, Женева, ноябрь 2013 г.).

В кодировании видео цветности 4:2:0 способ, названный режимом предсказания линейной модели (LM), был хорошо изучен во время разработки стандарта HEVC. (См. J. Chen, V. Seregi, W.-J. Han, J.-S. Kim, В.-М. Joen. "СЕ6.а.4: Внутреннее предсказание цветности с помощью восстановленных выборок яркости", Объединенная команда по совместной деятельности в области кодирования видео (JCT-VC) ITU-T SG16 WP3 и ISO/IEC JTC1/SC29/WG11, JCTVC-E266, Женева, 16-23 марта 2011 г.). С режимом предсказания LM выборки цветности предсказываются на основе восстановленных выборок яркости того же блока с

использованием линейной модели следующим образом:

$$pred_c(i, j) = \alpha \cdot rec_L(i, j) + \beta \quad (3)$$

где  $pred_c(i, j)$  представляет предсказание выборок цветности в блоке и  $rec_L(i, j)$  представляет подвергнутые субдискретизации восстановленные выборки яркости того же блока. Параметры  $\alpha$  и  $\beta$  получаются посредством минимизации ошибки регрессии между соседними восстановленными выборками яркости и цветности вокруг текущего блока.

$$E(\alpha, \beta) = \sum_i (y_i - (\alpha \cdot x_i + \beta))^2 \quad (4)$$

Параметры  $\alpha$  и  $\beta$  решаются следующим образом:

$$\alpha = \frac{N \sum x_i \cdot y_i - \sum x_i \cdot \sum y_i}{N \sum x_i \cdot x_i - \sum x_i \cdot \sum x_i} \quad (5)$$

$$\beta = (\sum y_i - \alpha \cdot \sum x_i) / N \quad (6)$$

где  $x_i$  является подвергнутой субдискретизации восстановленной опорной выборкой яркости,  $y_i$  является восстановленной опорной выборкой цветности и  $N$  является количеством опорных выборок.

Проблема с режимом предсказания LM заключается в том, что в режиме предсказания LM восстановленные выборки яркости используются для предсказания компонент цветности. Это вводит дополнительные задержки кодирования/декодирования в кодирование цветности. Такая задержка может значительно увеличить стоимость реализации, особенно для аппаратной реализации. Другим недостатком режима LM является то, что его нелегко использовать в режиме внешнего предсказания.

Проблема со способом кросс-компонентного предсказания (ССР) заключается в том, что улучшение кодирования ограничено вследствие стоимости сигнализации блочного уровня весовых коэффициентов (или любых других параметров для линейной модели).

В кросс-компонентном предсказании остатка весовые коэффициенты получаются как в кодере 20, так и в декодере 30 на основе пространственных или временных соседних восстановленных выборок. Остаток может быть, в то же время не ограничен внутренне и внешне предсказанным остатком, и может быть остатком любого типа способа предсказания. Нет необходимости передавать параметры линейного блока, как в ССР.

Далее рассматриваются способы определения весовых коэффициентов. Они могут быть применены индивидуально или в любой комбинации.

В общем, линейная модель может быть разработана для минимизации ошибки предсказания между двумя наборами сигналов  $X$  и  $Y$ , при этом  $X$  обозначается как сигнал предсказания, а  $Y$  обозначается как целевой сигнал во время оптимизации:

$$Min(E(\alpha, \beta)), \quad \text{где} \quad E(\alpha, \beta) = \sum_i (y_i - (\alpha \cdot x_i + \beta))^2 \quad (7)$$

один или несколько весовых коэффициентов  $\alpha$  и/или  $\beta$  применяются к внешне или внутренне предсказанным остаткам одной цветовой компоненты для предсказания другой цветовой компоненты. В одном примерном подходе один или несколько параметров линейной модели (например, весовой коэффициент  $\alpha$  или смещение  $\beta$ ), полученных из линейной модели выше, применяются к внешне или внутренне

предсказанным остаткам одной цветовой компоненты (например, компоненты яркости) для предсказания другой цветовой компоненты (например, компоненты Cb и/или Cr). Такая модель также может применяться к другим компонентам в порядке декодирования (например, от Cb до Cr). В дополнение порядок декодирования компонент может быть косвенно изменен, и технология кросс-компонентного предсказания все еще применяется.

Фиг. 4 представляет собой блок-схему последовательности операций, иллюстрирующую пример кодирования видеоданных в соответствии с одним аспектом этого раскрытия. В примерном подходе с фиг. 4 видеокодер 20 определяет предсказанный остаточный сигнал целевой цветовой компоненты (100). В одном таком примерном подходе кодер 20 определяет предсказанный остаточный сигнал целевой цветовой компоненты текущего блока в качестве функции одного или более параметров линейной модели и остаточного сигнала цветовой компоненты источника текущего блока. Видеокодер 20 определяет оставшийся остаточный сигнал целевой цветовой компоненты (102). В одном таком подходе кодер 20 определяет оставшийся остаточный сигнал целевой цветовой компоненты посредством вычитания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала целевой цветовой компоненты. Кодер 20 кодирует оставшийся остаточный сигнал целевой цветовой компоненты и остаточный сигнал цветовой компоненты источника (104) до вывода кодированных сигналов в битовом видеопотоке (106).

В одном примерном подходе линейная модель предсказывает предсказанный остаточный сигнал целевой цветовой компоненты из остаточного сигнала цветовой компоненты источника на основе параметров линейной модели. И кодер 20, и декодер 30 поддерживают параметры линейной модели упомянутой линейной модели в памяти и применяют их, как рассматривается ниже по тексту. В одном примерном подходе кодер 20 и декодер 30 вычисляют параметры линейной модели в качестве функции опорных блоков. В одном таком примерном подходе значения параметров по умолчанию поддерживаются в памяти для использования, когда вычисление значений параметров не является надлежащим.

