



(51) МПК

H04N 19/50 (2014.01)

H04N 19/513 (2014.01)

H04N 19/61 (2014.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2013140464/08, 31.01.2012

(24) Дата начала отсчета срока действия патента:
31.01.2012

Приоритет(ы):

(30) Конвенционный приоритет:

31.01.2011 KR 10-2011-0009636;

03.03.2011 KR 10-2011-0019166;

27.05.2011 KR 10-2011-0050853;

01.07.2011 KR 10-2011-0065707;

31.01.2012 KR 10-2012-0010096

(43) Дата публикации заявки: 10.03.2015 Бюл. № 7

(45) Опубликовано: 10.06.2016 Бюл. № 16

(56) Список документов, цитированных в отчете о поиске: US 2005/0013372 A1, 20.01.2005. US 2004/0057521 A1, 25.03.2004. US 2004/0105589 A1, 03.06.2004. US 2007/0237232 A1, 11.10.2007. RU 2106759 C1, 10.03.1998. RU 2335859 C2, 10.10.2008.

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: 02.09.2013

(86) Заявка РСТ:
KR 2012/000770 (31.01.2012)(87) Публикация заявки РСТ:
WO 2012/105807 (09.08.2012)

Адрес для переписки:

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

(72) Автор(ы):

ЛИМ Сунг Чанг (KR),
КИМ Хой Йонг (KR),
ДЗЕОНГ Се Йоон (KR),
ЧО Сук Хи (KR),
ДЗУН Донг Сан (KR),
КИМ Дзонг Хо (KR),
ЛИ Ха Хиун (KR),
ЛИ Дзин Хо (KR),
ЧОИ Дзин Соо (KR),
КИМ Дзин Воонг (KR)

(73) Патентообладатель(и):

ЭЛЕКТРОНИКС ЭНД
ТЕЛЕКОММЬЮНИКЕЙШНЗ РИСЕРЧ
ИНСТИТЮТ (KR)

(54) СПОСОБ И УСТРОЙСТВО ДЛЯ КОДИРОВАНИЯ/ДЕКОДИРОВАНИЯ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ВЕКТОРА ДВИЖЕНИЯ

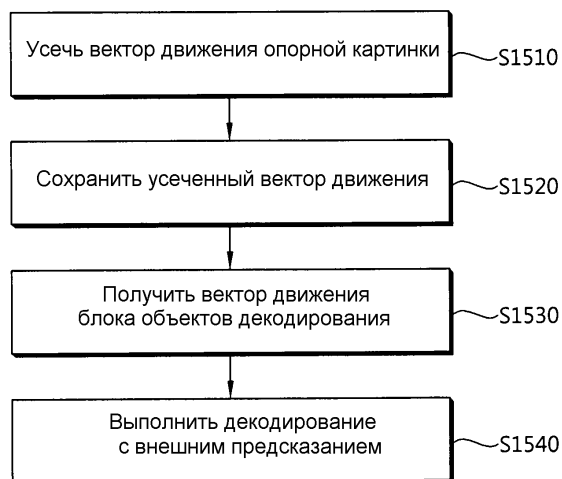
(57) Реферат:

Изобретение относится к технологиям кодирования/декодирования изображения с использованием вектора движения. Техническим результатом является уменьшение объема памяти, требуемого для хранения векторов движения, за счет усечения вектора движения. Предложено устройство декодирования изображения.

Устройство содержит буфер опорной картинки для хранения опорной картинки. Устройство содержит блок компенсации движения для генерирования блока предсказания с использованием опорной картинки и вектора движения опорной картинки. При этом вектор движения опорной картинки отсекается в

предварительно определенном фиксированном диапазоне. Устройство декодирования изображения декодирует битовый поток в режиме

внутреннего предсказания или в режиме внешнего предсказания для вывода реконструированного изображения. 4 з.п. ф-лы, 15 ил., 18 табл.



Фиг.15



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.

