



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

## (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

*H04N 19/117 (2023.08); H04N 19/70 (2023.08); H04N 19/82 (2023.08)*

(21)(22) Заявка: 2021127239, 24.03.2020

(24) Дата начала отсчета срока действия патента:  
24.03.2020Дата регистрации:  
30.01.2024

Приоритет(ы):

(30) Конвенционный приоритет:  
24.03.2019 CN PCT/CN2019/079395

(43) Дата публикации заявки: 16.03.2023 Бюл. № 8

(45) Опубликовано: 30.01.2024 Бюл. № 4

(85) Дата начала рассмотрения заявки PCT на  
национальной фазе: 16.09.2021(86) Заявка PCT:  
CN 2020/080827 (24.03.2020)(87) Публикация заявки PCT:  
WO 2020/192644 (01.10.2020)Адрес для переписки:  
101000, Москва, ул. Мясницкая, 13, стр. 5, ООО  
"Союзпатент", С.Б. Фелициной

(72) Автор(ы):

**ЧЖАН, Ли (US),  
ЧЖАН, Кай (US),  
ЛЮ, Хунбинь (CN),  
ВАН, Юэ (CN)**

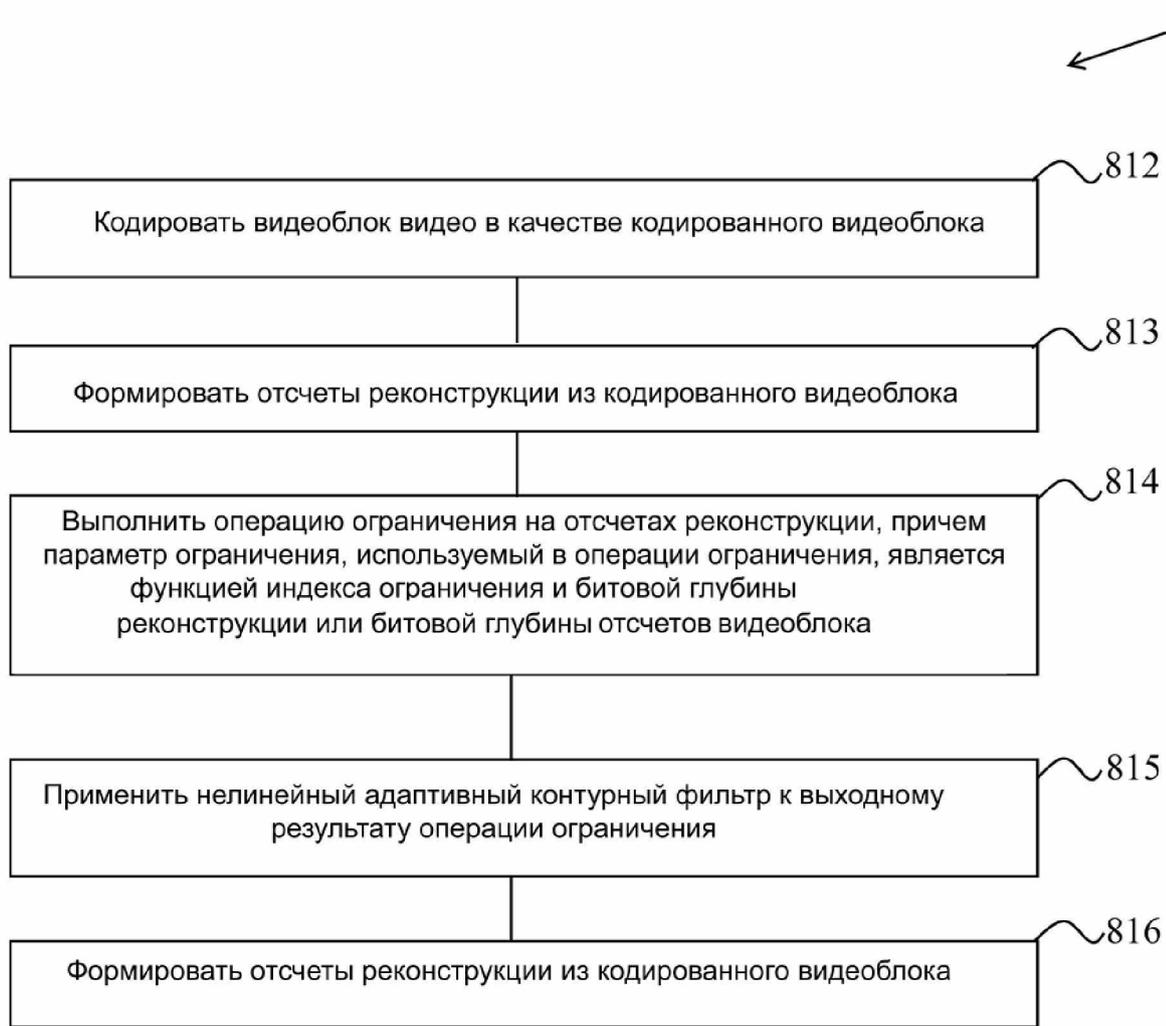
(73) Патентообладатель(и):

**БЕЙДЖИН БАЙТДЭНС НЕТВОРК  
ТЕКНОЛОДЖИ КО., ЛТД. (CN),  
БАЙТДЭНС ИНК. (US)**(56) Список документов, цитированных в отчете  
о поиске: EP 3297282 A1, 2018.03.21. US  
2014023136 A1, 2014.01.23. US 2018091812 A1,  
2018.03.29. US 9998737 B2, 2018.06.12. US  
2012020413 A1, 2012.01.26. US 2015092847 A1,  
2015.04.02. RU 2584501 C1, 2016.05.20. US  
2018115787 A1, 2018.04.26.(54) НЕЛИНЕЙНАЯ АДАПТИВНАЯ КОНТУРНАЯ ФИЛЬТРАЦИЯ ПРИ ОБРАБОТКЕ  
ВИДЕОДАНЫХ

(57) Реферат:

Изобретение относится к технологиям обработки видеоданных. Техническим результатом является повышение эффективности видеокодирования. Результат достигается тем, что кодируют видеоблок видео в качестве кодированного видеоблока; формируют отсчеты реконструкции из кодированного видеоблока; выполняют операцию ограничения на отсчетах реконструкции, причем параметр ограничения,

используемый в операции ограничения, является функцией индекса ограничения и битовой глубины отсчетов реконструкции или битовой глубины отсчетов видеоблока; применяют нелинейный адаптивный контурный фильтр к выходному результату операции ограничения и формируют кодированное представление видео с использованием кодированного видеоблока. 4 н. и 11 з.п. ф-лы, 19 ил., 2 табл.



Фиг. 8А



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC  
*H04N 19/117 (2023.08); H04N 19/70 (2023.08); H04N 19/82 (2023.08)*

(21)(22) Application: **2021127239, 24.03.2020**

(24) Effective date for property rights:  
**24.03.2020**

Registration date:  
**30.01.2024**

Priority:

(30) Convention priority:  
**24.03.2019 CN PCT/CN2019/079395**

(43) Application published: **16.03.2023 Bull. № 8**

(45) Date of publication: **30.01.2024 Bull. № 4**

(85) Commencement of national phase: **16.09.2021**

(86) PCT application:  
**CN 2020/080827 (24.03.2020)**

(87) PCT publication:  
**WO 2020/192644 (01.10.2020)**

Mail address:  
**101000, Moskva, ul. Myasnitskaya, 13, str. 5, OOO "Soyuzpatent", S.B. Felitsinoj**

(72) Inventor(s):

**ZHANG, Li (US),  
ZHANG, Kai (US),  
LIU, Hongbin (CN),  
WANG, Yue (CN)**

(73) Proprietor(s):

**BEIJING BYTEDANCE NETWORK  
TECHNOLOGY CO., LTD. (CN),  
BYTEDANCE INC. (US)**

(54) **NONLINEAR ADAPTIVE CONTOUR FILTERING IN VIDEO DATA PROCESSING**

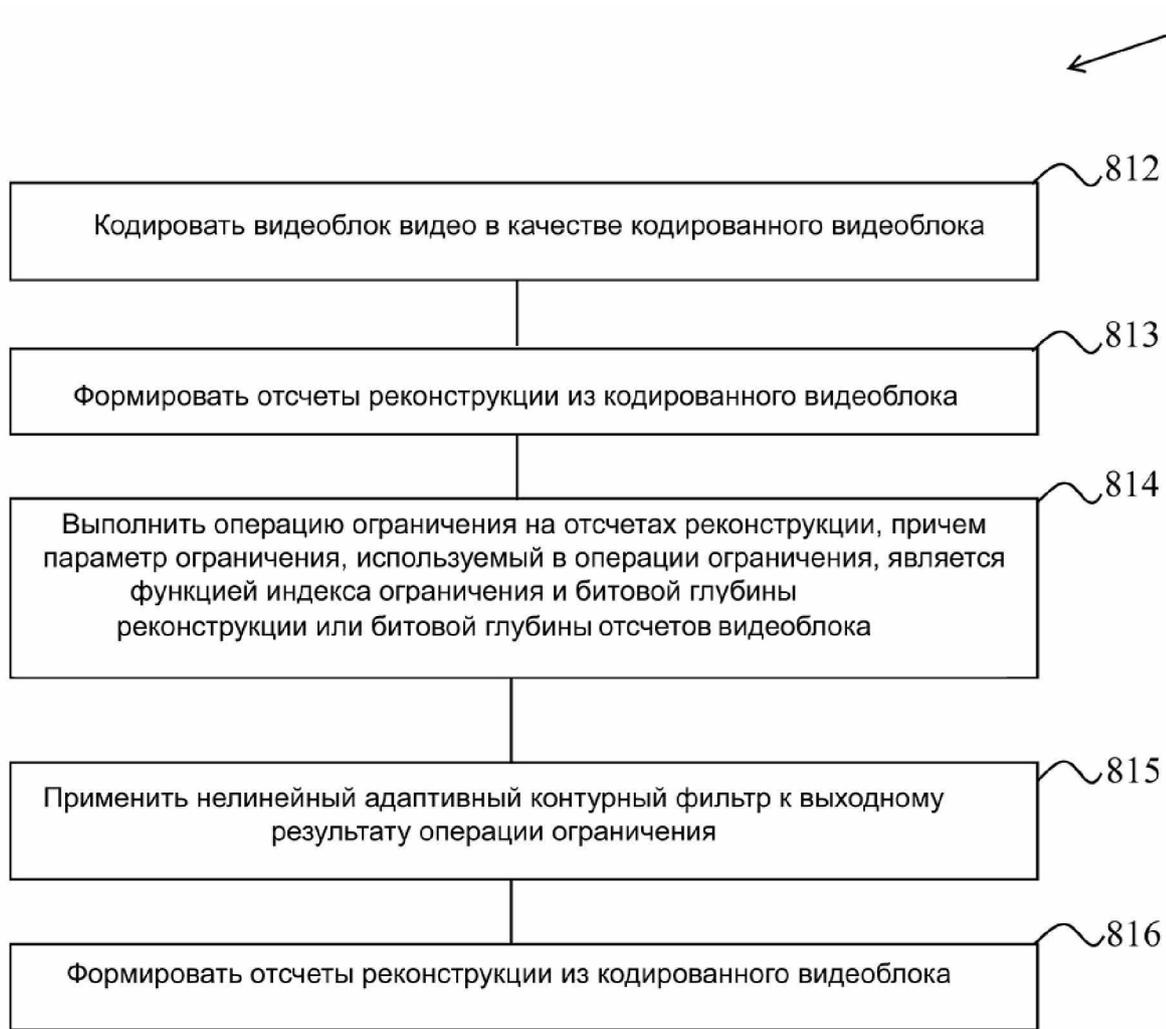
(57) Abstract:

FIELD: video data processing technologies.

SUBSTANCE: encoding the video block of video as an encoded video block; generating reconstruction samples from the encoded video block; performing a constraint operation on the reconstruction samples, wherein the constraint parameter used in the constraint operation is a function of the constraint index and the

bit depth of the reconstruction samples or the bit depth of the video block samples; applying a nonlinear adaptive contour filter to the output result of the constraint operation and generating an encoded video representation using the encoded video block.

EFFECT: increase of efficiency of video encoding.  
15 cl, 19 dwg, 2 tbl



Фиг. 8А

Область техники, к которой относится изобретение

Настоящий патентный документ относится к технологиям обработки видеоданных, устройствам и системам.

Уровень техники

5 Несмотря на прогресс в области сжатия видеосигналов, цифровое видео все еще рассчитывает на использование самой большой полосы пропускания в Интернете и других сетях цифровой связи. Поскольку количество присоединенных пользовательских устройств, способных принимать и отображать видео, увеличивается, ожидается, что потребность в ширине полосы пропускания для использования цифрового видео будет  
10 продолжать расти.

Раскрытие сущности изобретения

Описываются устройства, системы и способы, связанные с обработкой цифрового видео и, например, с нелинейной адаптивной контурной фильтрацией для обработки видеоданных. Описанные способы могут применяться как к существующим стандартам  
15 видеокодирования (например, к стандартам высокоэффективного видеокодирования (High Efficiency Video Coding, HEVC)), так и к будущим стандартам (например, к универсальному видеокодированию (Versatile Video Coding, VVC)) или кодекам.

В одном из репрезентативных подходов раскрытая технология может использоваться для обеспечения способа обработки видеоданных. Этот способ содержит этапы, на  
20 которых: кодируют видеоблок видео в виде кодированного видеоблока; формируют отсчеты реконструкции из кодированного видеоблока; выполняют операции ограничения на отсчетах реконструкции, при которой параметр ограничения, используемый в операции ограничения, является функцией индекса ограничения и битовой глубины отсчетов реконструкции или битовой глубины отсчетов видеоблока; применяют  
25 нелинейный адаптивный контурный фильтр к результату операции ограничения; и формируют кодированное представление видео с использованием кодированного видеоблока.

При другом подходе раскрытая технология может использоваться для обеспечения способа обработки видеоданных. Этот способ содержит этапы, на которых  
30 осуществляют синтаксический анализ (парсинг) кодированного представления видео для кодированного видеоблока, представляющего видеоблок видео; формируют отсчеты реконструкции видеоблока из кодированного видеоблока; выполняют операцию ограничения на отсчетах реконструкции, причем параметр ограничения, используемый в операции ограничения, является функцией индекса ограничения и битовой глубины  
35 отсчетов реконструкции или битовой глубины видеоблока; и применяют нелинейный адаптивный контурный фильтр к результату операции ограничения для формирования конечного декодированного видеоблока.

При другом подходе раскрытая технология может использоваться для обеспечения способа обработки видеоданных. Этот способ содержит этапы, на которых выполняют  
40 преобразование между кодированным представлением видео, содержащим одну или более видеообластей, и видео; и определяют параметр ограничения для фильтрации реконструкции видеоблока видеообласти с использованием нелинейного адаптивного контурного фильтра, причем определение основано на кодированной информации видео, и/или видеообласти, и/или видеоблока.

45 При другом подходе раскрытая технология может использоваться для обеспечения способа обработки видеоданных. Этот способ содержит этапы, на которых выполняют преобразование между кодированным представлением видео, содержащим одну или более видеообластей, и видео; и определяют параметр ограничения для фильтрации

реконструкции видеоблока видеообласти с использованием нелинейного адаптивного контурного фильтра, при этом параметр ограничения является функцией цветового формата представления.