Фиг. 5 представляет собой блок-схему последовательности операций, иллюстрирующую более подробный пример кодирования видеоданных в соответствии с одним аспектом этого раскрытия. В примерном подходе с фиг. 5 видеокодер 20 определяет предсказанный остаточный сигнал цветовой компоненты цветности (120). В одном таком примерном подходе кодер 20 определяет предсказанный остаточный сигнал целевой цветовой компоненты текущего блока в качестве функции одного или более параметров линейной модели и остаточного сигнала цветовой компоненты источника текущего блока. В одном примерном подходе параметры линейной модели включают в себя весовой коэффициент  $\alpha$  и смещение  $\beta$ . Кодер 20 использует линейную модель для предсказания предсказанного остаточного сигнала цветовой компоненты Cr цветности в качестве функции остаточного сигнала цветовой компоненты яркости. Кодер 20 определяет оставшийся остаточный сигнал для цветовой компоненты Cr цветности (122), кодирует оставшийся остаточный сигнал с остаточным сигналом цветовой компоненты (124) яркости и выводит кодированный оставшийся остаточный сигнал цветовой компоненты Cr цветности с кодированным остаточным сигналом яркости в битовом видеопотоке.

Далее будет рассмотрено декодирование кодированных остатков сигнала предсказания цветовой компоненты. Фиг. 6 представляет собой блок-схему последовательности операций, иллюстрирующую предсказание остаточного сигнала первой цветовой компоненты из остаточного сигнала второй цветовой компоненты в соответствии с одним аспектом этого раскрытия. В одном примерном подходе видеодекодер 30 принимает кодированный остаточный сигнал цветовой компоненты источника и кодированный оставшийся остаточный сигнал целевой цветовой компоненты, сохраняет кодированные остаточные сигналы в памяти видеодекодера 30 и декодирует кодированные остаточные значения. Видеодекодер 30 предсказывает предсказанный остаточный сигнал целевой цветовой компоненты текущего блока на основе декодированного остаточного сигнала цветовой составляющей источника текущего блока (140). Затем декодер 30 определяет остаточный сигнал для целевой цветовой компоненты в качестве функции предсказанного остаточного сигнала для целевой цветовой компоненты и декодированного оставшегося остаточного сигнала для целевой цветовой компоненты (142).

Фиг. 7 представляет собой блок-схему последовательности операций, иллюстрирующую предсказание остаточного сигнала цветовой компоненты Cb из остаточного сигнала цветовой компоненты Cr в соответствии с одним аспектом этого раскрытия. В одном примерном подходе видеодекодер 30 принимает кодированный остаточный сигнал цветовой компоненты Cr и оставшийся остаточный сигнал цветовой компоненты Cb, сохраняет кодированные остаточные сигналы в памяти видеодекодера 30 и декодирует кодированные остаточные сигналы. Видеодекодер 30 определяет предсказанный остаточный сигнал цветовой компоненты Cb текущего блока на основе декодированного остаточного сигнала цветовой компоненты Cr и параметров линейной модели (например, весового коэффициента  $\alpha$  и смещения  $\beta$ ) (160). Затем декодер 30 определяет остаточный сигнал для цветовой компоненты Cb в качестве функции предсказанного остаточного сигнала для цветовой компоненты Cb и декодированного оставшегося остаточного сигнала для цветовой компоненты Cb (162).

В одном примерном подходе значения по умолчанию параметров линейной модели используются

при предсказании остаточного сигнала целевой цветовой компоненты. В другом примерном подходе значения одного или более параметров получаются из дискретизированных сигналов, как будет подробно описано ниже по тексту. В одном таком примерном подходе полученные значения параметров линейной модели сохраняются в памяти видеокодера 20 и видеодекодера 30 и используются как значения по умолчанию до изменения.

Способы получения параметров линейной модели для внешнего и внутреннего предсказания рассматриваются ниже по тексту.

Фиг. 8 представляет собой блок-схему последовательности операций, иллюстрирующую выведение параметров линейной модели на основе внешнего предсказания в соответствии с одним аспектом этого раскрытия. В примерном подходе с фиг. 8 видеокодер 20 выбирает опорную внешне предсказанную цветовую компоненту  $X$  источника (200) и соответствующую опорную внешне предсказанную целевую цветовую компоненту  $Y$  (202). Видеокодер 20 вычисляет один или более весовых коэффициентов в линейной модели в качестве функции цветовой компоненты  $X$  источника и целевой цветовой компоненты  $Y$  (204), при этом весовые коэффициенты вычисляются для минимизации ошибки предсказания, как показано в уравнении (7) выше по тексту. В одном таком примерном подходе видеокодер 20 предсказывает значения целевой цветовой компоненты  $Y$  в качестве функции вычисленных весовых коэффициентов и значений цветовой компоненты  $X$  источника.

В другом примере линейная модель упрощается посредством установки  $\beta$  равным 0 или какому-либо значению по умолчанию. В другом примере линейная модель может быть упрощена посредством установки  $\alpha$  равным 1 или какому-либо значению по умолчанию.

В одном примерном подходе временные соседние выборки (например, выборки предсказания с компенсацией движения) используются в качестве опорных выборок для получения параметров  $\alpha$  и/или  $\beta$  линейной модели. В одном таком примере сигнал  $X$  предсказания представляет собой сигнал с компенсацией движения, принадлежащий одной компоненте (например, компоненте яркости с компенсацией движения, также известной как компонента яркости опорного блока), а целевой сигнал  $Y$  представляет собой сигнал с компенсацией движения, принадлежащий другой компоненте (например, компоненте  $S_b$  с компенсацией движения, также известной как компонента  $S_b$  опорного блока).

В одном примерном подходе сигнал  $X$  предсказания является остаточным сигналом опорного блока для одной компоненты (например, компоненты яркости), а целевой сигнал  $Y$  является остаточным сигналом опорного блока для другой компоненты (например, компоненты  $S_b$  и/или  $S_r$ ). В одном таком случае, остаток опорного блока вычисляется на ходу на основе, например, движения опорного или масштабированного вектора движения, основанного на векторе движения текущего блока.

В одном примерном подходе, кодер 20 сохраняет остаточное изображение опорного изображения и получает остаток опорного блока из сохраненного остаточного изображения опорного изображения. В одном таком примере остаточный блок расположен на основе вектора движения текущего блока. В другом таком примере остаточный блок дополнительно устанавливается к остаточному блоку, который тесно (например, в сетках  $4 \times 4$ ) перекрывается с опорным блоком.

В одном примерном подходе кодер 20 применяет интерполяцию при генерировании остаточного блока на основе вектора движения текущего блока. В одном таком примере интерполяция осуществляется через билинейный фильтр.

В одном примерном подходе кодер 20 использует восстановленные пространственные (например, верхние и/или левые) соседние вы (как во внутренней LM) в качестве опорных выборок для получения параметров  $\alpha$  и/или  $\beta$  линейной модели. В одном таком примере сигнал  $X$  предсказания включает в себя восстановленные значения одной компоненты соседних выборок (например, компоненты яркости), а целевой сигнал  $Y$  включает в себя восстановленные значения другой компоненты соседних выборок (например,  $S_b$  и/или  $S_r$ , или  $S_o$  и/или  $S_g$ ).