H04N 19/50 (2014.01)*H04N 19/513* (2014.01)*H04N 19/61* (2014.01)(12) **ABSTRACT OF INVENTION**(21)(22) Application: **2013140464/08**, 31.01.2012(24) Effective date for property rights:
31.01.2012

Priority:

(30) Convention priority:

31.01.2011 **KR** 10-2011-0009636;03.03.2011 **KR** 10-2011-0019166;27.05.2011 **KR** 10-2011-0050853;01.07.2011 **KR** 10-2011-0065707;31.01.2012 **KR** 10-2012-0010096

(43) Application published: 10.03.2015 Bull. № 7

(45) Date of publication: 10.06.2016 Bull. № 16

(85) Commencement of national phase: 02.09.2013

(86) PCT application:

KR 2012/000770 (31.01.2012)

(87) PCT publication:

WO 2012/105807 (09.08.2012)

Mail address:

129090, Moskva, ul. B. Spasskaja, 25, stroenie 3,
OOO "JUrIdicheskaja firma Gorodisskij i Partnery"

(72) Inventor(s):

LIM Sung CHang (KR),**KIM KHoj Jong (KR),****DZEONG Se Joon (KR),****CHO Suk KHi (KR),****DZUN Dong San (KR),****KIM Dzung KHo (KR),****LI KHa KHiun (KR),****LI Dzin KHo (KR),****CHOI Dzin Soo (KR),****KIM Dzin Voong (KR)**

(73) Proprietor(s):

**ELEKTRONIKS END TELEKOMMYUNIK
EJSHNZ RISERCH INSTITYUT (KR)**(54) **METHOD AND APPARATUS FOR ENCODING/DECODING IMAGES USING MOTION VECTOR**

(57) Abstract:

FIELD: information technology.

SUBSTANCE: invention relates to video encoding/decoding image using motion vector. Image decoding device is disclosed. Device has buffer support picture for storage of reference pictures. Device comprises unit for motion compensation for generating prediction unit using reference picture and motion vector of reference picture. At that, motion vector of reference picture is truncated in predetermined fixed range. Image decoding device decodes bit stream in internal prediction mode or in mode of external prediction for outputting reconstructed image.

EFFECT: reduced amount of memory required for storing motion vectors, due to truncation motion vector.

5 cl, 15 dwg, 18 tbl



Фиг.15

ОБЛАСТЬ ТЕХНИКИ

Настоящее изобретение относится к обработке изображения, и более конкретно, к способу кодирования/декодирования видео с использованием усеченного вектора движения и устройству для этого.

УРОВЕНЬ ТЕХНИКИ

В последнее время в соответствии с распространением систем вещания, поддерживающих разрешение высокой четкости (HD) в стране и во всем мире, многие пользователи привыкли к изображению с высоким разрешением и четкостью, так что многие организации провели много попыток для разработки видеоустройств следующего поколения. К тому же, так как интерес к HDTV и сверхвысокому разрешению (UHD), имеющему разрешение в четыре раза выше, чем разрешение HDTV, увеличился, потребовалась технология сжатия для видео с более высоким разрешением и более высокой четкостью.

Для сжатия изображения могут быть использованы технология внешнего предсказания для предсказания значений пикселей, включенных в текущую картинку из картинки до и/или после текущей картинки, технология внутреннего предсказания для предсказания значений пикселей с использованием информации пикселей в данной картинке, технология энтропийного кодирования для присвоения сокращенного кода символу, имеющему высокую частоту появления, длинного кода символу, имеющему низкую частоту появления, или подобные.

РАСКРЫТИЕ

ТЕХНИЧЕСКАЯ ПРОБЛЕМА

Настоящее изобретение предоставляет способ кодирования/декодирования видео с использованием усеченного вектора движения и устройство для него.

Настоящее изобретение также предоставляет способ для усечения вектора движения опорной картинки.

Настоящее изобретение также предоставляет способ для передачи информации о векторе движения.

ТЕХНИЧЕСКОЕ РЕШЕНИЕ

1. Вариант осуществления настоящего изобретения относится к способу кодирования изображения. Способ включает в себя усечение вектора движения опорной картинки в предварительно определенном динамическом диапазоне, чтобы сгенерировать усеченный вектор движения, сохранение усеченного вектора движения в буфере и кодирование вектора движения древовидного блока кодирования с использованием вектора движения, сохраненного в буфере.

2. В п.1, динамический диапазон может быть задан уровнем видеокодека.

3. В п.1, динамический диапазон может быть определен посредством предварительно определенной глубины в битах, и глубина в битах может быть задана уровнем видеокодека.