5 При другом подходе раскрытая технология может использоваться для обеспечения способа обработки видеоданных. Этот способ содержит этапы, на которых выполняют преобразование между кодированным представлением видео, содержащим одну или более видеообластей, и видео; и определяют параметр ограничения для фильтрации реконструкции видеоблока видеообласти с использованием нелинейного адаптивного контурного фильтра, при этом параметр ограничения зависит от того, применяется ли  
10 внутриконтурное переформирование (in-loop reshaping, ILR) для реконструкции видеоблока на основании представления видеоблока в первой области и во второй области и/или масштабирования остатка цветности видеоблока цветности.

При другом подходе раскрытая технология может использоваться для обеспечения способа обработки видеоданных. Этот способ содержит этапы, на которых выполняют  
15 преобразование между кодированным представлением видео, содержащим одну или более видеообластей, и видео, причем кодированное представление содержит первую побочную информацию, которая обеспечивает параметр ограничения для фильтрации реконструкции видеоблока видеообласти с использованием нелинейного адаптивного контурного фильтра; при этом первая побочная информация сообщается вместе со  
20 второй побочной информацией, указывающей коэффициенты фильтра, используемые в нелинейном адаптивном контурном фильтре.

При другом подходе раскрытая технология может использоваться для обеспечения способа обработки видеоданных. Этот способ содержит этап, на котором выполняют преобразование между кодированным представлением видео, содержащим одну или  
25 более видеообластей, и видео, причем кодированное представление содержит побочную информацию, указывающую множество наборов параметров ограничения для фильтрации реконструкции видеоблока видеообласти с использованием нелинейного адаптивного контурного фильтра.

При другом подходе раскрытая технология может использоваться для обеспечения  
30 способа обработки видеоданных. Этот способ содержит этап, на котором выполняют преобразование между кодированным представлением видео, содержащим одну или более видеообластей, и видео; при этом кодированное представление содержит побочную информацию, которая обеспечивает один или более параметров ограничения для фильтрации реконструкции видеоблока цветности видеообласти с использованием  
35 нелинейного адаптивного контурного фильтра, причем указанный один или более параметров ограничения зависят от цветового формата.

При другом подходе раскрытая технология может использоваться для обеспечения способа обработки видеоданных. Этот способ содержит этап, на котором выполняют преобразование между кодированным представлением видео, содержащим одну или  
40 более видеообластей, и видео, при этом кодированное представление содержит побочную информацию, которая обеспечивает параметр ограничения для фильтрации реконструкции видеоблока видеообласти с использованием адаптивного контурного фильтра; при этом на этапе выполнения формируют отфильтрованный видеоблок путем применения операции ограничения к разностям отсчетов на уровне видеообласти.

45 При еще одном репрезентативном подходе вышеописанный способ реализуется в виде исполняемого процессором кода и сохраняется на считываемом компьютере носителе программ.

При еще одном репрезентативном подходе раскрывается устройство, выполненное

с возможностью или способное действовать для выполнения описанного выше способа. Устройство может содержать процессор, который программируется для реализации этого способа.

В еще одном репрезентативном подходе один или более из раскрытых выше способов могут быть реализацией на стороне кодера или реализацией на стороне декодера.

Представленные выше и другие подходы и признаки раскрытой технологии описываются более подробно на чертежах, в описании и формуле изобретения

Краткое описание чертежей

Фиг. 1 - примерная блок-схема кодера для видеокодирования.

Фиг. 2А, 2В и 2С - примеры форм основанных на геометрическом преобразовании адаптивных контурных фильтров (geometry transformation-based adaptive loop filter, GALF).

Фиг. 3 - пример блок-схемы последовательности выполнения операций для принятия решения кодером GALF.

Фиг. 4А-4D - пример вычислений суботчетов лапласианов для классификации адаптивных контурных фильтров (adaptive loop filter, ALF).

Фиг. 5 - пример формы фильтра сигнала яркости.

Фиг. 6 - пример деления на области с последовательностью разрешения 800x480 (Wide Video Graphic Array, WVGA).

Фиг. 7 - пример блок-схемы последовательности выполнения операций декодирования с помощью переформирования.

Фиг. 8А-8С - блок-схемы последовательности выполнения операций примерных способов обработки видеосигналов в соответствии с некоторыми реализациями раскрытой технологии.

Фиг. 9 - блок-схема последовательности выполнения операций примерного способа обработки видеосигналов в соответствии с некоторыми реализациями раскрытой технологии.

Фиг. 10А и 10В - блок-схемы примеров аппаратной платформы для реализации технологии обработки видеосигналов в представленном документе.

Осуществление изобретения

Благодаря растущему спросу на видео с повышенным разрешением, способы и методы видеокодирования получили широкое распространение в современной технологии. Видеокодеки обычно содержат электронную схему или программное обеспечение, которые сжимают или разворачивают цифровые видеосигналы и непрерывно совершенствуются, чтобы обеспечивать повышенную эффективность кодирования. Видеокодек преобразует несжатый видеосигнал в сжатый формат или наоборот. Между качеством видео, объемом данных, используемых для представления видео (определяется битовой скоростью), сложностью алгоритмов кодирования и декодирования, чувствительностью к потере данных и ошибкам, простотой редактирования, произвольным доступом и сквозной задержкой (запаздыванием) существуют сложные взаимосвязи. Сжатый формат обычно подчиняется стандарту технических требований к видеосжатию, например, стандарту высокоэффективного видеокодирования (High Efficiency Video Coding, HEVC) (также известному как стандарт H.265 или MPEG-H Часть 2), стандарту универсального видеокодирования (Versatile Video Coding, VVC), который должен быть завершен, или другим текущим и/или будущим стандартам видеокодирования.

В некоторых вариантах осуществления будущие технологии видеокодирования применяются, используя ссылочное программное обеспечение, известное как Joint Exploration Model (JEM). Согласно JEM, предсказание, основанное на субблоках,

применяется в нескольких инструментах кодирования, таких как афинное предсказание, альтернативное временное предсказание вектора движения (alternative temporal motion vector prediction, ATMVP), пространственно-временное предсказание вектора движения (spatial-temporal motion vector prediction, STMVP), двунаправленный оптический поток (bi-directional optical flow, BIO), преобразование с повышением частоты кадров (Frame-Rate Up Conversion, FRUC), локально адаптивное разрешение вектора движения (Locally Adaptive Motion Vector Resolution, LAMVR), компенсация движения с наложенными блоками (Overlapped Block Motion Compensation, ОБМС), компенсация локальной подсветки (Local Illumination Compensation, LIC) и улучшение вектора движения на стороне декодера (Decoder-side Motion Vector refinement, DMVR).

Варианты осуществления раскрытой технологии могут применяться к существующим стандартам видеокодирования (например, HEVC, H.265) и к будущим стандартам, стандартам для улучшения характеристик рабочего цикла. Заголовки разделов используются в настоящем документе для улучшения чтения описания и никоим образом не ограничивают обсуждение или варианты осуществления (и/или реализации) только соответствующими разделами.

### 1 Примеры получения суботсчетов цветового пространства и цветности

Цветовое пространство, также известное как цветовая модель (или цветовая система), является абстрактной математической моделью, которая просто описывает диапазон цветов как наборы из  $n$  элементов, обычно из 3 или 4 значений или цветовых компонентов (например, RGB). По существу говоря, цветовое пространство является развитием координатной системы и субпространства.

Для видеосжатия наиболее часто используемыми цветовыми пространствами являются YCbCr и RGB.

YCbCr, Y'CbCr или Y Pb/Cb Pr/Cr также записываются как YCBCR или Y'CBCR, также является семейством цветовых пространств, используемых как часть продвижения цветного изображения в системах видео и цифровой фотографии. Y' является компонентом яркости, а CB и CR являются сине-разностными и красно-разностными цветовыми компонентами. Y' (с основным цветом) отличается от Y, который является яркостью, означая, что интенсивность света кодируется нелинейно, основываясь на гамма-скорректированных основных цветах RGB.

Суботсчет цветности является практическим применением кодируемых изображений, реализуя меньшее разрешение для информации о цветности, чем для информации о яркости, отдавая предпочтение меньшей разрешающей способности человеческой визуальной системы при различении цветов, по сравнению с яркостью.

#### 1.1 Цветовой формат 4:4:4

Каждый из трех компонентов Y'CbCr имеет одну и ту же частоту отсчетов, таким образом, никакого суботсчета цветности не существует. Эта схема иногда используется в высококачественных пленочных сканерах и при окончательном монтаже в кинематографии.

#### 1.2 Цветовой формат 4:2:2

Два компонента цветности подвергаются отсчетам с половинной частотой отсчетов цветности, например, с половинным разрешением цветности по горизонтали. Это сокращает полосу частот несжатого видеосигнала на одну треть почти с отсутствием визуального отличия.

#### 1.3 Цветовой формат 4:2:0

При формате 4:2:0 горизонтальные отсчеты удваиваются во сравнению с форматом 4:1:1, но поскольку в этой схеме каналы Cb и Cr подвергаются отсчетам только для

каждой чередующейся строки, разрешение по вертикали становится половинным. Частота данных, таким образом, является одной и той же. Каждый из Cb и Cr подвергается отсчету с коэффициентом 2 как по горизонтали, так и по вертикали. Существуют три варианта схемы 4:2:0, каждый из которых имеет различное

5 горизонтальное и вертикальное расположение.

- В MPEG-2 Cb и Cr совмещаются по горизонтали. Cb и Cr располагаются между пикселями в вертикальном направлении (располагаются в порядке поступления).

- В JPEG/JFIF, H.261 и MPEG Cb и Cr располагаются в порядке поступления, на полпути между чередующимися отсчетами яркости.

10 - В 4:2:0 DV Cb и Cr размещаются совместно в горизонтальном направлении. В вертикальном направлении они располагаются совместно на чередующихся строках.

## 2 Примеры потока кодирования типичного видеокодека

На фиг. 1 представлена примерная блок-схема кодера VVC. Как показано на фиг.

1, поток видеокodирования может начинаться с входного видео, которое подвергается

15 внутрикадровому предсказанию и/или оценке движения/компенсации движения (motion estimation / motion compensation, ME/MC). Эти операции используют обратную связь от

реконструированной копии ранее кодированного участка видео. Выходные сигналы внутрикадрового предсказания и/или ME/MC обрабатываются по-другому через

20 операцию преобразования (transform, T), сопровождаемую операцией квантования (quantization, Q), которая статистически кодируется в выходное кодированное

представление. В контуре обратной связи кодированное представление (выходной сигнал блока Q) может проходить через операцию инверсного квантования (inverse

quantization, IQ), сопровождаемую операцией инверсного преобразования (inverse transform, IT), чтобы сформировать отсчеты реконструкции кодированных видеоблоков.

25 Реконструированные отсчеты могут обрабатываться дополнительно посредством различных операций "контурной" фильтрации для формирования опорных отсчетов

или опорных блоков или опорных изображений, используемых для дополнительного кодирования. Цепь процессов контурной фильтрации содержит три блока контурной

30 фильтрации: деблокирующий фильтр (deblocking filter, DF), блок адаптивного смещения отсчетов (sample adaptive offset, SAO) и ALF. В отличие от DF, который использует

заранее определенные фильтры, SAO и ALF используют первоначальные отсчеты текущего изображения, чтобы уменьшить среднеквадратичные ошибки между

первоначальными отсчетами и реконструированными отсчетами, добавляя смещение и применяя фильтр с конечной импульсной характеристикой (finite impulse response,

35 FIR), соответственно, с помощью информации кодированной стороны, сообщающей смещения и коэффициенты фильтра. ALF располагается на последнем этапе обработки

каждого изображения и может рассматриваться как инструмент, пытающийся захватить и зафиксировать артефакты, созданные на предыдущих этапах.

На стороне декодера несколько операций кодирования выполняются в обратном

40 порядке, чтобы сформировать декодированные и реконструированные отсчеты видео. Например, обращаясь к фиг. 1, декодер может синтаксически анализировать

кодированное изображение, которое является результатом статистического кодирования, и получить кодированные блоки видео, которые затем проходят операцию инверсного

45 квантования (inverse quantization, IQ) и операцию инверсного преобразования (inverse transformation, IT), чтобы сформировать пробы реконструкции видеоблока.

Окончательный декодированный видеоблок может быть сформирован, применяя операции контурной фильтрации, как описано выше в отношении контура обратной связи видеокодера.

### 3 Примеры основанного на преобразовании геометрии адаптивного контурного фильтра в JEM

Согласно JEM, применяется адаптивный контурный фильтр на основе преобразования геометрии (geometry transformation-based adaptive loop filter, GALF) с адаптацией фильтра, основанной на блоках. Для компонента яркости выбирается один из 25 фильтров для каждого блока 2x2, основываясь на направлении и активности локальных градиентов.

#### 3.1 Примеры формы фильтра

Согласно JEM, для компонента яркости могут быть выбраны до трех ромбовидных форм фильтра (как показано на фиг. 2А, 2В и 2С для ромба 5x5, ромба 7x7 и ромба 9x9, соответственно). Индекс сообщается на уровне изображения, чтобы указать форму фильтра, используемого для компонента яркости. Для компонента цветности на изображении всегда используется форма ромб 5x5.

##### 3.1.1 Классификация блоков

Каждый блок 2x2 классифицируется как один из 25 классов. Индекс С классификации получают, основываясь на его направленности D и квантованном значении активности  $\hat{A}$ , как показано ниже:

$$C = 5D + \hat{A}. \quad (1)$$

Для вычисления D и  $\hat{A}$ , используя одномерный лапласиан, сначала вычисляют градиенты горизонтального, вертикального и двух диагональных направлений:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|, \quad (2)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|, \quad (3)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (4)$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (5)$$

Индексы i и j относятся к координатам верхнего левого отсчета в блоке 2x2 и R(i, j) указывает реконструированный отсчет в координате (i, j).

Затем максимальное и минимальное значения D градиента горизонтального и вертикального направления устанавливаются следующим образом:

$$g_{h,v}^{\max} = \max(g_h, g_v), \quad g_{h,v}^{\min} = \min(g_h, g_v), \quad (6)$$

и максимальное и минимальное значения градиента двух диагональных направлений устанавливаются следующим образом:

$$g_{d0,d1}^{\max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{\min} = \min(g_{d0}, g_{d1}), \quad (7)$$

Чтобы получить значение направленности D, эти значения сравниваются относительно друг друга и с двумя порогами  $t_1$  и  $t_2$ .

Этап 1: Если оба  $g_{h,v}^{\max} \leq t_1 \cdot g_{h,v}^{\min}$  и  $g_{d0,d1}^{\max} \leq t_1 \cdot g_{d0,d1}^{\min}$  верны, D устанавливается равным 0.

Этап 2. Если  $g_{h,v}^{\max}/g_{h,v}^{\min} > g_{d0,d1}^{\max}/g_{d0,d1}^{\min}$ , продолжить с этапа 3; в противном случае, продолжить с этапа 4.

Этап 3. Если  $g_{h,v}^{\max} > t_2 \cdot g_{h,v}^{\min}$ , D установить равным 2; в противном случае, D установить равным 1.

Этап 4. Если  $g_{d0,d1}^{\max} > t_2 \cdot g_{d0,d1}^{\min}$ , D установить равным 4; в противном случае, D установить равным 3.