В одном примерном подходе кодер 20 использует как пространственные, так и временные соседние выборки для получения  $\alpha$  и/или  $\beta$ . То есть, сигнал  $X$  предсказания представляет собой выборочную композицию из сигнала  $X$ , полученного из временного опорного блока (как описано выше по тексту), и сигнала  $X$ , из пространственных соседних выборок (как описано выше по тексту). В то же время, целевой сигнал  $Y$  представляет собой (соответствующую) выборочную композицию из сигнала  $Y$ , полученного из временного опорного блока (как описано выше по тексту), и сигнала  $Y$ , из пространственных соседних выборок (как описано выше по тексту).

В некоторых примерных подходах кодер 20 выбирает между пространственными соседними выборками и временными соседними выборками при выборе опорных выборок, используемых для получения параметров линейной модели. В одном таком примере кодер 20 выбирает между использованием либо пространственных соседних выборок, либо временных соседних выборок в качестве функции размера текущего блока, или на основе другой статистики, соответствующей текущему блоку или одному или более соседним блокам. Репрезентативная статистика включает в себя значения пикселей, автоматическую корреляцию выборок одной цветовой компоненты и взаимную корреляцию между выборками двух цветовых компонент.

В некоторых примерных подходах устройство-источник 12 сигнализирует дополнительную информацию по линии 16 связи или через устройство 31 хранения. Дополнительная информация (может быть флагом) может сигнализироваться, например, в наборе параметров последовательности (SPS), наборе параметров изображения (PPS), заголовке слайса, LCU, группе уровня LCU или каком-либо другом уровне блока или текстовой части синтаксиса высокого уровня, для указания выборок X и Y, которые должны быть использованы для получения весовых коэффициентов.

В некоторых примерных подходах параметры  $\alpha$  и/или  $\beta$  линейной модели получаются посредством минимизации ошибки регрессии между опорными выборками яркости и цветности, как это сделано в режиме предсказания LM.

В некотором кросс-компонентном предсказании внутренне предсказанных остаточных примеров пространственные (например, верхние и/или левые) соседние восстановленные выборки могут быть использованы в качестве опорных выборок для получения параметров линейной модели.

Фиг. 9 представляет собой блок-схему последовательности операций, иллюстрирующую примерное кросс-компонентное предсказание внутренне предсказанных остатков в соответствии с одним аспектом этого раскрытия. В примерном подходе с фиг. 9, один или более весовых коэффициентов  $\alpha$  и/или  $\beta$  применяются к внутренне предсказанному остаточному сигналу одной цветовой компоненты для предсказания другой цветовой компоненты. В одном примерном подходе кодер 20 применяет один или более весовых коэффициентов  $\alpha$  и  $\beta$ , полученных из линейной модели уравнения (7), к внутренне предсказанному остаточному сигналу одной цветовой компоненты (например, компоненты яркости) для определения предсказанного остаточного сигнала другой цветовой компоненты (например, компоненты Cb и/или Cr). Как отмечено выше по тексту в обсуждении режима внешнего предсказания, такая модель также может применяться к другим компонентам в порядке декодирования (например, от Cb до Cr). В дополнение, порядок декодирования компонент может быть косвенно изменен, и технология кросс-компонентного предсказания все еще применяется. Затем предсказанный остаточный сигнал цветовой компоненты может быть объединен с оставшимся остаточным сигналом той же цветовой компоненты, чтобы достичь остаточного сигнала цветовой компоненты.

В подходе с фиг. 9 видеокодер 20 выбирает внутренне предсказанную цветовую компоненту X (300). В одном примерном подходе видеокодер 20 использует восстановленные выборки из соседних слева в качестве цветовой компоненты X источника. В другом примерном подходе видеокодер 20 использует восстановленные выборки из соседних сверху в качестве цветовой компоненты X источника. В еще одном примерном подходе видеокодер 20 вычисляет цветовую компоненту X источника в качестве функции восстановленных выборок из каждых соседних сверху и слева.

Видеокодер 20 выбирает соответствующую внутренне предсказанную целевую цветовую компоненту Y (302). Видеокодер 20 вычисляет один или более весовых коэффициентов в качестве функции опорной цветовой компоненты X источника и опорной целевой цветовой компоненты Y (304), при этом весовые коэффициенты вычисляются для минимизации ошибки предсказания, как показано в уравнении (7) выше по тексту. Затем видеокодер 20 предсказывает значения целевой цветовой компоненты Y в качестве функции вычисленных параметров линейной модели и значений цветовой компоненты X источника.

В одном примере, линейная модель может быть упрощена посредством установки  $\beta$  равным 0 или какому-либо значению по умолчанию. В другом примере линейная модель упрощается посредством установки  $\beta$  равным 1 или какому-либо значению по умолчанию.

#### **Учет шума или квантованной ошибки**

В кросс-компонентном предсказании остатка, когда линейная модель решается на основе пространственных или временных соседних восстановленных выборок, значение параметров линейной модели может быть субоптимальным вследствие шума/квантованной ошибки опорных выборок или вследствие различия между выборками текущего блока и опорными выборками. Чтобы улучшить производительность кросс-компонентного предсказания остатка, в некоторых примерах включено ограничение, так чтобы проблема оптимизации зависела от ограничения того, что  $\alpha$  ожидается близким к значению по умолчанию. Аналогичным образом, может быть включено ограничение, так чтобы проблема оптимизации зависела от ограничения того, что  $\beta$  ожидается близким к некоторому другому значению по умолчанию. В некоторых примерных подходах, смещая вычисление  $\alpha$  к одному или нескольким ограничениям, видеокодер 20 улучшает производительность кросс-компонентного предсказания остатка.

В некоторых примерных подходах видеокодер 20 использует предварительно заданный весовой коэффициент  $\alpha$  в качестве ограничения при определении  $\alpha$ . В одном таком примерном подходе предварительно заданный весовой коэффициент  $\alpha$  является функцией цветовой компоненты, вовлеченных в предсказание остатка. В некоторых примерных подходах, например, видеокодер 20 использует предварительно заданный весовой коэффициент по умолчанию -0,5 при предсказании остатка Cr с использованием остатка Cr. В некоторых примерных подходах видеокодер 20 использует предварительно заданный весовой коэффициент по умолчанию 0 при предсказании остатка цветности с использованием остатка яркости. В некоторых таких примерных подходах, предварительно заданные весовые коэффициенты по умолчанию  $\alpha$  варьируются посредством цветовой компоненты и цветового пространства.