4. В п.1, X-составляющая и Y-составляющая вектора движения опорной картинки могут быть усечены в разных динамических диапазонах.

5. Еще один вариант осуществления настоящего изобретения относится к способу декодирования изображения. Способ включает в себя усечение вектора движения опорной картинки в предварительно определенном динамическом диапазоне, чтобы сгенерировать усеченный вектор движения, сохранение усеченного вектора движения в буфере, получение вектора движения древовидного блока кодирования с использованием вектора движения, сохраненного в буфере и выполнение процесса декодирования с внешним предсказанием с использованием вектора движения

древовидного блока кодирования.

6. В п.5, динамический диапазон может быть задан уровнем видеокодека.

7. В п.5, динамический диапазон может быть определен посредством предварительно определенной глубины в битах, и глубина в битах может быть задана уровнем видеокодека.

8. В п.5, динамический диапазон может быть определен посредством предварительно определенной глубины в битах, и глубина в битах может быть получена посредством набора параметров последовательности, передаваемого из устройства кодирования изображения.

9. В п.8, набор параметров последовательности может включать в себя флаг, указывающий является ли вектор движения опорной картинки усеченным, и параметр для получения глубины в битах.

10. В п.9, способ декодирования изображения может включать в себя сжатие вектора движения опорной картинки, в котором набор параметров последовательности включает в себя флаг, указывающий является ли вектор движения опорной картинки сжатым, и параметр для получения коэффициента сжатия вектора движения опорной картинки.

11. В п.5, способ декодирования изображения может включать в себя ограничение разрешения представления вектора движения опорной картинки.

12. В п.5, усеченный вектор движения может быть сохранен согласно приоритету.

13. В п.5, усеченный вектор движения может быть вектором движения блока, кодированного в режиме внешнего предсказания.

14. В п.5, способ декодирования изображения может включать в себя выполнение масштабирования вектора движения опорной картинки.

15. В п.5, X-составляющая и Y-составляющая вектора движения опорной картинки могут быть усечены в разных динамических диапазонах.

16. В п.15, динамический диапазон X-составляющей и динамический диапазон Y-составляющей могут быть заданы уровнем видеокодека.

17. Еще один вариант осуществления настоящего изобретения относится к устройству декодирования изображения. Устройство включает в себя буфер опорной картинки, хранящий опорную картинку, и компенсатор движения, генерирующий предсказанный блок с использованием опорной картинки и вектора движения опорной картинки, при этом вектор движения опорной картинки усекается в предварительно определенном динамическом диапазоне.

18. В п.17, динамический диапазон может быть задан уровнем видеокодека.

19. В п.17, динамический диапазон может быть определен посредством предварительно определенной глубины в битах, и глубина в битах может быть задана уровнем видеокодека.

20. В п.17, динамический диапазон может быть определен посредством предварительно определенной глубины в битах, и глубина в битах может быть получена посредством набора параметров последовательности, передаваемого из устройства кодирования изображения.

ПОЛЕЗНЫЕ РЕЗУЛЬТАТЫ

Согласно примерному варианту осуществления настоящего изобретения видео может быть кодировано с использованием усеченного вектора движения.

Согласно примерному варианту осуществления настоящего изобретения размер памяти, требуемой для хранения векторов движения, может быть уменьшен.

Согласно примерному варианту осуществления настоящего изобретения полоса пропускания для доступа к памяти, требуемая для извлечения данных из памяти, может

быть уменьшена.

ОПИСАНИЕ ЧЕРТЕЖЕЙ

Фиг.1 является блок-схемой, показывающей пример структуры кодера согласно
примерному варианту осуществления настоящего изобретения.

5 Фиг.2 является блок-схемой, показывающей пример структуры декодера согласно
примерному варианту осуществления настоящего изобретения.

Фиг.3 показывает примеры картинки объектов кодирования/декодирования и опорной
картинки.

Фиг.4 показывает пример ограничения динамического диапазона вектора движения.

10 Фиг.5-8 являются схемами последовательностей операций, показывающими способ
сохранения вектора движения опорной картинки.

Фиг.9 показывает пример квантования вектора движения.

Фиг.10-13 показывают примеры извлечения информации движения из опорной
картинки.

15 Фиг.14 является схемой последовательности операций, показывающей способ
кодирования изображения согласно примерному варианту осуществления настоящего
изобретения.