Значение активности A вычисляется следующим образом:

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}). \quad (8)$$

A дополнительно квантуется в диапазоне от 0 до 4, включительно, и квантованное значение обозначается как  $\hat{A}$ .

Для обоих компонентов цветности в изображении никакой способ классификации не применяется, то есть, для каждого компонента цветности применяется один единственный набор коэффициентов ALF.

### 3.1.2 Геометрические преобразования коэффициентов фильтра

Перед фильтрацией каждого блока  $2 \times 2$  к коэффициентам  $f(k,l)$ , в зависимости от значений, вычисленных для этого блока, применяются геометрические преобразования, такие как вращение или диагональный или вертикальный переворот. Это эквивалентно применению этих преобразований к отсчетам в области поддержки фильтра. Идея заключается в том, чтобы сделать различные блоки, к которым применяется ALF, более схожими, выравнивая их направленности.

Три геометрических преобразования, в том числе, диагональный, вертикальный переворот и вращение, вводятся следующим образом:

25 Диагональный:  $f_D(k,l) = f(l,k)$ ,

Вертикальный переворот:  $f_V(k,l) = f(k, K - 1 - l)$ ,

30 Вращение:  $f_R(k,l) = f(K - 1 - l, k)$ .

(9)

Здесь, K – размер фильтра и  $0 \leq k, l \leq K-1$  являются координатами коэффициентов, так что местоположение (0, 0) находится в верхнем левом углу, а расположение (K – 1, K – 1) находится в нижнем правом углу. Преобразования применяются к коэффициентам фильтра  $f(k, l)$  в зависимости от градиентных значений, вычисленных для этого блока. Взаимосвязь между преобразованием и четырьмя градиентами четырех направлений обобщается в таблице 1.

Таблица 1. Представление градиента, вычисленного для одного блока и преобразования

Значение градиента	Преобразование
$g_{d2} < g_{d1}$ и $g_h < g_v$	Нет преобразования
$g_{d2} < g_{d1}$ и $g_v < g_h$	Диагональное
$g_{d1} < g_{d2}$ и $g_h < g_v$	Вертикальный переворот
$g_{d1} < g_{d2}$ и $g_v < g_h$	Вращение

### 3.1.3 Сообщение параметров фильтра

Согласно JEM, параметры фильтра GALF сообщаются для первого CTU, то есть, после заголовка фрагмента и перед параметрами SAO первого CTU. Могут сообщаться до 25 наборов коэффициентов фильтра яркости. Чтобы сократить битовую служебную сигнализацию, могут объединяться коэффициенты фильтра другой классификации.

Кроме того, коэффициенты GALF опорных изображений запоминаются и разрешаются для повторного использования в качестве коэффициентов GALF текущего изображения. Текущее изображение может выбираться, чтобы использовать коэффициенты GALF, запомненные для опорных изображений, и обойти сигнализацию коэффициентов GALF.

5 В этом случае сигнализируется только индекс в одном из опорных изображений и запомненные коэффициенты GALF указанного опорного изображения применяются для текущего изображения.

Для поддержки временного предсказания GALF, ведется список возможных кандидатов наборов фильтра GALF. В начале декодирования новой последовательности список кандидатов пустой. После декодирования одного изображения в список кандидатов может быть добавлен соответствующий набор фильтров. Когда размер списка кандидатов достигает максимального разрешенного значения (например, 6 в текущем JEM), новый набор фильтров переписывает самый старый набор в порядке декодирования и, то есть, для обновления списка кандидатов применяется правило

15 "первым вошел-первым вышел" (first-in-first-out, FIFO). Чтобы избежать дублирования, набор может добавляться в список, только когда соответствующее изображение ин использует временное предсказание GALF. Для поддержки временной масштабируемости существуют многочисленные списки возможных кандидатов наборов фильтров и каждый список связывается с временным уровнем. Более конкретно, матрица,

20 назначенная временным индексом уровня (TempIdx) может состоять из наборов фильтров ранее декодированных изображений с равным или меньшим индексом TempIdx. Например, k-ая матрица назначается быть связанной с TempIdx, равным k, и она содержит только наборы фильтров из изображений с TempIdx, меньшим или равным k. После кодирования определенного изображения наборы фильтров, связанные с

25 изображением, будут использоваться для обновления этих матриц, связанных с равным или более высоким TempIdx.

Временное предсказание коэффициентов GALF используется для кадров с межкадровым кодированием, чтобы минимизировать служебную сигнализацию. Для кадров с внутрикадровым кодированием временное предсказание недоступно и каждому

30 классу назначается набор из 16 фиксированных фильтров. Для указания использования фиксированных фильтров для каждого класса сообщается флаг и, если требуется, индекс выбранного фиксированного фильтра. Даже когда для заданного класса выбирается фиксированный фильтр, коэффициенты  $f(k, l)$  адаптивного фильтра продолжают

35 применяться к реконструированному изображению, являются суммой обоих наборов коэффициентов.

Процесс фильтрации компонента яркости может управляться на уровне CU. Флаг устанавливается, чтобы указать, применяется ли GALF к компоненту яркости. Для компонента цветности, применяется ли GALF или нет, указывается только на уровне

40 изображения.

#### 3.1.4 Процесс фильтрации

На стороне декодера, когда для блока разрешается GALF, каждый отсчет  $R(i, j)$  внутри блока фильтруется, приводя в результате к значению отсчета  $R'(i, j)$ , как показано ниже, где  $L$  обозначает длину фильтра,  $f_{m,n}$  представляет коэффициент фильтра и  $f(k,$

45  $l)$  обозначает декодированные коэффициенты фильтра.

$$R'(i,j) = \sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k, l) \times R(i+k, j+l) \quad (10)$$

### 3.1.5 Процесс определения параметров фильтра на стороне кодера

Общий процесс нахождения решения для кодера в отношении GALF показан на фиг. 3. Для отсчетов яркости каждого CU кодер принимает решение о том, применяется ли GALF, и в заголовок фрагмента вводится соответствующий флаг сигнализации. Для отсчетов цветности решение о применении фильтра принимается на основе уровня изображения, а не на уровне CU. Дополнительно, GALF для цветности изображения проверяется, только когда для изображения разрешается GALF яркости.

4 Примеры адаптивного контурного фильтра на основе преобразования геометрии для VVC

Текущая конструкция GALF для VVC имеет следующие основные изменения по сравнению с используемой в JEM:

1) Форма адаптивного фильтра удалена. Для компонента яркости разрешается только форма фильтра 7x7, и форма фильтра 5x5 разрешается для компонента цветности.

2) Временное предсказание параметров ALF и предсказание для фиксированных фильтров оба удаляются.

3) Для каждого STU устанавливается однобитовый флаг, независимо от того, разрешается или запрещается ALF.

4) Вычисление индекса класса выполняется на уровне 4x4 вместо 2x2. Кроме того, как предложено в JVET-L0147, для классификации ALF используется способ вычисления лапласиана с суботсчетами. Более конкретно, нет необходимости вычислять горизонтальные/вертикальные/диагональные под 45/135 градусов градиенты для каждого отсчета внутри одного блока. Вместо этого, используются суботсчеты 1:2.

5 Примеры основанного на области фильтрации адаптивного контурного фильтра в AVS2

ALF является последним этапом контурной фильтрации. В этом процессе существуют два этапа. Первый этап – получение коэффициентов фильтра. Чтобы подготовить коэффициенты фильтра, кодер классифицирует реконструированные пиксели компонента яркости в 16 областях и один набор коэффициентов фильтра подготавливается для каждой категории, используя уравнения Винера-Хопфа, чтобы минимизировать среднеквадратичную ошибку между первоначальным кадром и реконструированным кадром. Чтобы понизить избыточность между этими 16 наборами коэффициентов фильтра, кодер будет адаптивно их соединять, основываясь на рабочих характеристиках искажения-скорости. В его максимуме 16 различных наборов фильтра могут быть назначены компоненту яркости и только один – компоненту цветности. Второй этап – принятие решения о фильтре, который содержит как уровень кадра, так и уровень LCU. Сначала кодер решает, выполняется ли адаптивная контурная фильтрация на уровне кадра. Если уровень кадра ALF включен, то кодер далее решает, выполняется ли ALF на уровне LCU.

#### 5.1 Форма фильтра

Формой фильтра, принятой в AVS-2, является крестообразная форма 7x7, наложенная на квадратную форму 3x3, такая, как показано на фиг. 5, как для яркости, так и для компонент цветности. Каждый квадрат на фиг. 5 соответствует отсчету. Поэтому для получения отфильтрованного значения для отсчета в позиции C8 используются в общей сложности 17 отсчетов. Учитывая служебные издержки на передачу коэффициентов, фильтр с симметричной точкой используется только с девятью остающимися коэффициентами, {C0, C1..., C8}, что сокращает количество коэффициентов фильтра наполовину, а также количество умножений при фильтрации. Фильтр с симметричной точкой может также уменьшать наполовину вычисления для одного отфильтрованного

отсчета, например, только 9 операций умножения и 14 операций сложения для одного фильтрованного отсчета.

## 5.2 Адаптивное слияние, основанной на области

Чтобы приспособиться к различным ошибкам кодирования, AVS 2 приспособливает основанные на области адаптивные контурные мультифильтры для компонента яркости. Компонент яркости делится на 16 примерно равного размера основных областей, где каждая основная область выравнивается с границами самого большого блока кодирования (largest coding unit, LCU), как показано на фиг. 6, и для каждой области получают один фильтр Винера. Чем больше используется фильтров, тем больше снижаются искажения, но количество битов, используемых для кодирования этих коэффициентов, увеличивается вместе с количеством фильтров. Чтобы достигнуть наилучшей характеристики "искажения-скорость", эти области могут сливаться в меньшее количество областей большего размера, которые совместно используют те же самые коэффициенты фильтра. Чтобы упростить процесс слияния, каждой области назначается индекс в соответствии с модифицированным порядком Гильберта, основанным на корреляциях с предшествующими изображениями. Две области со следующими друг за другом индексами могут объединяться, основываясь на затратах на показатель "искажение-скорость".

Информация о преобразовании между областями должна сообщаться декодеру. В AVS 2 количество основных областей используется для представления результатов объединения и коэффициенты фильтра последовательно сжимаются в соответствии с их порядком областей. Например, когда {0, 1}, {2, 3, 4}, {5, 6, 7, 8, 9} и левые основные области объединяются в одну область, соответственно, только три целых числа кодируются, чтобы представить это объединенное преобразование, то есть, 2, 3, 5.

## 5.3 Сообщение побочной информации

Также используются многочисленные флаги переключения. Флаг переключателя последовательности `adaptive_loop_filter_enable` используется для управления тем, применяется ли адаптивный контурный фильтр для всей последовательности. Флаги переключателя изображения `picture_alf_enable[i]` управляют тем, применяется ли ALF для соответствующего *i*-ого компонента изображения. Соответствующие флаги LCU-уровня и коэффициенты фильтра для этого компонента цветности будут передаваться, только если разрешается `picture_alf_enable[i]`. Флаги уровня LCU, `lcu_alf_enable[k]`, управляют тем, разрешается ли ALF для соответствующего *k*-ого LCU и чередуются ли они в данных фрагмента. Решение о других флагах регулируемых уровней полностью основывается на затратах на показатель "искажения-скорость". Высокая гибкость дополнительно заставляет ALF повышать эффективность кодирования гораздо более значительно.

В некоторых вариантах осуществления и для компонента яркости может иметься до 16 наборов коэффициентов фильтра.

В некоторых вариантах осуществления и для каждого компонента цветности (Cb и Cr) может передаваться один набор коэффициентов фильтра.

### 6 GALF в VTM-4

Согласно VTM4.0, процессе фильтрации адаптивным контурным фильтром выполняется следующим образом:

$$O(x, y) = \sum_{(i,j)} w(i, j) \cdot I(x + i, y + j) \quad (11)$$

$I(x + i, y + j)$  где отсчеты являются входными отсчетами,  $O(x, y)$  является

фильтрованной выходным отсчетом (то есть, результатом фильтрации), и  $w(i, j)$  обозначает коэффициенты фильтра. На практике в VTM4.0 это осуществляется, используя целочисленную арифметику для вычислений точности фиксированной точки:

$$O(x, y) = \left( \sum_{i=-\frac{L}{2}}^{\frac{L}{2}} \sum_{j=-\frac{L}{2}}^{\frac{L}{2}} w(i, j) \cdot I(x + i, y + j) + 64 \right) \gg 7 \quad (12)$$

где  $L$  обозначает длину фильтра, и где  $w(i, j)$  - коэффициенты фильтра в точке с фиксированной точностью.

7 Фильтрация с нелинейный адаптивным контурным фильтром (ALF)

#### 10 7.1 Переформулирование фильтрации

Уравнение (11) может быть переформулировано без влияния на эффективность кодирования в следующем выражении:

$$O(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot (I(x + i, y + j) - I(x, y)) \quad (13)$$

15 Здесь,  $w(i, j)$  являются теми же самыми коэффициентами фильтра, что и в уравнении (11) [за исключением  $w(0, 0)$  который равен 1 в уравнении (13), в то время как в уравнении (11) он равен  $1 - \sum_{(i,j) \neq (0,0)} w(i, j)$  .

#### 20 7.2 Модифицированный фильтр

Используя эти приведенные выше формулы (13) для фильтра, мы можем легко ввести нелинейность, чтобы сделать ALFA более эффективным, используя простую функцию ограничения, чтобы уменьшить влияние значений соседних отсчетов  $I(x + i, y + j)$  (когда они слишком отличаются от текущего значения  $I(x, y)$ ) фильтруемого отсчета.

В этом предложении фильтр ALFA модифицируется следующим образом:

$$25 O'(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot K(I(x + i, y + j) - I(x, y), k(i, j)) \quad (14)$$

Здесь  $K(d, b) = \min(b, \max(-b, d))$  - функция ограничения, и  $k(i, j)$  являются параметрами ограничения, которые зависят от коэффициента фильтра  $(i, j)$ . Кодер выполняет оптимизацию, чтобы обнаружить лучшее значение  $k(i, j)$ .

30 В реализации JVET-N0242 параметры ограничения  $k(i, j)$  определяются для каждого фильтра ALFA, одно значение ограничения сообщается для каждого коэффициента фильтра. Это означает, что в битовом потоке для каждого фильтра яркости может сообщаться до 12 значений ограничения и до 6 значений ограничения могут сообщаться для фильтра цветности.

35 Чтобы ограничить затраты на сигнализацию и сложность кодера, мы ограничиваем оценку значений ограничения малым набором возможных значений. Согласно предложению, мы используем только 4 фиксированных значения, которые являются одинаковыми для мозаичных групп INTER и INTRA.

40 Поскольку расхождение локальных различий часто бывает более высоким для яркости (Luma), чем для насыщенности цвета (Chroma), мы используем два различных набора для фильтров Luma и Chroma. Мы также вводим максимальное значение отсчета (здесь 1024 для 10-разрядной битовой глубины) в каждом наборе, с тем, чтобы ограничение можно было отключить, если оно не требуется.

45 Наборы значений ограничения, используемых в тестах JVET-N0242, приводятся в таблице 2. 4 значения были выбраны путем примерно одинакового разбиения в логарифмической области полного диапазона значений отсчета (кодированных на 10 битах) для Luma, и диапазона от 4 до 1024 для Chroma.