В одном примерном подходе видеокодер 20 адаптивно модифицирует значение по умолчанию  $\alpha$  по мере необходимости. Например, видеокодер 20 может сигнализировать значение по умолчанию  $\alpha$  как 0,5 при предсказании остатка  $C_r$  с использованием остатка  $C_r$  на основе значений  $\alpha$ , определенных экспериментально для пространства  $YCbCr$ , но затем модифицировать значение по умолчанию на основе содержимого битового видеопотока. В одном примерном подходе такие изменения могут возникать и сигнализироваться на высоком уровне, чтобы видеокодер 20 и видеодекoder 30 могли адаптироваться к изменениям в видеопотоке.

В некоторых примерных подходах устройство-источник 12 сигнализирует значение по умолчанию весового коэффициента  $\alpha$  и/или значение смещения  $\beta$  в битовом потоке. В различных примерных подходах значения по умолчанию для весовых коэффициентов сигнализируются в SPS, PPS, заголовке слайса или любых других местах в качестве синтаксиса высокого уровня. В некоторых примерах значения весовых коэффициентов по умолчанию сигнализируются в CU, LCU, группе LUC или другом уровне блока.

В некоторых примерных подходах, когда значение по умолчанию не сигнализируется, используются предварительно заданные значения. В некоторых таких примерных подходах для разных цветовых пространств может быть использовано различное значение по умолчанию. В некоторых примерных подходах значение параметра по умолчанию используется только для определенных цветовых пространств.

В некоторых примерных подходах значение по умолчанию весового коэффициента  $\alpha$  и/или значения смещения  $\beta$  обновляются с использованием ранее полученного на ходу значения  $\alpha$  и/или значения смещения  $\beta$  для размеров блоков, превышающих предварительно заданный порог, которые, как правило, более терпимы к ошибке шума/квантования. В некоторых таких примерах используется порог размера блока  $16 \times 16$ ,  $32 \times 32$  или более.

В некоторых примерных подходах включено ограничение на вариацию значения  $\alpha$  взвешивания и/или значения смещения  $\beta$ , так что разница между полученным в настоящее время значением  $\alpha$  и/или значением смещения  $\beta$  и соседним полученным значением  $\alpha$  и/или значениями смещения  $\beta$  меньше предварительно заданного порогового значения или отношения.

В некоторых примерах кодер 20 включает в себя механизм выбора, который определяет, учитывается ли выделенное значение по умолчанию для выведения весового коэффициента. В некоторых таких примерах выбор зависит от контекста текущего блока. Например, значение по умолчанию весового коэффициента  $\alpha$  может быть учтено при получении весового коэффициента, когда кросс-компонентное предсказание применяется между двумя компонентами цветности, в то время как это может не учитываться, когда компонента цветности предсказывается из компоненты яркости. В некоторых примерах значение по умолчанию весового коэффициента  $\alpha$  может быть учтено при получении выведения весового коэффициента, когда размер блока выше или ниже порогового размера или на основе других признаков текущего блока.

В одном примерном подходе выделенное значение по умолчанию может быть учтено при получении весового коэффициента и может зависеть от контекста текущего блока. В одном примерном подходе стоимость регрессии относительно значения  $\alpha$  по умолчанию добавляется к функции ошибки следующим образом:

$$E(\alpha, \beta) = \sum_i (y_i - (\alpha \cdot x_i + \beta))^2 + \lambda \cdot (\alpha - \alpha_{\text{default}})^2 \quad (8)$$

И оптимальное  $\alpha$  решается как:

$$\alpha = \frac{N \sum x_i \cdot y_i - \sum x_i \cdot \sum y_i + \alpha_{\text{default}} \cdot \lambda}{N \sum x_i \cdot x_i - \sum x_i \cdot \sum x_i + \lambda} \quad (9)$$

Основываясь на уравнении (9), полученное значение  $\alpha$  должно оставаться близким к значению по умолчанию.

Значение  $\lambda$  определяет, насколько велико влияние значения  $\alpha_{\text{default}}$  по умолчанию на окончательное полученное значение  $\alpha$ .

Значение  $\alpha_{\text{default}}$  по умолчанию имеет большее влияние, когда используется большее значение  $\lambda$ . Значение  $\lambda$  может быть предварительно заданным как для кодера, так и для декодера, в противном случае значение  $\lambda$  может сигнализироваться в битовом потоке. В примерном подходе значение  $\lambda$  может быть установлено равным  $(\sum x_i \cdot x_i) \gg k$ , где  $k$  представляет собой предварительно заданное фиксированное значение (например, 7, 8 или 9). В другом примерном подходе значение  $\lambda$  может изменяться на основе контекста. Например, значение  $\lambda$  может варьироваться на основе признака текущего блока (например, размера блока, используются ли пространственные или временные опорные выборки и так далее).

Этот процесс также может быть использован при получении значения  $\alpha$  и/или значения смещения  $\beta$  в режиме предсказания LM, описанном выше по тексту, где восстановленные выборки одной компоненты (например, яркости или  $C_b$ ) используются для предсказания выборки другой компоненты (например,  $C_b$  или  $C_r$ ) с линейной моделью.

В некоторых примерных подходах кодер 20 уменьшает сложность процесса выведения параметров в кросс-компонентном предсказании остатка посредством субдискретизации или подвыборки опорных выборок (например, пространственных или временных соседних восстановленных выборок) перед использованием подвергнутых субдискретизации или подвыборке опорных выборок для получения параметров  $\alpha$  и  $\beta$ . В некоторых примерах кодер 20 определяет, когда и применяется ли субдискретизация или подвыборка к опорным выборкам. В некоторых таких примерах решение о субдискретизации или подвыборке зависит от контекста текущего блока и/или опорных выборок, например, от размера текущего блока, относится ли опорная выборка к временной соседней или пространственной соседней, или от других признаков текущего блока или опорных выборок.

В некоторых таких примерных подходах кодер 20 варьирует отношение субдискретизации или подвыборки в качестве функции размера текущего блока. В некоторых примерных подходах более высокое отношение субдискретизации/подвыборки применяется, когда обрабатывается более крупный блок.