Фиг.15 является схемой последовательности операций, показывающей способ
20 декодирования изображения согласно примерному варианту осуществления настоящего
изобретения.

ПОДРОБНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

В дальнейшем, примерные варианты осуществления настоящего изобретения будут
описаны подробно со ссылкой на прилагаемые чертежи. Однако, при описании
примерных вариантов осуществления настоящего изобретения хорошо известные
25 функции или конструкции не будут описаны подробно, так как они могут сделать
понимание настоящего изобретения затруднительным.

Следует понимать, что когда какой-либо элемент называется являющимся
"соединенным с" или "связанным с" другим элементом, он может быть соединен
напрямую или связан напрямую с другим элементом или быть соединен или связан с
30 другим элементом, так что другой элемент находится между ними. К тому же, в
настоящем описании в случае, когда описывается как "включающий в себя" конкретный
компонент, следует понимать, что помимо соответствующего компонента, не исключены
другие компоненты, и они могут быть включены в примерные варианты осуществления
или технический объем настоящего изобретения.

35 Термины, используемые в данном описании, "первый", "второй" и т.д., могут быть
использованы для описания различных компонентов, но эти компоненты не должны
толковаться как ограниченные данными терминами. То есть, данные термины
используются для отличия одного компонента от другого компонента. Например,
"первый" компонент может быть назван "вторым" компонентом, или наоборот, без
40 отступления от объема настоящего изобретения.

К тому же, компоненты, описанные в примерных вариантах осуществления
настоящего изобретения, показаны раздельно только для того, чтобы указать, что они
выполняют разные характерные функции. Вследствие этого, компоненты, которые
показаны раздельно, не означают, что каждый из компонентов не может быть
45 реализован как одно аппаратное средство или программное средство. То есть, каждый
из компонентов разделен для удобства разъяснения, множество компонентов могут
быть объединены друг с другом, чтобы тем самым работать как один компонент, или
один компонент может быть разделен на множество компонентов, чтобы тем самым

работать как множество компонентов, которые включены в объем настоящего изобретения, пока он отступает от существенных характеристик настоящего изобретения.

К тому же, некоторые из компонентов могут не быть необходимыми компонентами, выполняющими ключевые функции настоящего изобретения, но могут быть выборочными компонентами, улучшающими только его эффективность. Настоящее изобретение может также быть реализовано только посредством структуры, включающей в себя необходимые компоненты, за исключением выборочных компонентов, и структура, включающая в себя только необходимые компоненты, также включена в объем настоящего изобретения.

Фиг.1 является блок-схемой, показывающей пример структуры кодера согласно примерному варианту осуществления настоящего изобретения.

Ссылаясь на фиг.1, кодер 100 включает в себя модуль 111 предсказания движения, компенсатор 112 движения, модуль 120 внутреннего предсказания, переключатель 115, модуль 125 вычитания, преобразователь 130, модуль 140 квантования, энтропийный кодер 150, модуль 160 деквантования, обратный преобразователь 170, сумматор 175, фильтрующий модуль 180 и буфер 190 опорной картинки.

Кодер 100 кодирует входные изображения в режиме внутреннего предсказания или режиме внешнего предсказания для вывода кодером битового потока. Внутреннее предсказание означает предсказание внутри картинки, и внешнее предсказание означает предсказание между картинками. Кодер 100 переключается между режимом внутреннего предсказания и режимом внешнего предсказания посредством переключения переключателя 115. Кодер 100 генерирует предсказанный блок для входного блока входного изображения и затем кодирует остаток между входным блоком и предсказанным блоком.

В случае режима внутреннего предсказания модуль 120 внутреннего предсказания выполняет пространственное предсказание с использованием значений пикселей соседних блоков, которые уже кодированы, чтобы сгенерировать предсказанные блоки.

В случае режима внешнего предсказания модуль 111 предсказания движения осуществляет поиск опорного блока, оптимально совпадающего с входным блоком, в опорной картинке, хранящейся в буфере 190 опорной картинки, во время процесса предсказания движения, чтобы получить вектор движения. Компенсатор 112 движения выполняет компенсацию движения с использованием вектора движения, чтобы сгенерировать предсказанный блок. Здесь, вектор движения может быть двухмерным вектором, используемым для внешнего предсказания, и представлять смещение между текущим древовидным блоком кодирования и опорным блоком.