Более точно, таблица значений ограничений для Luma была получена с помощью

следующей формулы:

$$\underline{\text{AlfClip}}_L = \left\{ \text{round} \left( \left( \left( M \right)^{\frac{1}{N}} \right)^{N-n+1} \right) \text{ for } n \in 1..N \right\} \text{ с } M=2^{10} \text{ и } N=4.$$

5 Точно также, таблицы значений ограничения для Chroma получают согласно следующей формуле:

$$\underline{\text{AlfClip}}_C = \left\{ \text{round} \left( A \cdot \left( \left( \frac{M}{A} \right)^{\frac{1}{N-1}} \right)^{N-n} \right) \text{ for } n \in 1..N \right\} \text{ с } M=2^{10}, N=4 \text{ и } A=4.$$

Таблица 2

Разрешенные значения ограничения

	Группа мозаик INTRA/INTER
ЯРКОСТЬ	{ 1024, 181, 32, 6 }
ЦВЕТНОСТЬ	{ 1024, 161, 25, 4 }

15 Выбранные величины ограничения кодированы в элементе синтаксиса “alf\_data”, используя схему кодирования Golomb, соответствующую индексу значения ограничения в приведенной выше таблице 2. Эта схема кодирования является такой же, как схема кодирования для индекса фильтра.

8 Контурное переформирование (in-loop reshaping, ILR) в JVET-M0427

Контурное переформирование (in-loop reshaping, ILR) также известно как преобразование Luma с масштабированием Chroma (Luma Mapping with Chroma Scaling, LMCS).

25 Основная идея контурного переформирования (ILR) состоит в преобразовании начального (в первой области) сигнала (сигнала предсказания/реконструкции) во вторую область (область восстановления формы).

30 Формирователь контурного восстановления формы для яркости реализуется осуществляется как пара таблиц поиска (look-up table, LUT), но только одна из этих двух таблиц LUT должна сообщаться, поскольку другая может быть вычислена на основании сообщенной LUT. Каждая LUT является одномерной 10-разрядной таблицей преобразования с 1024 входами (1D-LUT). Одна LUT является прямой LUT, FwdLUT, которая преобразует входные кодовые значения яркости  $Y_i$  в измененные значения:  $Y_r = \text{FwdLUT}[Y_i]$ . Другая LUT является обратной LUT, InvLUT, которая преобразует измененные кодовые значения  $Y_r$  в  $Y_i$ :  $Y_i = \text{InvLUT}[Y_r]$ . ( $Y_i$  представляет реконструированные значения  $Y_i$ ).

### 8.1 Модель PWL

40 Концептуально, кусочно-линейная (piece-wise linear, PWL) модель осуществляется следующим образом:

Пусть  $x_1$ ,  $x_2$  являются двумя входными поворотными точками, а  $y_1$ ,  $y_2$  являются их соответствующими выходными поворотными точками для одного куска. Выходное значение  $y$  для любого входного значения  $x$  между  $x_1$  и  $x_2$  может быть интерполировано следующим уравнением:

$$45 \quad y = ((y_2 - y_1) / (x_2 - x_1)) * (x - x_1) + y_1$$

В реализации фиксированной точки уравнение может быть переписано как:

$$y = ((m * x + 2\text{FP\_PREC} - 1) \gg \text{FP\_PREC}) + c$$

Здесь  $m$  является скаляром,  $c$  - смещение и  $\text{FP\_PREC}$  – постоянное значение,

указывающее точность.

Заметим, что в программном обеспечении SE-12, модель PWL используется для предварительного вычисления FwdLUT с 1024 входами и таблиц преобразования InvLUT; но модель PWL также позволяет реализациям вычислять идентичные значения преобразования непрерывно, без предварительного вычисления LUT.

## 8.2 Тест SE12-2 на 4-ом совещании по VVC

### 8.2.1 Восстановление формы Luma

Тест 2 контурного восстановления формы для яркости (то есть, SE12-2 в предложении) обеспечивает последовательность обработки с меньшей сложностью, что также исключает задержку декодирования для внутрикадрового поблочного предсказания при межкадровой реконструкции фрагментов. Внутрикадровое предсказание выполняется в области с восстановленной формой для фрагментов как межкадрового, так и внутрикадрового предсказания.

Внутрикадровое предсказание всегда выполняется в области переформирования, независимо от типа фрагмента. При таком построении, внутрикадровое предсказание может начинаться сразу после того, как сделана предыдущая реконструкция TU. Такое построение может также обеспечивать унифицированный процесс для внутрикадрового режима вместо того, чтобы быть зависимым от фрагмента. На фиг. 7 представлена блок-схема процесса декодирования SE12-2, основанного на режиме.

SE12-2 также тестирует 16-тикусочные кусочно-линейные (PWL) модели для яркости и цветности масштабируя остаток вместо использования 32-хкусочных моделей PWL из SE12-1.

Межкадровая реконструкция фрагментов с помощью контурного устройства восстановления формы в SE12-2 (более светло окрашенные блоки указывают сигнал в области переформирования: остаток яркости ; внутрикадрово предсказанная яркость и внутрикадрово реконструированная яркость).

### 8.2.2 Масштабирование остатка цветности, зависимо от яркости

Масштабирование остатка цветности, зависимо от яркости, является мультипликативным процессом, осуществляемым с операцией с целыми числами с фиксированной точкой. Масштабирование остатка цветности компенсирует взаимодействие сигнала яркости с сигналом цветности. Масштабирование остатка цветности применяется на уровне TU. Более конкретно, применяется нижеследующее:

- Для внутрикадрового предсказания реконструированная яркость усредняется.
- Для межкадрового предсказания предсказание яркости усредняется.

Среднее значение используется для идентификации индекса в модели PWL. Индекс идентифицирует масштабный коэффициент cScaleInv. Остаточная цветность умножается на этот коэффициент.

Заметим, что коэффициент масштабирования цветности вычисляется из предсказанных значений яркости при прямом преобразовании, а не из реконструированных значений яркости.

### 8.2.3 Сообщение побочной информации ILR

Параметры (в настоящий момент) посылаются в заголовке мозаичной группы (подобно ALF). Они, по сообщениям, занимают 40-100 битов. Нижеследующая спецификация основана на версии 9 JVET-L1001. Добавленный синтаксис выделяется ниже шрифтом, набранным курсивом.

Синтаксис RBSP с установленными в 7.3.2.1 параметрами последовательности

	seq_parameter_set_rbsp() {	Дескриптор
	sps_seq_parameter_set_id	ue(v)
5	...	
	sps_triangle_enabled_flag	u(1)
	sps_ladf_enabled_flag	u(1)
	if ( sps_ladf_enabled_flag ) {	
	sps_num_ladf_intervals_minus2	u(2)
10	sps_ladf_lowest_interval_qp_offset	se(v)
	for( i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++ ) {	
	sps_ladf_qp_offset[ i ]	se(v)
	sps_ladf_delta_threshold_minus1[ i ]	ue(v)
	}	
15	}	
	sps_resaper_enabled_flag	u(1)
	rbsp_trailing_bits()	
	}	

Синтаксис RBSP с установленными в 7.3.2.1 параметрами последовательности

20	seq_parameter_set_rbsp() {	Дескриптор
	sps_seq_parameter_set_id	ue(v)
	...	
	sps_triangle_enabled_flag	u(1)
	sps_ladf_enabled_flag	u(1)
25	if ( sps_ladf_enabled_flag ) {	
	sps_num_ladf_intervals_minus2	u(2)
	sps_ladf_lowest_interval_qp_offset	se(v)
	for( i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++ ) {	
	sps_ladf_qp_offset[ i ]	se(v)
30	sps_ladf_delta_threshold_minus1[ i ]	ue(v)
	}	
	}	
	sps_resaper_enabled_flag	u(1)
	rbsp_trailing_bits()	
35	}	

40

45

## Синтаксис заголовка мозаичной группы General в 7.3.3.1

	tile_group_header() {	Дескриптор
	...	
5	if( num_tiles_in_tile_group_minus1 > 0 ) {	
	offset_len_minus1	ue(v)
	for( i = 0; i < num_tiles_in_tile_group_minus1; i++ )	
	entry_point_offset_minus1[ i ]	u(v)
	}	
10	if( sps_resaper_enabled_flag ) {	
	tile_group_resaper_model_present_flag	u(1)
	if( tile_group_resaper_model_present_flag )	
	tile_group_resaper_model ( )	
	tile_group_resaper_enable_flag	u(1)
15	if ( tile_group_resaper_enable_flag && (!( qtbt_dual_tree_intra_flag && tile_group_type == I ) ) )	
	tile_group_resaper_chroma_residual_scale_flag	u(1)
	}	
	byte_alignment( )	
20	}	

Добавляют новую таблицу синтаксиса модели восстановления формы мозаичной группы:

	tile_group_resaper_model ( ) {	Дескриптор
	reshaper_model_min_bin_idx	ue(v)
25	reshaper_model_delta_max_bin_idx	ue(v)
	reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
	for ( i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++ ) {	
	reshape_model_bin_delta_abs_CW [ i ]	u(v)
	if( reshaper_model_bin_delta_abs_CW[ i ] > 0 )	
30	reshaper_model_bin_delta_sign_CW_flag[ i ]	u(1)
	}	
	}	

В семантику RBSP набора параметров последовательности General добавляют следующую семантику:

35 sps\_resaper\_enabled\_flag, равный 1, указывает, что формирователь восстановления формы используется в кодированной видеопоследовательности (coded video sequence, CVS). sps\_resaper\_enabled\_flag, равный 0, указывает, что формирователь восстановления формы не используется в CVS.

40 В синтаксис заголовка мозаичной группы добавляют следующую семантику tile\_group\_resaper\_model\_present\_flag, равный 1, указывает, что tile\_group\_resaper\_model() присутствует в заголовке не присутствует в заголовке мозаичной группы. tile\_group\_resaper\_model\_present\_flag, равный 0, указывает, что tile\_group\_resaper\_model() не присутствует в заголовке мозаичной группы. Когда tile\_group\_resaper\_model\_present\_flag не присутствует, подразумевается, что он должен быть 45 равен 0.

tile\_group\_resaper\_enabled\_flag, равный 1, указывает, что формирователь восстановления формы включен для текущей мозаичной группы. tile\_group\_resaper\_enabled\_flag, равный 0, указывает, что формирователь восстановления формы не включен

для текущей мозаичной группы. Когда `tile_group_reshaper_enable_flag` не присутствует, подразумевается, что он должен быть равен 0.

`tile_group_reshaper_chroma_residual_scale_flag`, равный 1, указывает, что остаточное масштабирование цветности включено для текущей мозаичной группы. `tile_group_reshaper_chroma_residual_scale_flag`, равный 0, указывает, что остаточное масштабирование цветности не включено для текущей мозаичной группы. Когда `tile_group_reshaper_chroma_residual_scale_flag` не присутствует, подразумевается, что он должен быть равен 0.

Добавляют синтаксис `tile_group_reshaper_model( )`

`reshape_model_min_bin_idx` указывает индекс минимального бина (куска), который должен использоваться в процессе конструирования формирователя восстановления формы. Значение `reshape_model_min_bin_idx` должно быть в диапазоне от 0 до `MaxBinIdx`, включительно. Значение `MaxBinIdx` должно быть равно 15.

`reshape_model_delta_max_bin_idx` указывает индекс максимального разрешенного бина (куска) `MaxBinIdx` минус максимальный индекс бина, который должен использоваться в процессе конструирования формирователя восстановления формы. Значение `reshape_model_delta_max_bin_idx` устанавливается равным `MaxBinIdx - reshape_model_delta_max_bin_idx`.

`reshaper_model_bin_delta_abs_cw_prec_minus1` плюс 1 указывает количество битов, используемых для представления синтаксиса `reshape_model_bin_delta_abs_CW[ i ]`.

`reshape_model_bin_delta_abs_CW[ i ]` указывает абсолютное значение кодового слова дельта для *i*-ого бина.

`reshaper_model_bin_delta_sign_CW_flag[ i ]` указывает знак `reshape_model_bin_delta_abs_CW[ i ]` следующим образом:

– Если `reshaper_model_bin_delta_sign_CW_flag[ i ]` равен 0, соответствующая переменная `RspDeltaCW[ i ]` имеет положительное значение.

– В противном случае, (`reshaper_model_bin_delta_sign_CW_flag[ i ]` не равен 0), соответствующая переменная `RspDeltaCW[ i ]` имеет отрицательное значение.

Когда `reshaper_model_bin_delta_sign_CW_flag[ i ]` не присутствует, подразумевается, что он должен быть равен 0.

Переменная `RspDeltaCW[ i ]` равна  $(1 \ll \text{BitDepth}_Y) * \text{reshape\_model\_bin\_delta\_abs\_CW}[ i ]$ ;

Переменную `RspCW[ i ]` получают на следующих этапах:

Переменная `OrgCW` устанавливается равной  $(1 \ll \text{BitDepth}_Y) / (\text{MaxBinIdx} + 1)$ .

– Если `reshaper_model_min_bin_idx <= i <= reshaper_model_max_bin_idx`  
`RspCW[ i ] = OrgCW + RspDeltaCW[ i ]`.

– В противном случае, `RspCW[ i ] = 0`.

Значение `RspCW[ i ]` должно быть в диапазоне от 32 до  $2 * \text{OrgCW} - 1$ , если значение `BitDepthY` равно 10.

Переменные `InputPivot[ i ]` с *i* в диапазоне от 0 до `MaxBinIdx + 1`, включительно, получают следующим образом:

`InputPivot[ i ] = i * OrgCW`

Переменную `ReshapePivot[ i ]` с *i* в диапазоне от 0 до `MaxBinIdx + 1`, включительно, переменные `ScaleCoeff[ i ]` и `InvScaleCoeff[ i ]` с *i* в диапазоне от 0 до `MaxBinIdx`, включительно, получают следующим образом:

```

    shiftY = 14
    ReshapePivot[ 0 ] = 0;
    for( i = 0; i <= MaxBinIdx ; i++) {
5      ReshapePivot[ i + 1 ] = ReshapePivot[ i ] + RspCW[ i ]
      ScaleCoeff[ i ] = ( RspCW[ i ] * ( 1 << shiftY) + ( 1 << (Log2(OrgCW) - 1))) >>
      (Log2(OrgCW))
10     if ( RspCW[ i ] == 0 )
        InvScaleCoeff[ i ] = 0
    else
        InvScaleCoeff[ i ] = OrgCW * ( 1 << shiftY) / RspCW[ i ]
15  }

```

Переменную ChromaScaleCoeff[ i ] с i в диапазоне от 0 до MaxBinIdx , включительно, получают следующим образом:

```

20  ChromaResidualScaleLut[64] = { 16384, 16384, 16384, 16384, 16384, 16384, 16384, 8192,
    8192, 8192, 8192, 5461, 5461, 5461, 5461, 4096, 4096, 4096, 4096, 3277, 3277, 3277, 3277,
    2731, 2731, 2731, 2731, 2341, 2341, 2341, 2048, 2048, 2048, 1820, 1820, 1820, 1638, 1638,
    1638, 1638, 1489, 1489, 1489, 1489, 1365, 1365, 1365, 1365, 1260, 1260, 1260, 1260, 1170,
    1170, 1170, 1092, 1092, 1092, 1092, 1024, 1024, 1024, 1024};
25  shiftC = 11
    - if ( RspCW[ i ] == 0 )
        ChromaScaleCoeff [ i ] = ( 1 << shiftC) - В противном случае, (RspCW[ i ] != 0),
    ChromaScaleCoeff[ i ] = ChromaResidualScaleLut[RspCW[ i ] >> 1]

```

#### 8.2.4 Использование ILR

30 На стороне кодера каждое изображение (или мозаичная группа) сначала преобразуется в область переформирования. И весь процесс кодирования выполняется в области переформирования. Для внутрикадрового предсказания соседний блок находится в области переформирования; для межкадрового предсказания опорные блоки (формируемые из первоначальной области из декодированного буфера изображения) сначала преобразуется в область переформирования. Затем остаток формируется и кодируется в битовый поток.