В некоторых примерных подходах отношение субдискретизации/подвыборки отличается в зависимости от того, являются ли опорные выборки пространственными соседними выборками или временными соседними выборками. В одном таком примерном подходе к временным опорным выборкам может быть применено относительно высокое отношение субдискретизации/подвыборки, а к пространственным опорным выборкам может быть применено относительно меньшее отношение субдискретизации/подвыборки (или без отношения субдискретизации/подвыборки).

Этот способ субдискретизации/подвыборки также может быть применен к режиму предсказания LM, где восстановленные выборки одной компоненты (например, яркости или  $S_b$ ) используются для предсказания с использованием линейной модели выборки другой компоненты (например,  $S_b$  или  $S_r$ ).

Необходимо признать, что, в зависимости от примера, конкретные действия или события любой из технологий, описанных в этом документе, могут быть выполнены в другой последовательности, могут быть добавлены, соединены или пропущены (например, не все описанные действия или события необходимы для осуществления технологий). Более того, в некоторых примерах, действия или события могут выполняться одновременно, например, благодаря использованию многопоточной обработки, обработки прерываний, или нескольких процессоров, а не последовательно.

В одном или более примерах, описанные функции могут быть реализованы в аппаратном обеспечении, программном обеспечении, программно-аппаратном обеспечении, или любой их комбинации. При реализации в программном обеспечении, функции могут быть сохранены в или переданы через одну или более инструкции или код на компьютерно-читаемом носителе информации и исполнены основанном на аппаратном обеспечении блоком обработки. Компьютерно-читаемые носители могут включать в себя компьютерно-читаемый запоминающий носитель, который соответствует материальному носителю, например, носители хранения данных, или носитель связи, включая какой-либо носитель, который содействует переносу компьютерной программы из одного места в другое, например, согласно протоколу связи. Таким образом, компьютерно-читаемые носители, в общем, могут соответствовать (1) материальному компьютерно-читаемому запоминающему носителю, который является энергонезависимым, или (2) среде связи, такой как сигнал или несущая. Носители хранения данных могут быть какими-либо доступными носителями, которые могут быть доступны посредством одного или более компьютеров или одного или более процессоров для извлечения инструкций, кода и/или структур данных для реализации технологий, описанных в этом раскрытии. Компьютерный программный продукт может включать в себя компьютерно-читаемый носитель. Используемый в этом документе термин "сигнализация" может включать в себя хранение или иное включение данных с кодированным битовым потоком. Другими словами, в различных примерах в соответствии с этим раскрытием термин "сигнализация" может быть ассоциирован с передачей данных в режиме реального времени или, в качестве альтернативы, с передачей данных, которая не выполняется в режиме реального времени.

Посредством примера, а не ограничения, такие компьютерно-читаемые запоминающие носители могут содержать RAM, ROM, EEPROM, CD-ROM или другое оптическое дисковое средство хранения, магнитное дисковое средство хранения или другие магнитные устройства хранения, флэш-память, или любой другой носитель, который может быть использован для хранения желаемого программного кода в форме инструкций или структур данных и который может быть доступным посредством компьютера. Также, любое соединение надлежащим образом именуется компьютерно-читаемым носителем. Например, если инструкции передаются с вебсайта, сервера, или другого удаленного источника с использованием коаксиального кабеля, оптоволоконного кабеля, витой пары, цифровой абонентской линии связи (DSL), или беспроводных технологий, таких как инфракрасные, радио и микроволновые, тогда коаксиальный кабель, оптоволоконный кабель, витая пара, DSL или беспроводные технологии, такие как инфракрасные, радио и микроволновые включаются в определение носителя. Тем не менее, следует понимать, что компьютерно-читаемые запоминающие носители и носители информации для хранения данных не включают в себя соединения, несущие волны, сигналы, или другие временные носители, а вместо этого относятся к не временным, вещественным запоминающим носителям. Диск (disk) и диск (disc), как используются в этом документе, включает в себя компакт-диск (CD), лазерный диск, оптический диск, цифровой универсальный диск (DVD), флоппи диск и Blu-ray диск, где диски (disks) обычно воспроизво-

дт данные магнитно, тогда как диски (discs) воспроизводят данные оптически с помощью лазеров. Комбинации вышеописанного также должны быть включены в объем компьютерно-читаемых носителей.

Инструкции могут быть исполнены одним или более процессорами, например, одним или более цифровыми сигнальными процессорами (DSP), микропроцессорами общего назначения, специализированными интегральными схемами (ASIC), программируемыми пользователем вентильными матрицами (FPGA), или другими эквивалентными интегрированными или дискретными схемами логики. Соответственно, термин "процессор", как используется в данном документе, может относиться к любой из вышеуказанных структур или любой другой структуре, подходящей для реализации технологий, описанных в этом документе. В дополнение, в некоторых аспектах, функциональность описанная в этом документе, может быть обеспечена в специализированных модулях аппаратного обеспечения и/или программного обеспечения, сконфигурированных с возможностью кодирования и декодирования, или включена в объединенный кодек. Также, технологии могут быть полностью реализованы в одной или более схемах или логических элементах.

Технологии этого раскрытия могут быть реализованы в широком ряде устройств или приборов, в том числе беспроводные микротелефонные трубки, интегральная схема (IC) или набор IC (например, набор микросхем). Различные компоненты, модули, или блоки описаны в этом раскрытии, чтобы подчеркнуть функциональные аспекты устройств, сконфигурированных с возможностью выполнения раскрытых технологий, но необязательно требуют реализации посредством разных блоков аппаратного обеспечения. Скорее, как описано выше тексту, различные модули могут быть объединены в аппаратном модуле кодека или обеспечены комплектом способных взаимодействовать друг с другом аппаратных модулей, включающих в себя один или более процессоров, как описано выше, в сочетании с подходящим программным обеспечением и/или программно-аппаратным обеспечением.