Модуль 125 вычитания генерирует остаточный блок на основе остатка между входным блоком и предсказанным блоком, и преобразователь 130 преобразует остаточный блок для вывода коэффициента преобразования. Модуль 140 квантования квантует коэффициент преобразования для вывода квантованного коэффициента.

Энтропийный кодер 150 выполняет энтропийное кодирование на основе информации, полученной во время процесса кодирования/квантования для вывода битового потока. Энтропийное кодирование представляет часто генерируемые символы как небольшое число битов, тем самым уменьшая размер битового потока для символа кодирования. Вследствие этого, можно ожидать улучшения эффективности сжатия видео посредством энтропийного кодирования. Энтропийный кодер 150 для энтропийного кодирования может использовать способ кодирования, такой как экспоненциальный код Голомба, контекстно-адаптированное кодирование с переменной длиной кодового слова (CAVLC),

контекстно-адаптированное двоичное арифметическое кодирование (CABAC), или подобный.

Кодированная картинка должна быть снова декодирована и сохранена, чтобы быть использованной в качестве опорной картинки для выполнения кодирования с внешним предсказанием. Вследствие этого, модуль 160 деквантования деквантует квантованный коэффициент, и обратный преобразователь 170 осуществляет обратное преобразование деквантованного коэффициента для вывода реконструированного остаточного блока. Сумматор 175 добавляет (суммирует с) реконструированный остаточный блок к предсказанному блоку, чтобы сгенерировать реконструированный блок.

Фильтрующий модуль 180 также называется адаптивным контурным фильтром и применяет к реконструированному блоку по меньшей мере одно из фильтрации блочности, адаптивной компенсации смещения выборок (SAO), адаптивной контурной фильтрации (ALF). Фильтрация блочности означает, что удаляется блочность изображения, возникающая на границе между блоками, и компенсация SAO означает, что соответствующее смещение добавляется к значению пикселя, для того, чтобы компенсировать ошибку кодирования. К тому же, ALF означает, что фильтрация выполняется на основе сравнения значения между реконструированным изображением и первоначальным изображением.

При этом, буфер 190 опорной картинки хранит реконструированный блок, пропущенный через фильтрующий модуль 180.

Фиг.2 является блок-схемой, показывающей пример структуры декодера согласно примерному варианту осуществления настоящего изобретения.

Ссылаясь на Фиг.2, декодер включает в себя энтропийный декодер 210, модуль 220 деквантования, обратный преобразователь 230, модуль 240 внутреннего предсказания, компенсатор 250 движения, сумматор 255, фильтрующий модуль 260 и буфер 270 опорной картинки.

Декодер 200 декодирует битовый поток в режиме внутреннего предсказания или режиме внешнего предсказания для вывода реконструированного изображения. Декодер 200 переключается между режимом внутреннего предсказания и режимом внешнего предсказания посредством переключения переключателя. Декодер 200 получает остаточный блок из битового потока, чтобы сгенерировать предсказанный блок, и затем добавляет остаточный блок и предсказанный блок друг к другу, чтобы сгенерировать реконструированный блок.

Энтропийный декодер 210 выполняет энтропийное декодирование на основе распределения вероятностей. Процесс энтропийного декодирования является процессом, противоположным вышеупомянутому процессу энтропийного кодирования. То есть энтропийный декодер 210 генерирует символ, включающий в себя квантованный коэффициент из битового потока, в котором часто генерируемый символ представлен как небольшое число битов.

Модуль 220 деквантования деквантует квантованный коэффициент, и обратный преобразователь 230 осуществляет обратное преобразование деквантованного коэффициента, чтобы сгенерировать остаточный блок.

В случае режима внутреннего предсказания модуль 240 внутреннего предсказания выполняет пространственное предсказание с использованием значений пикселей соседних блоков, которые уже кодированы, чтобы сгенерировать предсказанные блоки.

В случае режима внешнего предсказания компенсатор 250 движения выполняет компенсацию движения с использованием вектора движения и опорной картинки, хранящейся в буфере 270 опорной картинки, чтобы сгенерировать предсказанный блок.