После того, как все изображение (мозаичная группа) закончило кодирование/ декодирование, отсчеты в области переформирования преобразуются в первоначальную область и затем применяют деблокирующий фильтр и другие фильтры.

40 Прямое переформирование в сигнал предсказания запрещается в следующих случаях:

- Текущий блок имеет внутрикадровое кодирование

- Текущий блок кодирован как CPR (current picture referencing (текущее опорное изображение), также называемое intra block copy, IBC (копия блока с внутрикадровым кодированием))

- 45 - Текущий блок кодирован как объединенный в режиме межкадрового-внутрикадрового предсказания (combined inter-intra mode, СИП), и прямое переформирование запрещается для блока с внутрикадровым предсказанием.

#### 9 Недостатки существующих реализаций

Проект нелинейного ALF (NLALF) в JVET-N0242 имеет следующие проблемы:

(1) Он был разработан для цветового формата 4:2:0. Для цветового формата 4:4:4 компоненты яркости и цветности могут обладать схожим значением. Как лучше применять NLALF - неизвестно.

(2) Значения ограничения разработаны для 10-разрядного случая. Как определить NLALF для другой битовой глубины, пока еще не изучено.

(3) Взаимодействие способа контурного переформирования и NLALF не было изучено.

10 Примерные способы усовершенствований нелинейной адаптивной контурной фильтрации

Варианты осуществления раскрытой здесь технологии преодолевают недостатки существующих реализаций, обеспечивая, таким образом, видеокодирование с более высокой эффективностью кодирования. Нелинейная адаптивная контурная фильтрация, основанная на раскрытой технологии, может улучшить как существующие, так и будущие стандарты видеокодирования и объясняется в последующих примерах, описанных для различных реализаций. Примеры раскрытой технологии, представленные ниже, объясняют общие концепции и не предназначены интерпретироваться как ограничение. В примере, если явно не указывается обратное, могут сочетаться различные признаки, описанные в этих примерах.

1. Предполагается, что параметры (например, параметры ограничения, определенные в таблице 2), используемой в NLALF, могут зависеть от кодированной информации.

20 а. Предполагается, что параметры (например, параметры ограничения, определенные в таблице 2), используемой в NLALF, могут зависеть от индекса временного уровня/низкого флага проверки задержки/опорных изображений.

2. Многочисленные наборы параметров NLALF могут определяться или сообщаться.

25 а. Альтернативно, дополнительно, когда многочисленные наборы параметров NLALF сообщаются, они могут сообщаться в блоке данных, таком как набор параметров адаптации (Adaptation Parameter Set, APS)/заголовок мозаичной группы/блоки видеоданных.

б. В одном примере параметры NLALF сообщаются предсказываемым способом.

30 i. Например, один набор параметров NLALF, сообщенный в одном блоке данных (таким как APS или мозаичная группа или фрагмент), предсказывается другим набором параметров NLALF, сообщаемым в том же самом блоке данных.

ii. Например, один набор параметров NLALF, сообщаемый в одном блоке данных (таким как APS или мозаичная группа или фрагмент), предсказывается другим набором параметров NLALF, сообщенным в другом блоке данных.

35 3. Предполагается, что параметры (например, параметры ограничения, определенные в таблице 2), используемой в NLALF, могут зависеть от битовой глубины реконструированных отсчетов перед применением NLALF.

40 а. Альтернативно, предполагается, чтобы параметры (например, параметры ограничения, определенные в таблице 2), используемой в NLALF, могут зависеть от входной битовой глубины отсчетов перед кодированием/декодированием.

б. В одном примере параметры для заданной битовой глубины могут быть получены из тех, которые были назначены для другой битовой глубины.

i. В одном примере, операции сдвига, соответствующие битовой глубине, могут быть применимы, чтобы получить параметры для заданной битовой глубины.

45 4. Предполагается, что параметры (например, параметры ограничения, определенные в таблице 2), используемой в NLALF, могут зависеть от цветового формата представления.

а. В одном примере, для случая RGB, существует параметр с одним и тем же индексом

для компонента G цветности и для компонентов цветности V/R.

5. Предполагается, что параметры (например, параметры ограничения, определенные в таблице 2), используемой в NLALF, могут зависеть от того, применяется ли способ контурного переформирования (ILR).

5 а. В одном примере параметры могут отличаться, когда ILR разрешается или запрещается.

6. Предполагается запоминать параметры фильтра (такие как коэффициенты фильтра) и параметры NLALF (такие как параметры ограничения) вместе.

а. В одном примере они оба могут быть запомнены в APS.

10 б. В одном примере, когда один блок видеоданных (например, STU/область/мозаичная группа) использует коэффициенты фильтра использования, связанные с одним APS, связанные параметры NLALF также могут быть использованы.

15 в. Альтернативно, для кодирования/декодирования одного блока видеоданных (например, STU/область/мозаичная группа), когда предсказание из коэффициентов фильтра, связанных с одним APS, разрешается, связанные параметры NLALF также могут использоваться, чтобы предсказать параметры NLALF для одного блока видеоданных из того же самого APS.

7. То, как управлять NLALF для цветных компонентов цветности, может зависеть от цветового формата.

20 а. В одном примере, для заданного цветового формата (такого как 4:4:4) два цветных компонента могут использовать различные параметры NLALF.

8. Предполагается, чтобы ограничение в ALF могло включаться или выключаться на уровне последовательности, уровне изображения, уровне фрагмента, уровне мозаичной группы, уровне мозаики, уровне STU, уровне CU или на уровне блоков.

25 а. Например, включать ли ограничение в ALF, может сообщаться устройству декодирования, так как в SP, PPS, заголовке фрагмента, заголовке мозаичной группы, мозаике, STU, CU, или в блоке.

30 Примеры, описанные выше, могут быть включены в контекст способов, описанных ниже, например, способов 810-840, которые могут быть реализованы в видеодекодере или в видеокодере.

На фиг. 8А представлена блок-схема последовательности выполнения операций примерного способа обработки видеоданных. Согласно способу 810, на этапе 812 кодируют видеоблок видео как кодированный видеоблок. Способ 810 дополнительно содержит этап 813, на котором формируют из кодированного видеоблока отсчеты реконструкции. Способ 810 дополнительно содержит этапе 814, на котором выполняют операцию ограничения на отсчетах реконструкции, причем параметр ограничения, используемый в операции ограничения, является функцией индекса ограничения и битовой глубиной отсчетов реконструкции или битовой глубиной отсчетов видеоблока. Способ 810 дополнительно содержит этап 815, на котором нелинейный адаптивный контурный фильтр применяют к выходному результату операции ограничения. Способ 40 810 дополнительно содержит этап 816, на котором формируют кодированное представление видео, используя кодированный видеоблок.

На фиг. 8В представлена блок-схема последовательности выполнения операций примерного способа обработки видеоданных. Способ 820 содержит этап 822, на котором синтаксически анализируют кодированное представление видео для кодированного видеоблока, представляющего видеоблок видео. Способ 820 дополнительно содержит этап 823, на котором формируют отсчеты реконструкции видеоблока из кодированного видеоблока. Способ 820 содержит этап 824, на котором выполняют операцию

ограничения на отсчетах реконструкции, причем параметр ограничения, используемый в операции ограничения, является функцией индекса ограничения и битовой глубины отсчетов реконструкции или битовой глубины видеоблока. Способ 820 дополнительно содержит этап 825, на котором применяют нелинейный адаптивный контурный фильтр к выходному результату операции ограничения, чтобы сформировать окончательный декодированный видеоблок.

На фиг. 8С представлена блок-схема последовательности выполнения операций примерного способа обработки видеоданных. Способ 830 содержит этап 832, на котором выполняют преобразование между кодированным представлением видео, содержащего одну или более видеообластей, и видео. Способ 830 дополнительно содержит этап 834, на котором определяют параметр ограничения для фильтрации реконструкции видеоблока видеообласти, используя нелинейный адаптивный контурный фильтр. В некоторых реализациях определение основано на кодированной информации видео и/или видеообласти и/или видеоблока. В некоторых реализациях параметр ограничения является функцией цветового формата представления. В некоторых реализациях параметр ограничения зависит от того, применяется ли контурное переформирование (ILR) для реконструкции видеоблока, основанного на представлении видеоблока в первой области и второй области и/или для определения масштаба остатка цветности видеоблока цветности.

На фиг. 9 показана блок-схема последовательности выполнения операций способа обработки видеоданных. Способ 840 содержит выполнение преобразования между кодированным представлением видео, содержащего одну или более видеообластей и видео. В некоторых реализациях кодированное представление содержит первую побочную информацию, которая обеспечивает параметр ограничения для фильтрации реконструкции видеоблока видеообласти, используя нелинейный адаптивный контурный фильтр, и первая побочная информация сообщается вместе со второй побочной информацией, указывающей коэффициенты фильтра, используемые в нелинейном адаптивном контурном фильтре. В некоторых реализациях кодированное представление содержит побочную информацию, указывающую многочисленные наборы ограничения параметров для фильтрации реконструкции видеоблока видеообласти, используя нелинейный адаптивный контурный фильтр. В некоторых реализациях кодированное представление содержит побочную информацию, которая обеспечивает один или более параметров ограничения для фильтрации реконструкции видеоблока цветности видеообласти, используя нелинейный адаптивный контурный фильтр, в котором один или более параметров ограничения зависят от цветового формата. В некоторых реализациях кодированное представление содержит побочную информацию, которая обеспечивает параметр ограничения для фильтрации реконструкции видеоблока видеообласти, используя адаптивный контурный фильтр, в котором выполнение содержит формирование отфильтрованного видеоблока, применяя операцию ограничения к разностям отсчетов на уровне видеообласти.

#### 11 Примерные реализации раскрытой технологии

На фиг. 10А представлена блок-схема устройства 900 обработки видеоданных. Устройство 900 может использоваться для реализации одного или более описанных здесь способов. Устройство 900 может быть осуществлено в смартфоне, планшете, компьютере, приемнике Интернета вещей (IoT) и так далее. Устройство 900 может содержать один или более процессоров 902, одну или более памятей 904 и аппаратные средства 906 обработки видеоданных. Процессор(-ы) 902 может быть выполнен с возможностью осуществления одного или более способов (в том числе, но не

ограничиваясь только этим, способа 800), описанных в настоящем документе. Память (памяти) 904 может использоваться для хранения данные и управляющей программы, используемых для реализации описанных здесь способов и технологий. Аппаратные средства 906 обработки видеоданных могут использоваться для осуществления в схемах аппаратных средств некоторых способов, описанных в настоящем документе.

На фиг. 10В представлен другой пример блок-схемы системы обработки видеоданных, в которой могут быть осуществлены раскрытые технологии. На фиг. 10В представлена блок-схема, показывающая пример системы 4100 обработки видеоданных, в которой могут быть реализованы различные раскрытые здесь технологии. Различные реализации могут содержать некоторые или все компоненты системы 4100. Система 4100 может содержать вход 4102 для приема видеоконтента. Видеоконтент может приниматься в необработанном или в несжатом формате, например, 8-или 10-разрядных многокомпонентных пиксельных значений, или может приниматься в сжатом или кодированном формате. Вход 4102 может представлять собой сетевой интерфейс, периферийный шинный интерфейс или интерфейс запоминающего устройства. Примеры сетевого интерфейса содержат проводные интерфейсы, такие как Ethernet, пассивная оптическая сеть (passive optical network, PON) и т.д., и беспроводные интерфейсы раздела, такие как Wi-Fi или сотовые интерфейсы.

Система 4100 может содержать компонент 4104 кодирования, который может осуществить различное кодирование или способы кодирования, описанных в настоящем документе. Компонент 4104 кодирования может понижать среднюю скорость передачи видео от входа 4102 до выхода компонента 4104 кодирования, чтобы создавать кодированное представление видео. Технологии кодирования поэтому иногда называют видеосжатием или технологиями транскодирования видео. Выходной сигнал компонента 4104 кодирования может либо запоминаться, либо передаваться через присоединенное связанное соединение, что представлено компонентом 4106. Представление запомненного или переданного (или кодированного) битового потока, принятого на входе 4102, может использоваться компонентом 4108 для формирования пиксельных значений или отображаемого видео, которое посылается на интерфейс 4110 дисплея. Процесс формирования пригодного для просмотра пользователем видео из представления битового потока иногда называют видеораспаковкой. Дополнительно, хотя некоторые операции видеообработки упоминаются как операции или инструменты "кодирования", следует понимать, что инструменты или операции кодирования используются в кодере, а соответствующие инструменты или операции декодирования, которые дают результаты, обратные кодированию, должны выполняться декодером.

Примеры интерфейса периферийной шины или интерфейса дисплея могут содержать универсальную последовательную шину (universal serial bus, USB) или мультимедийный интерфейс высокой четкости (high definition multimedia interface, HDMI) или Displayport, и так далее. Примеры интерфейсов запоминающего устройства содержат интерфейсы SATA (serial advanced technology attachment), PCI, IDE и т. п. Технологии, описанные в настоящем документе, могут быть реализованы в различных электронных устройствах, таких как мобильные телефоны, ноутбуки, смартфоны или другие устройства, способные выполнять обработку цифровых данных и/или отображение видео.

Некоторые варианты осуществления раскрытой технологии содержат принятие решения или определение разрешения на инструмент или режим обработки видеоданных. В примере, когда инструмент или режим обработки видеоданных разрешаются, кодер будет использовать или реализовывать инструмент или режим при обработке блока видео, но может не обязательно модифицировать результирующий битовый поток,

основанный на использовании инструмента или режима. То есть, преобразование из блока видео в представление битового потока видео будет использовать инструмент или режим обработки видеоданных, когда это разрешается, основываясь на принятом решении или определении. В другом примере, когда инструмент или режим обработки видеоданных разрешаются, декодер будет обрабатывать битовый поток, зная что битовый поток был модифицирован, основываясь на инструменте или режиме обработки видеоданных. То есть, преобразование из представления битового потока видео в блок видео будет выполняться, используя инструмент или режим обработки видеоданных, который был разрешен на основе решения или определения.

Некоторые варианты осуществления раскрытой технологии содержат принятие решения или определение о запрете инструмента или режима обработки видеоданных. В примере, в котором инструмент или режим обработки видеоданных запрещается, кодер не будет использовать инструмент или режим при преобразовании блока видео в представление битового потока видео. В другом примере, когда инструмент или режим обработки видеоданных запрещается, декодер будет обрабатывать битовый поток, зная, что битовый поток не модифицировался, используя инструмент или режим обработки видеоданных, который был запрещен, основываясь на решении или определении.