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

#### ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ кодирования видеоданных для формата кодирования, использующего режим предсказания на основе линейной модели (LM), причем способ содержит

определение (100), в кодере, предсказанного остаточного сигнала целевой цветовой компоненты текущего блока в качестве функции от одного или более параметров линейной модели предсказания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала цветовой компоненты источника и от остаточного сигнала цветовой компоненты источника текущего блока;

определение (102), в кодере, оставшегося остаточного сигнала для целевой цветовой компоненты посредством вычитания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала целевой цветовой компоненты;

кодирование (104) значений упомянутого оставшегося остаточного сигнала целевой цветовой компоненты; и

вывод (106) кодированных значений упомянутого оставшегося остаточного сигнала в битовом видеопотоке,

при этом определение предсказанного остаточного сигнала включает в себя определение упомянутого одного или более параметров линейной модели,

при этом определение одного или более параметров линейной модели включает в себя:

(i) определение ошибки предсказания в качестве функции от значений опорных выборок опорной целевой цветовой компоненты и от значений опорных выборок опорной цветовой компоненты источника; и

(ii) выбор значений одного или более параметров линейной модели, которые минимизируют ошибку предсказания, причем выбор включает в себя получение одного из параметров линейной модели посредством наложения ограничения на один из параметров линейной модели, заключающегося в том, чтобы сделать его близким к предварительно заданному значению параметра,

при этом значения опорных выборок опорной целевой цветовой компоненты и значения опорных выборок опорной цветовой компоненты источника являются первой и второй цветовыми компонентами, соответственно, восстановленных пространственных соседних выборок.

2. Способ по п.1, при этом получение одного из параметров линейной модели дополнительно включает в себя одно из

ограничения вариации между полученным параметром линейной модели и параметрами линейной модели одного или более временных или пространственных соседних блоков,

замены предварительно заданного значения параметра на полученный параметр линейной модели и замены предварительно заданного значения параметра полученным параметром линейной модели, если размер блока больше, чем предварительно заданный пороговый размер блока.

3. Способ по п.1, в котором

выбор значений одного или более параметров линейной модели включает в себя определение, явля-

ется ли цветовая компонента источника первым типом цветовой компоненты, и

упомянутый один из параметров линейной модели получается в ответ на то, что цветовая компонента источника определяется, как упомянутый первый тип цветовой компоненты.

4. Способ по п.1, в котором линейная модель включает в себя набор параметров, включающих в себя весовой коэффициент  $\alpha$  и смещение  $\beta$ , при этом выбор значений одного или более параметров линейной модели включает в себя получение весового коэффициента  $\alpha$  и смещения  $\beta$ , которые минимизируют  $E(\alpha, \beta)$ , где

$$E(\alpha, \beta) = \sum_1 (y_i - (\alpha \cdot x_i + \beta))^2 + \lambda \cdot (\alpha - \alpha_{\text{default}})^2$$

где  $E(\alpha, \beta)$  - ошибка предсказания между двумя наборами значений  $x$  и  $y$ ,  $\alpha_{\text{default}}$  является значением по умолчанию  $\alpha$ ,  $y_i$  являются значениями опорных выборок опорной целевой цветовой компоненты,  $x_i$  являются значениями опорных выборок опорной цветовой компоненты источника, значение  $\lambda$  определяет, насколько велико влияние значения  $\alpha_{\text{default}}$  на значение  $\alpha$ .

5. Устройство для кодирования видеоданных для формата кодирования, использующего режим предсказания на основе линейной модели (LM), причем устройство содержит память, сконфигурированную с возможностью хранения видеоданных; и один или более процессоров, сконфигурированных с возможностью определения (100) предсказанного остаточного сигнала целевой цветовой компоненты текущего блока в качестве функции от одного или более параметров линейной модели предсказания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала цветовой компоненты источника и от остаточного сигнала цветовой компоненты источника текущего блока;

определения (102) оставшегося остаточного сигнала для целевой цветовой компоненты посредством вычитания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала целевой цветовой компоненты;

кодирования (104) значений упомянутого оставшегося остаточного сигнала целевой цветовой компоненты; и

вывода (106) кодированных значений упомянутого оставшегося остаточного сигнала в битовом видеопотоке,

при этом определение предсказанного остаточного сигнала включает в себя определение упомянутого одного или более параметров линейной модели,

при этом определение одного или более параметров линейной модели включает в себя:

(i) определение ошибки предсказания в качестве функции значений опорных выборок опорной целевой цветовой компоненты и значений опорных выборок опорной цветовой компоненты источника; и

(ii) выбор значений одного или более параметров линейной модели, которые минимизируют ошибку предсказания, причем выбор включает в себя получение одного из параметров линейной модели посредством наложения ограничения на один из параметров линейной модели, заключающегося в том, чтобы сделать его близким к предварительно заданному значению параметра,

при этом значения опорных выборок опорной целевой цветовой компоненты и значения опорных выборок опорной цветовой компоненты источника являются первой и второй цветовыми компонентами, соответственно, восстановленных пространственных соседних выборок.

6. Устройство по п.5, при этом выбор значений одного или более параметров линейной модели включает в себя выбор предварительно заданного значения параметра на основе типа цветовой компоненты целевой цветовой компоненты и на основе типа цветовой компоненты цветовой компоненты источника, и получение одного из параметров линейной модели посредством наложения ограничения на параметр линейной модели, заключающегося в том, чтобы сделать его близким к выбранному предварительно заданному значению параметра.

7. Устройство по п.5, при этом получение одного из параметров линейной модели дополнительно включает в себя одно из

ограничения вариации между полученным параметром линейной модели и параметрами линейной модели одного или более временных или пространственных соседних блоков,

замены предварительно заданного значения параметра на полученный параметр линейной модели, и

замены предварительно заданного значения параметра полученным параметром линейной модели, если размер блока больше, чем предварительно заданный пороговый размер блока.

8. Устройство по п.5, при этом

выбор значений одного или более параметров линейной модели включает в себя определение, относится ли цветовая компонента источника к первому типу цветовой компоненты, и

упомянутый один из параметров линейной модели получается в ответ на определение, что цветовая компонента источника относится к упомянутому первому типу цветовой компоненты.

9. Устройство по п.5, при этом линейная модель включает в себя набор параметров, включающих в себя весовой коэффициент  $\alpha$  и смещение  $\beta$ , при этом выбор значений одного или более параметров ли-

нейной модели включает в себя получение весового коэффициента  $\alpha$  и смещения  $\beta$ , которые минимизируют  $E(\alpha, \beta)$ , где

$$E(\alpha, \beta) = \sum_1 (y_i - (\alpha \cdot x_i + \beta))^2 + \lambda \cdot (\alpha - \alpha_{\text{default}})^2$$

где  $E(\alpha, \beta)$  - ошибка предсказания между двумя наборами значений  $x$  и  $y$ ,  $\alpha_{\text{default}}$  является значением по умолчанию  $\alpha$ ,  $y_i$  являются значениями опорных выборок опорной целевой цветовой компоненты,  $x_i$  являются значениями опорных выборок опорной цветовой компоненты источника, значение  $\lambda$  определяет, насколько велико влияние значения  $\alpha_{\text{default}}$  на значение  $\alpha$ .