Сумматор 255 добавляет (суммирует с) предсказанный блок к остаточному блоку, и фильтрующий модуль 260 применяет по меньшей мере одно из фильтрации блочности, компенсации SAO, ALF, к блоку, пропущенному через сумматор, для вывода реконструированного изображения.

5 Реконструированное изображение может быть сохранено в буфере 270 опорной картинки, чтобы тем самым быть использованным для компенсации движения.

В дальнейшем, блок означает единицу кодирования/декодирования. В процессе кодирования/декодирования изображение разделяется на предварительно определенный размер и затем кодируется/декодируется. Вследствие этого, блок может также быть
10 назван единицей кодирования (CU), единицей предсказания (PU), единицей преобразования (TU) или подобным, и одиночный блок может также быть разделен на подблоки, имеющие меньший размер.

Здесь, единица предсказания означает базовую единицу, в которой выполняется предсказание и/или компенсация движения. Единица предсказания может быть разделена
15 на множество сегментов, и каждый из сегментов может быть также назван сегментом единицы предсказания. Когда единица предсказания разделена на множество сегментов, каждый из сегментов единицы предсказания может стать базовой единицей, в которой выполняются предсказание и/или компенсация движения. В дальнейшем, в примерном варианте осуществления настоящего изобретения, единица предсказания может также
20 означать сегменты единицы предсказания.

При этом, в кодировании видео высокой эффективности (HEVC) используется способ предсказания вектора движения, основанный на усовершенствованном предсказании вектора движения (AMVP).

В способе предсказания вектора движения, основанном на усовершенствованном
25 предсказании вектора движения, может быть использован вектор движения (MV) блока, существующего в положении, которое является таким же или соответствует положению древовидного блока кодирования, в опорной картинке, так же как и векторы движения реконструированных блоков, расположенных вокруг древовидного блока кодирования. Здесь, блок, существующий в положении, которое является таким же или
30 пространственно соответствует положению древовидного блока кодирования, в опорной картинке называется совмещенным блоком, и вектор движения совмещенного блока называется совмещенным вектором движения или временным вектором движения. Однако, совмещенный блок может быть блоком, существующим в положении, аналогичном (то есть, соответствующим) положению древовидного блока кодирования,
35 в опорной картинке, так же как и блоком, существующим в том же положении, что и положение древовидного блока кодирования.

В способе слияния информации движения информация движения оценивается исходя из совмещенного блока, так же как и реконструированных блоков, расположенных вокруг древовидного блока кодирования, чтобы тем самым быть использованной в
40 качестве информации движения древовидного блока кодирования. Здесь, информация движения включает в себя по меньшей мере одно из информации режима внешнего предсказания, указывающей индекс опорной картинки, вектор движения, однонаправленный, двунаправленный или подобное, требуемой в момент внешнего предсказания, списка опорных картинок и информации режима предсказания о том,
45 выполняется ли кодирование в режиме внутреннего предсказания или в режиме внешнего предсказания.

Предсказанный вектор движения в древовидном блоке кодирования может быть вектором движения совмещенного блока, который является блоком, временно смежным

с древовидным блоком кодирования, так же как векторы движения соседних блоков, пространственно смежных с древовидным блоком кодирования.

Фиг.3 показывает примеры картинки объектов кодирования/декодирования и опорной картинки.

5 На фиг.3, блок X указывает древовидный блок кодирования в картинке 310 объектов кодирования/декодирования, и блок А, блок В, блок С, блок D и блок Е указывают реконструированные блоки, расположенные вокруг древовидного блока кодирования. К тому же, блок Т в опорной картинке 320 указывает совмещенный блок, существующий в положении, соответствующем положению древовидного блока кодирования.

10 Какой вектор движения в древовидном блоке кодирования используется в качестве предсказанного вектора движения может быть распознано посредством индекса блока предсказания вектора движения.