В настоящем документе термин "обработка видеоданных" может относиться к видеокодированию, видеодекодированию, видеосжатию или видеораспаковке. Например, алгоритмы видеосжатия могут быть применены во время преобразования из пиксельного представления видео в соответствующее представление битового потока или наоборот. Представление битового потока текущего видеоблока может, например, соответствовать битам, которые располагаются совместно или разбросаны в различных местах внутри битового потока, как определяется синтаксисом. Например, макроблок может кодироваться с точки зрения остаточных значений преобразованных и кодированных ошибок, а также использования битов в заголовках и других полях в битовом потоке.

В некоторых вариантах осуществления способы видеокодирования могут быть реализованы, используя устройство, которое реализуется на аппаратной платформе как описано со ссылкой на фиг. 10А или 10В.

Различные технологии и варианты осуществления могут быть описаны, используя нижеследующий формат, основанный на пунктах .

Первый набор пунктов описывает признаки и подходы раскрытых технологий, перечисленных в предыдущем разделе.

1. Способ обработки видеоданных, содержащий этапы, на которых: определяют, основываясь на кодированной информации текущего видеоблока, набор параметров текущего видеоблока; и реконструируют, основываясь на выполнении операции нелинейной фильтрации, используя набор параметров, текущий блок, исходя из соответствующего представления битового потока.

2. Способ по п. 1, в котором операция нелинейной фильтрации содержит нелинейную адаптивную контурную фильтрацию.

3. Способ по п. 1 или 2, в котором набор параметров содержит по меньшей мере одно значение ограничения для компонента яркости или компонента цветности текущего видеоблока.

4. Способ по п. 3, в котором операция нелинейной фильтрации основана на цветном формате компонента цветности.

5. Способ по любому из пп. 1-3, в котором кодированная информация содержит временный индекс уровня, низкий флаг проверки задержки или одно или более опорных

изображений.

6. Способ по любому из пп. 1-3, в котором кодированная информация содержит битовую глубину реконструированных отсчетов, полученных до операции нелинейной фильтрации.

5 7. Способ по любому из пп. 1-3, в котором кодированная информация содержит цветовой формат представления.

8. Способ по любому из пп. 1-3, в котором кодированная информация содержит указание о применении способа контурного переформирования (in-loop reshaping, ILR).

9. Способ по любому из пп. 1-3, в котором соответствующее представление битового  
10 потока содержит многочисленные наборы параметров, которые содержат определенный набор параметров, и в котором многочисленные наборы параметров сообщаются в наборе параметров адаптации (Adaptation Parameter Set, APS), заголовке мозаичной группы или одном или более блоках видеоданных.

10. Способ по любому из пп. 1-3, в котором соответствующее представление битового  
15 потока содержит набор параметра адаптации (APS), который содержит набор параметров и один или более коэффициентов фильтра, связанных с операцией нелинейной фильтрации.

11. Способ по п. 1 или 2, в котором набор параметров содержит одно или более значений ограничения и в котором операция нелинейной фильтрации выполняется на  
20 уровне последовательности, уровне изображения, уровне фрагмента, уровне мозаичной группы, уровне мозаики, уровне блока дерева кодирования (coding tree unit, CTU), уровне блока кодирования (coding unit, CU) или уровне блока.

12. Устройство в видеосистеме, содержащее процессор и память долговременного хранения с командами, содержащимися в ней, в котором команды после их исполнения  
25 процессором, вызывают осуществление процессором способа по любому из пп. 1-11.

13. Компьютерный программный продукт, хранящийся на считываемом компьютере носителе длительного хранения, причем компьютерный программный продукт содержит управляющую программу для выполнения способа по любому из пп. 1-11.

Второй набор пунктов описывает некоторые признаки и подходы раскрытых  
30 технологий, перечисленные в предыдущем разделе, в том числе, например, примерные реализации 1 и 3-5.

1. Способ обработки видеоданных, содержащий этапы, на которых: кодируют видеоблок видео как кодированный видеоблок; формируют отсчеты реконструкции из кодированного видеоблока; выполняют операцию ограничения на отсчетах  
35 реконструкции, в котором параметр ограничения, используемый в операции ограничения, является функцией индекса ограничения и битовой глубины отсчетов реконструкции или битовой глубиной отсчетов видеоблока; применяют нелинейный адаптивный контурный фильтр к выходному результату операции ограничения; и формируют кодированное представление видео, используя кодированный видеоблок.

40 2. Способ по п. 1, в котором индекс ограничения сообщается в кодированном представлении.

3. Способ обработки видеоданных, содержащий этапы, на которых: синтаксически анализируют кодированное представление видео для кодированного видеоблока, представляющего видеоблок видео; формируют отсчеты реконструкции видеоблока  
45 из кодированного видеоблока; выполняют операцию ограничения на отсчетах реконструкции, в котором параметр ограничения, используемый в операции ограничения, является функцией индекса ограничения и битовой глубины отсчетов реконструкции или битовой глубиной видеоблока; и применяют нелинейный адаптивный

контурный фильтр к выходному результату операции ограничения, чтобы сформировать окончательный декодированный видеоблок.

4. Способ по п. 3, в котором индекс ограничения определяется, основываясь, по меньшей мере, на поле в кодированном представлении.

5 5. Способ по п. 3, в котором индекс ограничения определяется, используя заданное правило.

6. Способ по п. 1 или 3, в котором функция индекса ограничения и битовая глубина отсчетов реконструкции или битовая глубина видеоблока является такой, что функция возвращает различные значения для заданного значения индекса ограничения, основываясь на битовой глубине отсчетов реконструкции или битовой глубине видеоблока.

7. Способ по п. 1 или 3, в котором преобразование между индексом ограничения и параметром ограничения зависит от битовой глубины отсчета реконструкции или битовой глубины видеоблока.

15 8. Способ по п. 1 или 3, в котором первое значение ограничения, соответствующее первому индексу ограничения для данной битовой глубины, получают, основываясь на втором значении ограничения, соответствующем второму индексу ограничения для другой битовой глубины.

9. Способ по п. 8, в котором операция смещения, использующая другую битовую глубину, применяется для получения параметра ограничения для данной битовой глубины.

10. Способ по любому из пп. 1-9, в котором кодированное представление содержит параметр ограничения, который управляет верхней или нижней границей двух разностей отсчетов, используемых в нелинейном адаптивном контурном фильтре.

25 11. Способ обработки видеоданных, содержащий этапы, на которых: выполняют преобразование между кодированным представлением видео, содержащего одну или более видеообластей и видео; и определяют параметр ограничения для фильтрации реконструкции видеоблока видеообласти, используя нелинейный адаптивный контурный фильтр, и в котором определение основано на кодированной информации видео и/или видеообласти и/или видеоблока.

12. Способ по п. 11, в котором кодированная информация содержит временный индекс уровня.

13. Способ по п. 11, в котором кодированная информация содержит низкий флаг проверки задержки.

35 14. Способ по п. 11, в котором кодированная информация содержит одно или более опорных изображений.

15. Способ по любому из пп. 11-14, в котором видеообласть содержит видеоизображение.

16. Способ по любому из пп. 11-14, в котором видеоблок содержит блок кодирования.

40 17. Способ по любому из пп. 11-16, в котором параметр ограничения управляет верхней или нижней границей двух разностей отсчетов, используемых в нелинейном адаптивном контурном фильтре.

18. Способ обработки видеоданных, содержащий этапы, на которых: выполняют преобразование между кодированным представлением видео, содержащим одну или более видеообластей и видео; и определяют параметр ограничения для фильтрации реконструкции видеоблока видеообласти, используя нелинейный адаптивный контурный фильтр, и в котором параметр ограничения является функцией цветового формата представления.

19. Способ по п. 18, в котором для цветового формата RGB параметра ограничения имеет один и тот же индекс для зеленого цветового компонента и для синего или красного цветового компонента.

20. Способ обработки видеоданных, содержащий этапы, на которых: выполняют преобразование между кодированным представлением видео, содержащего одну или более видеообластей и видео; и определяют параметр ограничения для фильтрации реконструкции видеоблока видеообласти, используя нелинейный адаптивный контурный фильтр, и в котором параметр ограничения зависит от того, применяется ли контурное переформирование (in-loop reshaping, ILR) для реконструкции видеоблока, основываясь на представлении видеоблока в первой области и во второй области и/или на масштабировании остатка цветности видеоблока цветности.

21. Способ по любому из пп. 1-21, в котором параметр ограничения соответствует значению ограничения для компонента яркости или компонента цветности видеоблока.

22. Способ по любому из пп. 1-21, в котором способ дополнительно содержит этапы, на которых во время преобразования формируют отфильтрованный видеоблок, применяя нелинейный адаптивный контурный фильтр к реконструкции видеоблока, и используя отфильтрованный видеоблок для определения предсказания другого видеоблока видео.

23. Способ по любому из пп. 1-22, в котором выполнение преобразования содержит формирование кодированного представления из видео.

24. Способ по любому из пп. 1-22, в котором выполнение преобразования содержит формирование видео из кодированного представления.

25. Устройство в видеосистеме, содержащее процессор и память долговременного хранения с записанными в ней командами, в котором команды после их исполнения процессором вызывают осуществление процессором способа по любому из пп. 1-24.

26. Компьютерный программный продукт, хранящийся на считываемом компьютере носителе долговременного хранения, причем компьютерный программный продукт содержит управляющую программу для выполнения способа по любому из пп. 1-24.

Третий набор пунктов описывает некоторые признаки и подходы раскрытых технологий, перечисленных в предшествующем разделе, в том числе, например, пример реализаций 2 и 6-8.

1. Способ обработки видеоданных, содержащий этапы, на которых: выполняют преобразование между кодированным представлением видео, содержащим одну или более видеообластей и видео, в котором, кодированное представление содержит первую побочную информацию, которая обеспечивает параметр ограничения для фильтрации реконструкции видеоблока видеообласти, используя нелинейный адаптивный контурный фильтр; в котором первая побочная информация сообщается вместе со второй побочной информацией, указывающей коэффициенты фильтра, используемые в нелинейном адаптивном контурном фильтре.

2. Способ по п. 1, в котором первая побочная информация и вторая побочная информация сообщаются в одном и том же наборе параметров адаптации.

3. Способ по п. 1, в котором, в случае, когда по меньшей мере некоторые из коэффициентов фильтра, связанные с набором параметров адаптации, используются блоком видеоданных, параметр ограничения, связанный с набором параметров адаптации, также используется блоком видеоданных.

4. Способ по п. 1, в котором в случае, когда предсказание, по меньшей мере, некоторых из коэффициентов фильтра, связанных с набором параметров адаптации, разрешается для преобразования блока видеоданных, параметр, связанный с набором

параметров адаптации, используется для предсказания другого параметра для другого блока видеоданных из набора параметров адаптации.

5. Способ по п. 3 или 4, в котором блок видеоданных является блоком дерева кодирования, видеообластью или мозаичной группой.

5 6. Способ по любому из пп. 1-5, в котором параметр соответствует значению ограничения для компонента яркости или компонента цветности видеоблока.

7. Способ обработки видеоданных, содержащий этапы, на которых: выполняют преобразование между кодированным представлением видео, содержащим одну или более видеообластей и видео, в котором кодированное представление содержит  
10 побочную информацию, указывающую многочисленные наборы параметров ограничения для фильтрации реконструкции видеоблока видеообласти, используя нелинейный адаптивный контурный фильтр.

8. Способ по п. 7, в котором побочная информация содержит многочисленные наборы параметров ограничения.

15 9. Способ по п. 7, в котором многочисленные наборы параметров ограничения известны кодирователю и декодеру и побочная информация содержит индекс в одном или более из многочисленных наборов параметров ограничения.

10. Способ по п. 7, в котором многочисленные наборы параметров ограничения включаются в блок видеоданных или в заголовок видеоблока.

20 11. Способ по п. 10, в котором блок видеоданных содержит набор параметров адаптации, мозаичную группу или фрагмент.

12. Способ по п. 8, в котором один из набора многочисленных параметров ограничения, сообщаемых в блоке данных, предсказывается другим набором параметров ограничения, сообщенных в блоке данных.

25 13. Способ по п. 7, в котором один набор из числа многочисленных наборов параметров ограничения, сообщаемых в блоке данных, предсказывается другим набором параметров ограничения, сообщаемым в другом блоке данных.

14. Способ обработки видеоданных, содержащий этапы, на которых: выполняют преобразование между кодированным представлением видео, содержащим одну или  
30 более видеообластей, и видео; в котором кодированное представление содержит побочную информацию, которая обеспечивает один или более параметров ограничения для фильтрации реконструкции видеоблока цветности видеообласти, используя нелинейный адаптивный контурный фильтр, в котором один или более параметров ограничения зависят от цветового формата.

35 15. Способ по п. 14, в котором для определенного цветового формата два компонента цветности используют различные параметры ограничения.

16. Способ по п. 15, в котором конкретным цветовым форматом является формат 4:4:4.

40 17. Способ по любому из пп. 1-16, в котором способ ополнительно содержит формирование во время преобразования отфильтрованного видеоблока, применяя нелинейный адаптивный контурный фильтр к реконструкции видеоблока и используя отфильтрованный видеоблок для определения предсказания другого видеоблока видео.

18. Способ обработки видеоданных, содержащий этапы, на которых: выполняют преобразования между кодированным представлением видео, содержащим одну или  
45 более видеообластей и видео, в котором, кодированное представление содержит побочную информацию, обеспечивающую параметр ограничения для фильтрации реконструкции видеоблока видеообласти, используя адаптивный контурный фильтр, в котором выполнение содержит формирование отфильтрованного видеоблока,

применяя операцию ограничения, чтобы сделать отсчет разностей на уровне видеообласти.

19. Способ по п. 18, в котором уровень видеообласти является уровнем последовательности, уровнем изображения, уровнем фрагмента, уровнем мозаичной группы, уровнем мозаики, уровнем блока дерева кодирования, уровнем блока кодирования или уровнем блока.

20. Способ по п. 18, в котором указание разрешить операции ограничения сообщается в наборе параметров последовательности (sequence parameter set, SPS), наборе параметров изображения (picture parameter set, PPS), заголовке фрагмента, заголовке мозаичной группы, мозаике, блоке дерева кодирования, блоке кодирования или блоке.

21. Способ по любому из пп. 1-20, в котором видеообласть является видеоизображением.

22. Способ по любому из пп. 1-20, в котором видеоблок является блоком кодирования или блоком преобразования или фрагментом или деревом кодирования или строкой дерева кодирования.

23. Способ по любому из пп. 1-22, в котором выполнение преобразования содержит формирование кодированного представления из текущего блока.

24. Способ по любому из пп. 1-22, в котором выполнение преобразования содержит формирование текущего блока от кодированного представления.

25. Устройство в составе видеосистеме, содержащее процессор и память долговременного хранения с хранящимися на ней командами, в котором команды при их исполнении процессором, вызывают осуществление процессором способа по любому из пп. 1-24.

26. Компьютерный программный продукт, хранящийся на считываемом компьютером носителе долговременного хранения, причем компьютерный программный продукт содержит управляющую программу для выполнения способа по любому из пп. 1-24.

Из вышесказанного должно быть понятно, что конкретные варианты осуществления раскрытой здесь технологии были описаны с целью иллюстрации, но в ней могут быть сделаны различные изменения, не отступая от объема защиты изобретения.

Соответственно, раскрытая здесь технология не ограничивается ничем, кроме как приложенной формулой изобретения.