10. Способ декодирования видеоданных для формата кодирования, использующего режим предсказания на основе линейной модели (LM), содержащий

определение (140) предсказанного остаточного сигнала целевой цветовой компоненты в качестве функции от одного или более параметров линейной модели предсказания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала цветовой компоненты источника и от остаточного сигнала цветовой компоненты источника;

декодирование оставшегося остаточного сигнала целевой цветовой компоненты, полученного в кодере посредством вычитания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала целевой цветовой компоненты; и

определение (142) остаточного сигнала целевой цветовой компоненты посредством сложения декодированного оставшегося остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты,

при этом определение предсказанного остаточного сигнала включает в себя определение упомянутого одного или более параметров линейной модели,

при этом определение одного или более параметров линейной модели включает в себя:

(i) определение ошибки предсказания в качестве функции от значений опорных выборок опорной целевой цветовой компоненты и от значений опорных выборок опорной цветовой компоненты источника; и

(ii) выбор значений одного или более параметров линейной модели, которые минимизируют ошибку предсказания, причем выбор включает в себя получение одного из параметров линейной модели посредством наложения ограничения на один из параметров линейной модели, заключающегося в том, чтобы сделать его близким к предварительно заданному значению параметра,

при этом значения опорных выборок опорной целевой цветовой компоненты и значения опорных выборок опорной цветовой компоненты источника являются первой и второй цветовыми компонентами, соответственно, восстановленных пространственных соседних выборок.

11. Способ по п.10, при этом получение одного из параметров линейной модели включает в себя замену предварительно заданного значения параметра полученным параметром линейной модели.

12. Способ по п.10, при этом получение одного из параметров линейной модели включает в себя замену предварительно заданного значения параметра полученным параметром линейной модели, если размер блока больше, чем предварительно заданный пороговый размер блока.

13. Устройство для декодирования видеоданных для формата кодирования, использующего режим предсказания на основе линейной модели (LM), причем устройство содержит

память, сконфигурированную с возможностью хранения видеоданных и

один или более процессоров, сконфигурированных с возможностью

определения (140) предсказанного остаточного сигнала целевой цветовой компоненты в качестве функции от одного или более параметров линейной модели предсказания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала цветовой компоненты источника и от остаточного сигнала цветовой компоненты источника;

декодирования оставшегося остаточного сигнала целевой цветовой компоненты, полученного в кодере посредством вычитания предсказанного остаточного сигнала целевой цветовой компоненты из остаточного сигнала целевой цветовой компоненты и

определения (142) остаточного сигнала целевой цветовой компоненты посредством сложения декодированного оставшегося остаточного сигнала целевой цветовой компоненты и предсказанного остаточного сигнала целевой цветовой компоненты,

при этом определение предсказанного остаточного сигнала включает в себя определение упомянутого одного или более параметров линейной модели,

при этом определение одного или более параметров линейной модели включает в себя:

(i) определение ошибки предсказания в качестве функции значений опорных выборок опорной целевой цветовой компоненты и значений опорных выборок опорной цветовой компоненты источника и

(ii) выбор значений одного или более параметров линейной модели, которые минимизируют ошибку предсказания, причем выбор включает в себя получение одного из параметров линейной модели посредством наложения ограничения на один из параметров линейной модели, заключающегося в том, чтобы сделать его близким к предварительно заданному значению параметра,

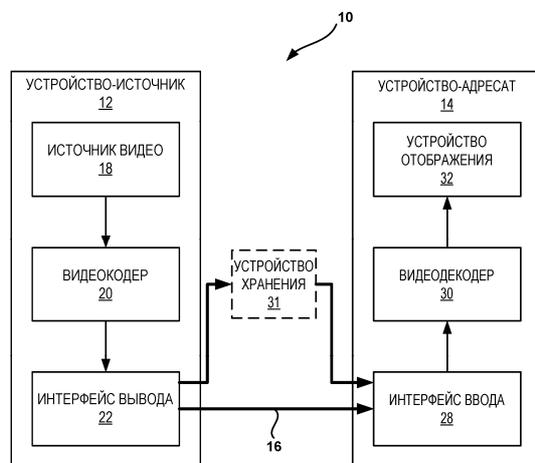
при этом значения опорных выборок опорной целевой цветовой компоненты и значения опорных выборок опорной цветовой компоненты источника являются первой и второй цветовыми компонентами, соответственно, восстановленных пространственных соседних выборок.

14. Устройство по п.13, при этом линейная модель включает в себя набор параметров, включающих в себя весовой коэффициент  $\alpha$  и смещение  $\beta$ , при этом выбор значений одного или более параметров линейной модели включает в себя получение весового коэффициента  $\alpha$  и смещения  $\beta$ , которые минимизируют  $E(\alpha, \beta)$ , где

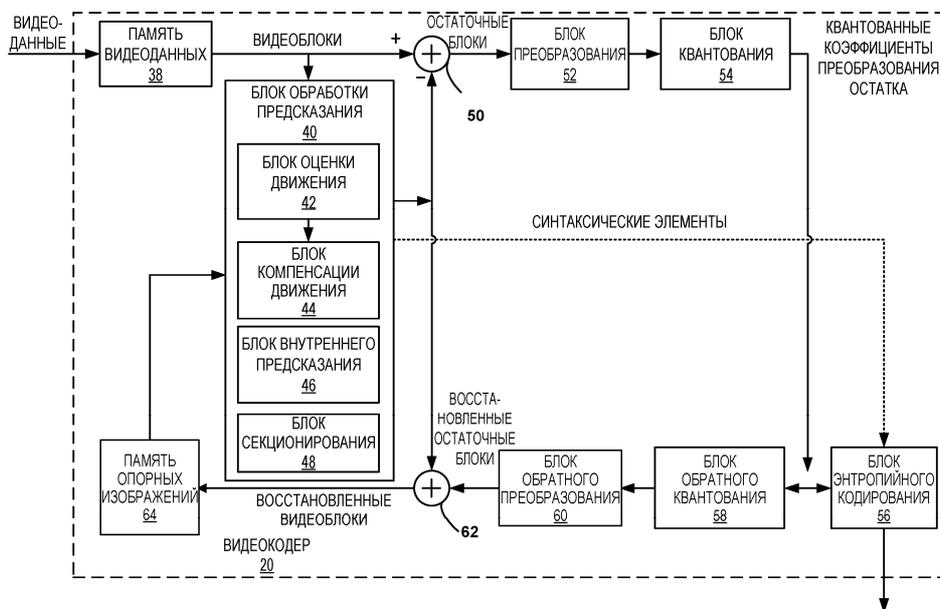
$$E(\alpha, \beta) = \sum_i (y_i - (\alpha \cdot x_i + \beta))^2 + \lambda \cdot (\alpha - \alpha_{\text{default}})^2$$