Таблица 1

15	prediction_unit(x0, y0, log2PUWidth, log2PUHeight) {	Дескриптор
	if(skip_flag[x0][y0]) {	
	if(NumMVPCand(L0) > 1)	
	mvp_idx_10[x0][y0]	ue(v) : ae(v)
	if(NumMVPCand(L1) > 1)	
20	mvp_idx_11[x0][y0]	ue(v) : ae(v)
	} else if(PredMode == MODE_INTRA) {	
	...	
	} else { /* MODE_MERGE, MODE_INTER */	
	if(merge_flag[x0][y0] && NumMergeCandidates > 1) {	
	...	
25	} else {	
	if(inter_pred_idc[x0][y0] != Pred_L1) {	
	if(NumMVPCand(L0) > 1)	
	mvp_idx_10	ue(v) : ae(v)
	}	
	if(inter_pred_idc[x0][y0] != Pred_L0) {	
30	if(NumMVPCand(L1) > 1)	
	mvp_idx_11	ue(v) : ae(v)
	}	
	}	
	}	
	}	

35 Как показано в таблице 1, индексы модуля предсказания вектора движения mvp_idx_10 и mvp_idx_11 для каждого списка опорных картинок передаются декодеру, и декодер использует такой же вектор движения как вектор движения, предсказанный кодером, в качестве предсказанного вектора движения.

40 В случае, в котором древовидный блок кодирования кодируется/декодируется с использованием векторов движения соседних блоков, пространственно смежных с древовидным блоком кодирования, вектор движения может храниться только с помощью памяти, имеющей относительно небольшой размер. Однако, в случае, в котором используется временный вектор движения, так как все векторы движения опорной картинки должны храниться в памяти, требуется память, имеющая

45 относительно большой размер, и размер полосы пропускания для доступа к памяти, требуемой для извлечения данных из памяти, также увеличивается. Вследствие этого, нужно более эффективно хранить временный вектор движения в прикладном окружении, в котором пространство памяти портативного терминала, или подобного, является

недостаточным или потребление электроэнергии минимизировано.

При этом в качестве технологии сохранения вектора движения в памяти, существует способ уменьшения пространственного разрешения вектора движения. В этом способе вектор движения сжимается с любым коэффициентом и затем сохраняется в памяти.

Например, вектор движения, сохраненный в единице блока 4×4, сохранен в 4×4 или большей единице блока для уменьшения числа сохраненных векторов движения. Здесь, для того, чтобы регулировать размер блока сохраненного вектора движения передается информация о коэффициенте сжатия. Информация передается посредством набора параметров последовательности (SPS), как показано в таблице 2.

Таблица 2

seq_parameter_set_rbsp() {	C	Дескриптор
.....		
motion_vector_buffer_comp_flag	0	u(1)
if (motion_vector_buffer_comp_flag)		
motion_vector_buffer_comp_ratio_log2	0	u(8)
rbps_trailing_bits()	0	
}		

Ссылаясь на таблицу 2, в случае, в котором motion_vector_buffer_comp_flag равен 1, выполняется процесс сжатия буфера вектора движения.

motion_vector_buffer_comp_ratio_log2 указывает коэффициент сжатия процесса сжатия буфера вектора движения. В случае, в котором motion_vector_buffer_comp_ratio_log2 не существует, motion_vector_buffer_comp_ratio_log2 расценивается как 0, и коэффициент сжатия буфера вектора движения представляется уравнением 1.

[Уравнение 1]

$MVBufferCompRatio = 1 < \text{motion_vector_buffer_comp_ratio_log2}$

Например, в случае, в котором все блоки 4×4 картинок 1920×1080 имеют разные вектора движения и используют два списка опорных картинок, причем каждый использует две опорные картинки, для хранения временного вектора движения требуется общее пространство памяти 3,21 Мбайта, как описано ниже.

1. Глубина в битах 26 бит на один вектор движения.

(1) Динамический диапазон X-составляющей вектора движения: от -252 до +7676 (глубина в битах: 13 битов)

(2) Динамический диапазон Y-составляющей вектора движения: от -252 до +4316 (глубина в битах: 13 битов)

(3) (Динамические диапазоны каждой составляющей вектора движения были вычислены на основе первой единицы предсказания в соответствующей картинке.)

2. В случае, в котором все из единиц блока 4×4 имеют разные векторы движения: 480×270 = 129600 блоков

3. Использование двух векторов движения на каждый блок.

4. Число списков опорных картинок: 2

5. Использование двух опорных картинок на каждый список опорных картинок
=> 26 битов × 129600 блоков × два вектора движения × два списка опорных картинок × две опорные картинки = 26956800 битов = 3,21 Мбайта

Согласно способу уменьшения пространственного разрешения вектор движения, как описано выше, возможно уменьшить размер требуемого пространства памяти и полосы пропускания для доступа к памяти с использованием пространственной корреляции вектора движения. Однако, способ уменьшения пространственного разрешения вектора движения не ограничивает динамический диапазон вектора движения.