Реализации предмета изобретения и функциональных операций, описанных в этом патентном документе, могут быть осуществлены в различных системах, цифровой электронной схмотехнике или в программном обеспечении, встроенных программах или аппаратных средствах, в том числе, в структурах, раскрытых в настоящем описании, и их структурных эквивалентах или в сочетании одной или более из них. Реализации предмета изобретения, описанного в настоящем описании, могут быть осуществлены как один или более компьютерных программных продуктов, то есть, как один или более модулей команд компьютерной программы, кодированных на физическом и считываемом компьютером носителе долговременного хранения для исполнения или управления работой устройства обработки данных. Считываемый компьютером носитель может быть машиночитаемым запоминающим устройством, машиночитаемым запоминающим носителем, составом веществ переносящих распространяющийся машиночитаемый сигнал или сочетанием одного или более из них. Термин "блок обработки данных" или "устройство обработки данных" охватывает все оборудование, устройства и машины для обработки данные, включая, например, программируемый процессор, компьютер или многочисленные процессоры или компьютеры. Устройство может содержать, в дополнение к аппаратным средствам, код, создающий среду

исполнения для компьютерной программы, например, код, который образует встроенные процессорные программы, стек протокола, систему управления базой данных, операционную систему или сочетание одного или более из них.

Компьютерная программа (также известная как программа, программное обеспечение, приложение, сценарий или код) может быть записана в любой форме языка программирования, включая скомпилированные или интерпретируемые языки, и может быть развернута в любой форме, включая как автономная программа, так и модуль, компонент, подпрограмма, или другой блок, пригодный для использования в компьютерной среде. Компьютерная программа не обязательно соответствует файлу в файловой системе. Программа может храниться в части файла, который хранит другие программы или данные (например, в одном или более сценариях, хранящихся в документе на языке с разметкой), в едином файле, назначенном для упомянутой программы, или в многочисленных скоординированных файлах (например, файлы, хранящие один или более модулей, подпрограммы или части кода). Компьютерная программа может быть развернута, чтобы исполняться на одном компьютере или на многочисленных компьютерах, которые располагаются в одном месте или распределены во множестве мест и соединяются системой связи.

Процессы и логические потоки, представленные в этом описании, могут выполняться одним или более программируемыми процессорами, исполняющими одну или более компьютерных программ для выполнения функций, работающих на входных данных и формирующих выходные данные. Процессы и логические потоки могут также выполняться и устройство может также быть реализовано как логическая схема специального назначения, например, FPGA (программируемая логическая интегральная схема) или ASIC (специализированная прикладная интегральная схема).

Процессоры, пригодные для выполнения компьютерной программы, содержат, например, универсальные и специального назначения микропроцессоры и любые один или более процессоров любого вида цифрового компьютера. Обычно процессор должен принимать команды и данные от постоянного запоминающего устройства или оперативной памяти или от них обоих. Существенными элементами компьютера являются процессор для выполнения команд и одно или более запоминающих устройств для хранения команд и данных. Обычно, компьютер будет также содержать или будет оперативно связан для приема данные или передачи данных или того и другого одному или более запоминающим устройствам большой емкости для хранения данных, например, магнитным дискам, магнито-оптическим дискам или оптическим дискам. Однако, компьютера необязательно должен иметь такие устройств. Считываемый компьютером носитель, пригодный для хранения команд компьютерной программы и данных, содержит любые формы энергонезависимой памяти, устройств носителей и запоминающих устройств, в том числе, например, устройства полупроводниковой памяти, например, EPROM, EEPROM и устройства флэш-памяти. Процессор и память могут дополняться или включаться в логические схемы специального назначения.

Подразумевается, что описание вместе с чертежами должно рассматриваться только как пример. Единственное число, как оно используется здесь, будет также включать в себя множественное число, если контекст явно не указывает иное. Дополнительно, использование "или" подразумевает использование "и/или", если контекст явно не указывает иное.

Хотя настоящий патентный документ содержит много специфических особенностей, они должны рассматриваться не как ограничения объема защиты какого-либо изобретения или того, что может быть заявлено, а скорее как описания признаков,

которые могут быть конкретными для частных вариантов осуществления частных изобретений. Некоторые признаки, которые описываются в этом патентном документе в контексте отдельных вариантов осуществления, могут также быть реализованы в сочетании в едином варианте осуществления. Наоборот, различные признаки, которые описываются в контексте единого варианта осуществления, могут также быть реализованы во многочисленных вариантах осуществления отдельно или в любом соответствующем субсочетании. Кроме того, хотя признаки могут быть описаны выше как действующие в некотором сочетании и даже первоначально быть заявленными как таковые, один или более признаков из заявленного сочетания могут в некоторых случаях быть извлечены из сочетания и заявленное сочетание может быть направлено в субсочетание или вариант субсочетания.

Аналогично, хотя операции показаны на чертежах в конкретном порядке, это не должно пониматься как требование, чтобы такие операции выполнялись в показанном конкретном порядке или в последовательном порядке или что все показанные операции должны быть выполнены, чтобы достигнуть требуемых результатов. Кроме того, разделение различных системных компонентов в вариантах осуществления, описанных в этом патентном документе, не должно пониматься как требование такого разделения во всех вариантах осуществления.

Были описаны только несколько реализаций и примеров и другие реализации, улучшения и вариации могут быть сделаны, основываясь на том, что описано и показано в этом патентном документе.

#### (57) Формула изобретения

1. Способ обработки видеоданных, содержащий этапы, на которых:

определяют, что к текущей видеообласти видео применяется операция нелинейной адаптивной контурной фильтрации, причем текущая видеообласть является блоком дерева кодирования;

получают по меньшей мере один первый индекс фильтрации для текущей видеообласти и получают один или более первых параметров фильтрации на основании указанного по меньшей мере одного первого индекса фильтрации;

получают один или более первых параметров ограничения из набора параметров ограничения на основании указанного по меньшей мере одного первого индекса фильтрации;

выполняют операцию ограничения, которая является частью операции нелинейной адаптивной контурной фильтрации, на основании указанного одного или более первых параметров ограничения, причем указанный один или более первых параметров ограничения используются для ограничения различий между контрольными значениями отсчетов текущего значения отсчета текущей видеообласти и текущим значением отсчета перед применением указанного одного или более первых параметров фильтрации; и

выполняют преобразование между текущей видеообластью и битовым потоком видео на основании операции нелинейной адаптивной контурной фильтрации;

при этом набор параметров ограничения создается на основании битовой глубины и индекса ограничения, основанного на заданной таблице, причем в заданной таблице значение одного параметра ограничения, соответствующего первому индексу ограничения для данной битовой глубины, получается на основании значения другого параметра ограничения, соответствующего первому индексу ограничения для другой битовой глубины.

2. Способ по п.1, в котором для получения значений параметров ограничения из

набора параметров ограничения используется операция сдвига.

3. Способ по п.1, в котором индекс ограничения связан с полем в битовом потоке.

4. Способ по п.1, в котором индекс ограничения определяется с использованием заданного правила.

5 5. Способ по п.1, в котором для набора параметров ограничения используются разные значения параметров ограничения для разной битовой глубины при заданном значении индекса ограничения.

6. Способ по п.1, в котором в заданной таблице значение одного параметра ограничения получается на основании применения операции сдвига к значению другого параметра ограничения.

7. Способ по п.1, в котором указанный по меньшей мере один первый индекс фильтрации получается на основании множества разностей отсчетов в различных направлениях.

8. Способ по п.7, в котором текущая видеообласть разбивается на множество видеообластей размером  $M \times M$ , и множество разностей отсчетов в различных направлениях получают для каждой субобласти видео размером  $M \times M$ , и где  $M$  равно 2 или 4.

9. Способ по п.8, в котором множество разностей отсчетов в различных направлениях получается на основании частоты взятия суботсчетов  $1:N$ , где  $N$  больше 1.

10. Способ по п.1, в котором указанный один или более первых параметров фильтрации и указанный один или более первых параметров ограничения получают на основании одной и той же идентификации набора параметров адаптации.

11. Способ по п.1, в котором на этапе преобразования кодируют текущую видеообласть в битовый поток.

12. Способ по п.1, в котором на этапе преобразования декодируют текущую видеообласть из битового потока.

13. Устройство обработки видеоданных, содержащее процессор и память длительного пользователя с командами, причем команды при их исполнении процессором вызывают выполнение процессором:

30 определения, что к текущей видеообласти видео применяется операция нелинейной адаптивной контурной фильтрации, причем текущая видеообласть является блоком дерева кодирования;

35 получения по меньшей мере одного первого индекса фильтрации для текущей видеообласти и получения одного или более первых параметров фильтрации на основании указанного по меньшей мере одного первого индекса фильтрации;

получения одного или более первых параметров ограничения из набора параметров ограничения на основании указанного по меньшей мере одного первого индекса фильтрации,

40 выполнения операции ограничения, которая является частью операции нелинейной адаптивной контурной фильтрации, на основании указанного одного или более первых параметров ограничения, причем указанный один или более первых параметров ограничения используются для ограничения различий между контрольными значениями отсчетов текущего значения отсчета текущей видеообласти и текущим значением отсчета перед применением указанного одного или более первых параметров фильтрации; и

45 выполнения преобразования между текущей видеообластью и битовым потоком видео на основании операции нелинейной адаптивной контурной фильтрации;

при этом набор параметров ограничения создается на основании битовой глубины и индекса ограничения, основанного на заданной таблице, причем в заданной таблице

значение одного параметра ограничения, соответствующего первому индексу ограничения для данной битовой глубины, получается на основании значения другого параметра ограничения, соответствующего первому индексу ограничения для другой битовой глубины.

5 14. Считываемый компьютером носитель запоминающего устройства длительного пользования, хранящий команды, вызывающие выполнение процессором:

определения, что к текущей видеообласти видео применяется операция нелинейной адаптивной контурной фильтрации, причем текущая видеообласть является блоком дерева кодирования;

10 получения по меньшей мере одного первого индекса фильтрации для текущей видеообласти и получения одного или более первых параметров фильтрации на основании указанного по меньшей мере одного первого индекса фильтрации;

15 получения одного или более первых параметров ограничения из набора параметров ограничения на основании указанного по меньшей мере одного первого индекса фильтрации,

выполнения операции ограничения, которая является частью операции нелинейной адаптивной контурной фильтрации, на основании указанного одного или более первых параметров ограничения, причем указанный один или более первых параметров ограничения используются для ограничения различий между контрольными значениями отсчетов текущего значения отсчета текущей видеообласти и текущим значением отсчета перед применением указанного одного или более первых параметров фильтрации; и

20 выполнения преобразования между текущей видеообластью и битовым потоком видео на основании операции нелинейной адаптивной контурной фильтрации;

при этом набор параметров ограничения создается на основании битовой глубины и индекса ограничения, основанного на заданной таблице, причем в заданной таблице значение одного параметра ограничения, соответствующего первому индексу ограничения для данной битовой глубины, получается на основании значения другого параметра ограничения, соответствующего первому индексу ограничения для другой битовой глубины.

30 15. Способ сохранения битового потока видео, содержащий этапы, на которых:

определяют, что к текущей видеообласти видео применяется операция нелинейной адаптивной контурной фильтрации, причем текущая видеообласть является блоком дерева кодирования;

35 получают по меньшей мере один первый индекс фильтрации для текущей видеообласти и получают один или более первых параметров фильтрации на основании указанного по меньшей мере одного первого индекса фильтрации;

получают один или более первых параметров ограничения из набора параметров ограничения на основании указанного по меньшей мере одного первого индекса фильтрации;

40 выполняют операцию ограничения, которая является частью операции нелинейной адаптивной контурной фильтрации, на основании указанного одного или более первых параметров ограничения, причем указанный один или более первых параметров ограничения используются для ограничения различий между контрольными значениями отсчетов текущего значения отсчета текущей видеообласти и текущим значением отсчета перед применением указанного одного или более первых параметров фильтрации;

45 вырабатывают битовый поток видео на основании операции ограничения; и

сохраняют битовый поток на считываемом компьютером носителе запоминающего устройства длительного пользования;

при этом набор параметров ограничения создается на основании битовой глубины и индекса ограничения, основанного на заданной таблице, причем в заданной таблице значение одного параметра ограничения, соответствующего первому индексу ограничения для данной битовой глубины, получается на основании значения другого параметра ограничения, соответствующего первому индексу ограничения для другой битовой глубины.