где  $E(\alpha, \beta)$  - ошибка предсказания между двумя наборами значений  $x$  и  $y$ ,  $\alpha_{\text{default}}$  является значением по умолчанию  $\alpha$ ,  $y_i$  являются значениями опорных выборок опорной целевой цветовой компоненты,  $x_i$  являются значениями опорных выборок опорной цветовой компоненты источника, значение  $\lambda$  определяет, насколько велико влияние значения  $\alpha_{\text{default}}$  на значение  $\alpha$ .

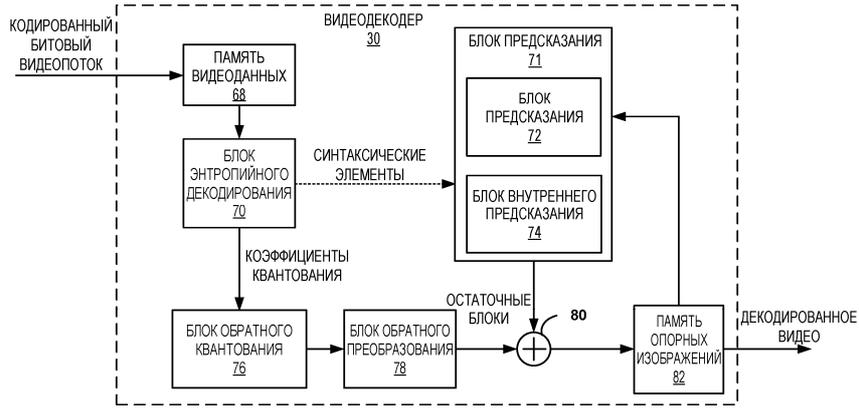
15. Компьютерно-читаемый запоминающий носитель, содержащий сохраненные на нем инструкции, которые при исполнении предписывают одному или более процессорам выполнять способ по любому из пп.1-4 и 10-12.



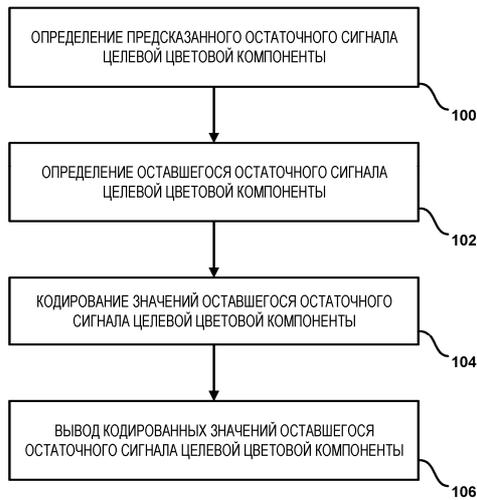
Фиг. 1



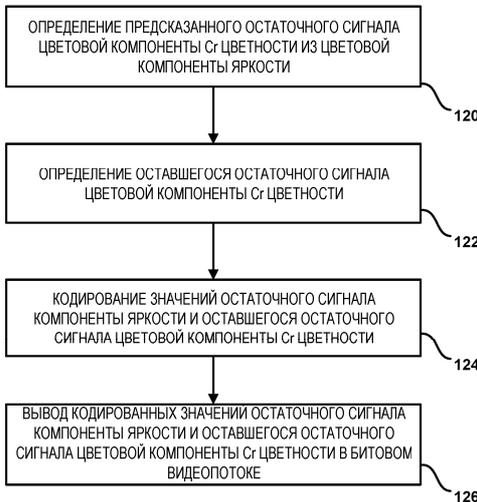
Фиг. 2



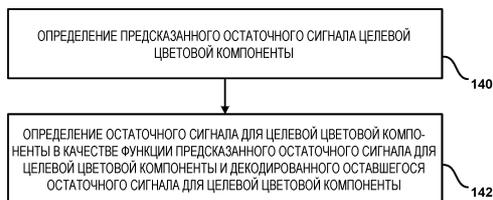
Фиг. 3



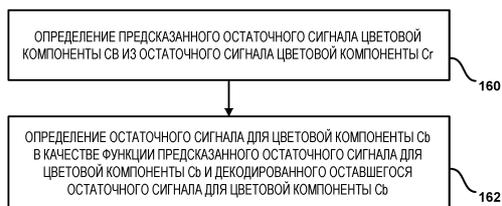
Фиг. 4



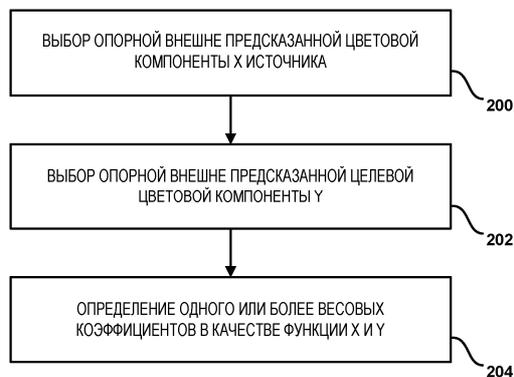
Фиг. 5



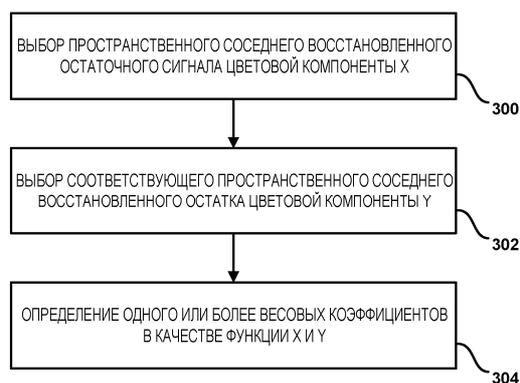
Фиг. 6



Фиг. 7



Фиг. 8



Фиг. 9