10

15

20

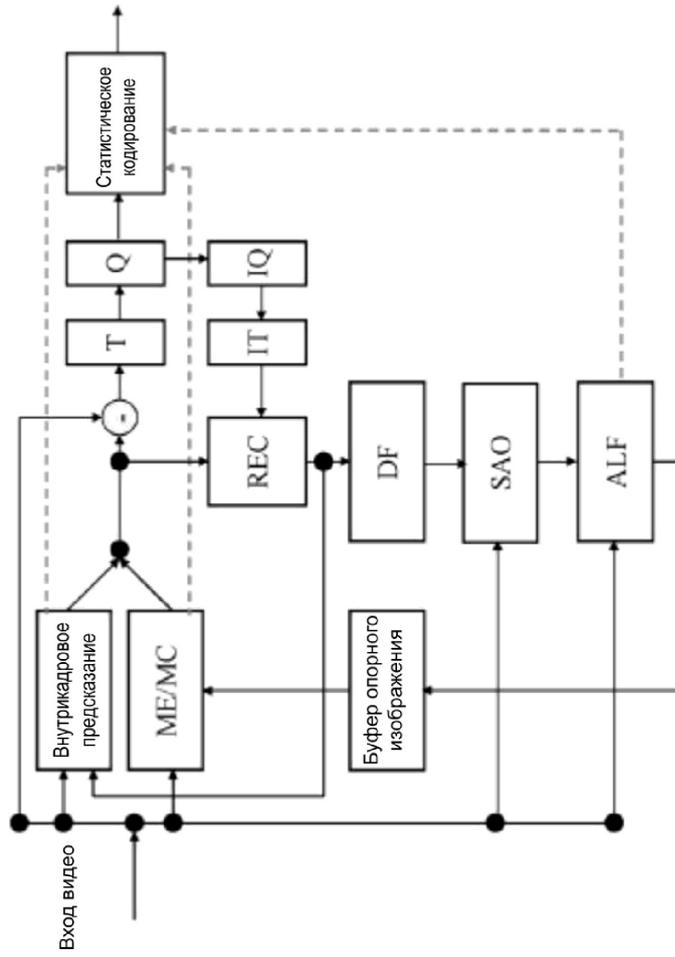
25

30

35

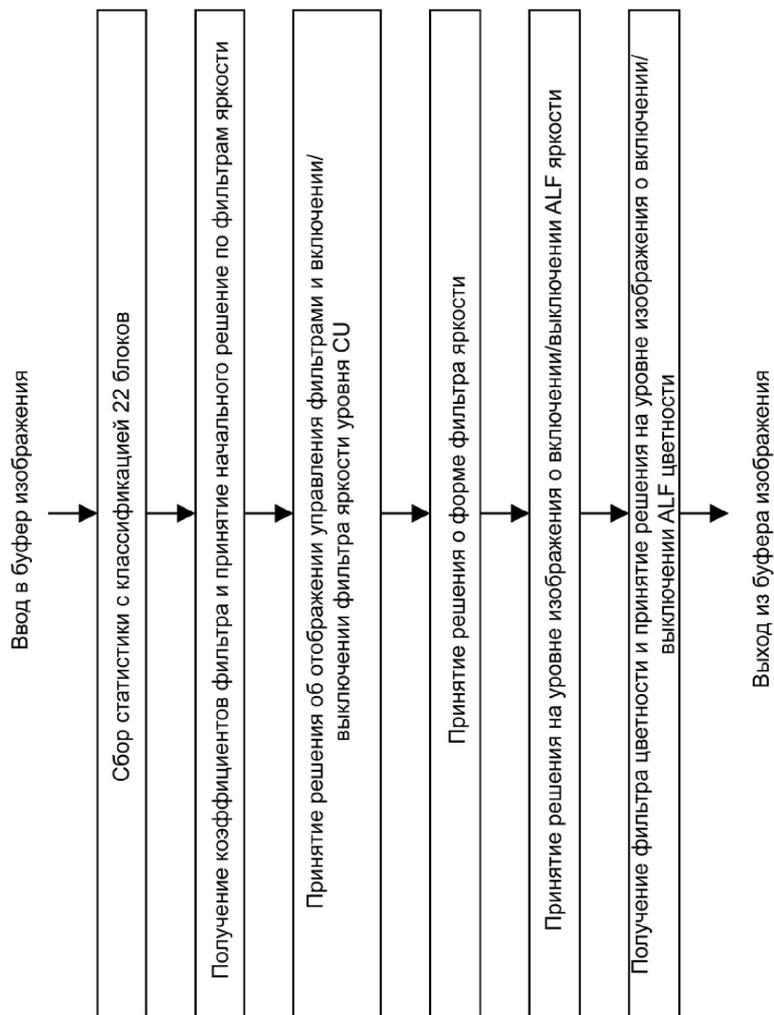
40

45



ФИГ. 1





ФИГ. 3

4/14

	Н		Н		Н		Н
Н		Н		Н		Н	
	Н	Н	Н	Н	Н		Н
Н		Н		Н		Н	
	Н	Н	Н	Н	Н		Н
Н		Н		Н		Н	
	Н		Н		Н		Н
Н		Н		Н		Н	

Фиг. 4В

	✓		✓		✓		✓
✓		✓		✓		✓	
	✓	✓	✓	✓	✓		✓
✓		✓		✓		✓	
	✓	✓	✓	✓	✓		✓
✓		✓		✓		✓	
	✓		✓		✓		✓
✓		✓		✓		✓	

Фиг. 4А

5/14

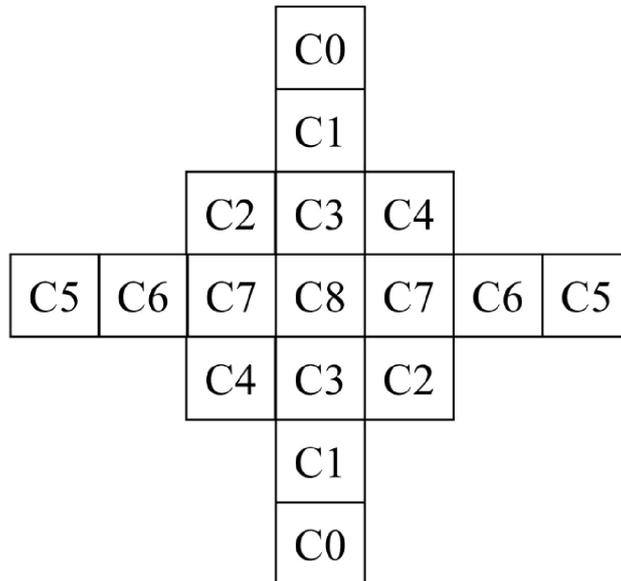
D2																			
	D2																		
D2		D2		D2		D2		D2		D2		D2		D2		D2		D2	
	D2																		
D2		D2		D2		D2		D2		D2		D2		D2		D2		D2	
	D2																		
D2		D2		D2		D2		D2		D2		D2		D2		D2		D2	
	D2																		
D2		D2		D2		D2		D2		D2		D2		D2		D2		D2	

ФИГ. 4D

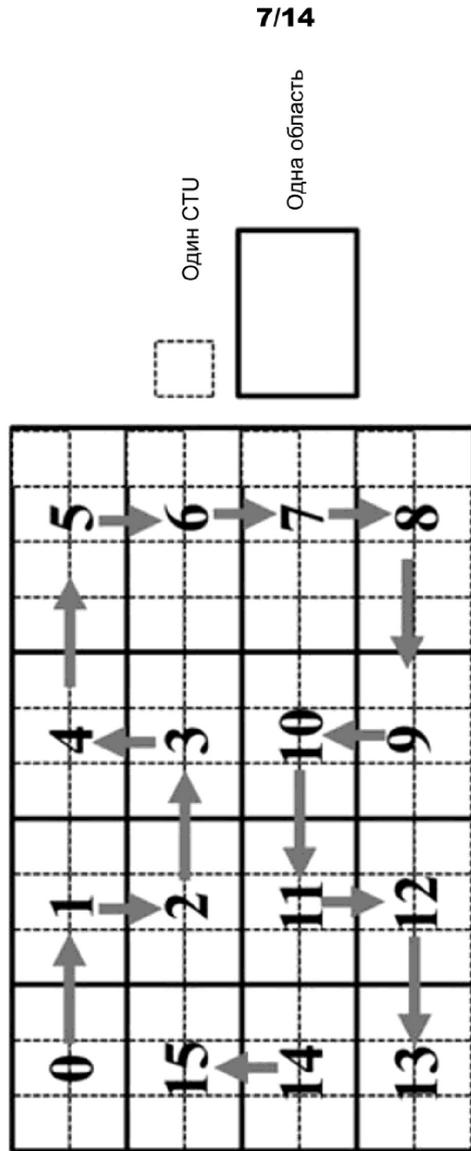
D1																			
	D1																		
D1		D1		D1		D1		D1		D1		D1		D1		D1		D1	
	D1																		
D1		D1		D1		D1		D1		D1		D1		D1		D1		D1	
	D1																		
D1		D1		D1		D1		D1		D1		D1		D1		D1		D1	
	D1																		
D1		D1		D1		D1		D1		D1		D1		D1		D1		D1	
	D1																		

ФИГ. 4C

6/14

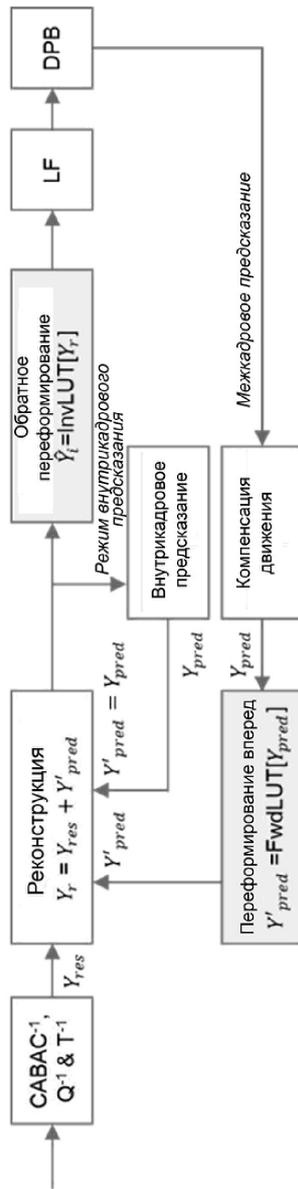


Фиг. 5

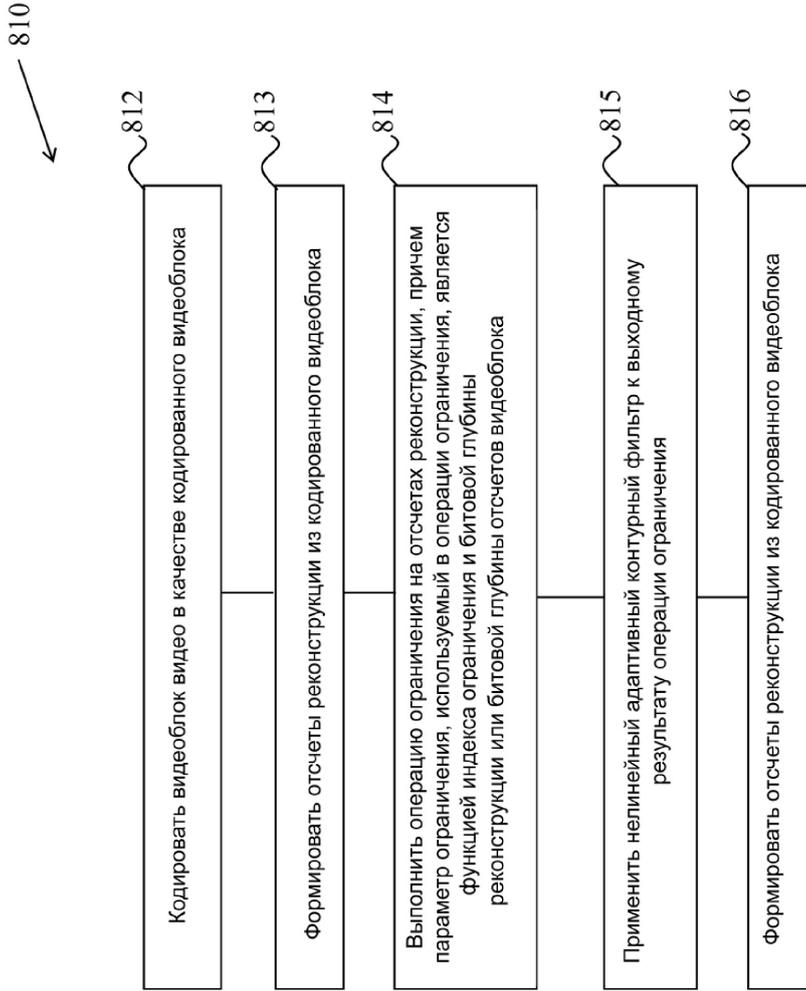


ФИГ. 6

8/14

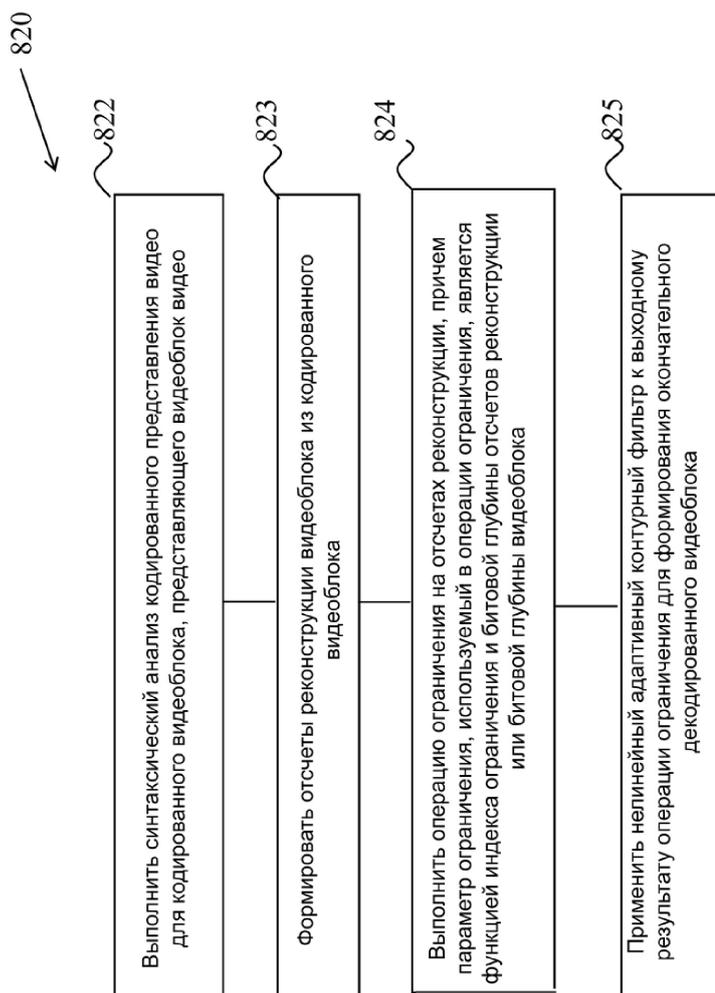


ФИГ. 7



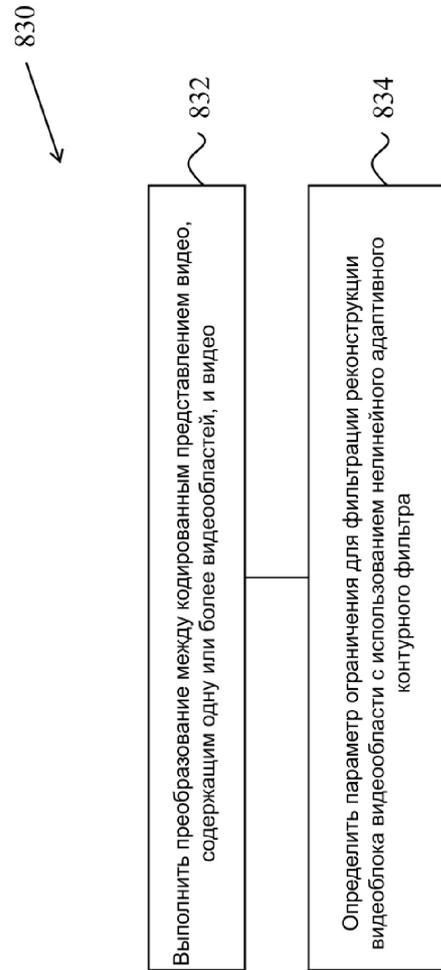
ФИГ. 8А

10/14



ФИГ. 8В

11/14



ФИГ. 8С

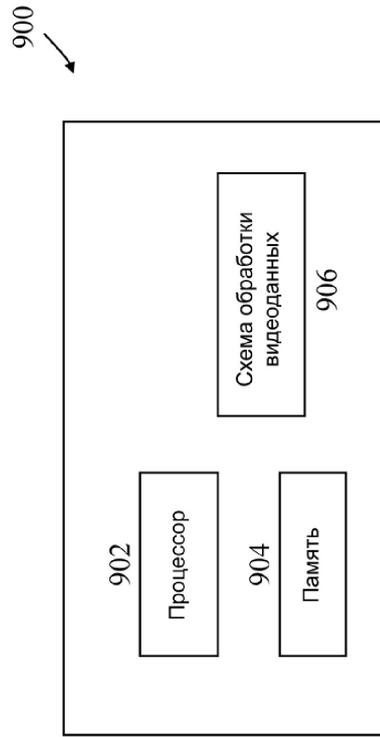
12/14

840  
↙

Выполнить преобразование между кодированным представлением видео, содержащим одну или более видеообластей и видео

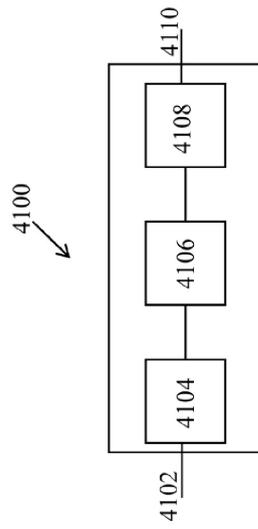
ФИГ. 9

13/14



Фиг. 10А

14/14



ФИГ. 10В