Когда размер пространства памяти уменьшается на 1/4, размер пространства памяти, требуемого в вышеупомянутом примере, уменьшается до почти 0,8 Мбайта. Здесь, когда только шесть битов глубины в битах, требуемой для хранения вектора движения, используются для каждой составляющей вектора движения посредством

5 дополнительного ограничения динамического диапазона вектора движения, размер требуемого пространства памяти может быть дополнительно уменьшен до 0,37 Мбайта.

Вследствие этого в примерном варианте осуществления настоящего изобретения, динамический диапазон вектора движения ограничивается, для того, чтобы уменьшить размер пространства памяти, требуемого для хранения вектора движения, и полосы пропускания для доступа к памяти, требуемой для извлечения данных из памяти. Вектор движения опорной картинки, динамический диапазон которой ограничен, может быть использован в качестве временного вектора движения в древовидном блоке кодирования.

В дальнейшем, динамический диапазон означает диапазон между минимальным значением и максимальным значением отрицательной составляющей или положительной составляющей вектора движения на основе 0, и глубина в битах, которая указывает размер пространства, требуемого для хранения вектора движения, означает ширину в битах. К тому же, пока не описано конкретно, вектор движения означает вектор движения опорной картинки, то есть, временный вектор движения.

В случае, в котором каждая составляющая вектора движения выходит за динамический диапазон, он представляется минимальным значением или максимальным значением соответствующего динамического диапазона. Например, в случае, в котором Х-составляющая вектора движения составляет 312 и максимальное значение динамического диапазона каждой составляющей вектора движения составляет 256, Х-составляющая вектора движения ограничивается до 256.

25 Аналогично, в случае, в котором глубина в битах каждой составляющей вектора движения составляет 16 битов, и вектор движения составляет (-36, 24), когда глубина в битах каждой составляющей вектора движения ограничена до 6 битов, каждая составляющая вектора движения имеет динамический диапазон от -32 до +31, так что вектор движения представляется посредством (-32, 24), что находится в его динамическом диапазоне.

К тому же, в случае, в котором глубина в битах каждой составляющей вектора движения составляет 16 битов, и вектор движения составляет (-49, 142), когда глубина в битах каждой составляющей вектора движения ограничена до 9 битов, каждая составляющая вектора движения имеет динамический диапазон от -256 до +255, так что вектор движения представляется посредством (-49, 142) без изменения

Фиг.4 показывает пример ограничения динамического диапазона вектора движения.

Ссылаясь на фиг.4, когда динамический диапазон вектора движения, имеющего динамический диапазон от -4096 до +4095, ограничивается от -128 до +127, глубина в битах может быть уменьшена с 13 битов до 8 битов.

40 Каждая составляющая временного вектора движения усекается как представлено уравнениями 2 и 3, чтобы быть сохраненной при глубине в битах из N бита(ов). Где N является положительным целым числом.

[Уравнение 2]

$$\text{clippedMV_X} = \min(1 \ll (N-1) - 1, \max(-1 \ll (N-1), \text{MV_X}))$$

45 [Уравнение 3]

$$\text{clippedMV_Y} = \min(1 \ll (N-1) - 1, \max(-1 \ll (N-1), \text{MV_Y}))$$

где MV_X указывает Х-составляющую вектора движения, MV_Y указывает Y-составляющую вектора движения, min(a,b) означает операцию вывода меньшего значения

в а и b, и $\max(a,b)$ означает операцию вывода большего значения в а и b. Каждый из clippedMV_X и clippedMV_Y указывает X- и Y-составляющие усеченного временного вектора движения, и хранится в памяти, чтобы тем самым быть использованным в качестве временного вектора движения древовидного блока кодирования.

Например, как показано в таблице 3, в случае, в котором размер пространства памяти составляет 48 байт, и каждая составляющая вектора движения использует глубину в битах из 16 битов, могут быть сохранены всего двенадцать векторов движения.

Таблица 3							
MV1-X	MV1-Y	MV2-X	MV2-Y	MV3-X	MV3-Y	MV4-X	MV4-Y
MV5-X	MV5-Y	MV6-X	MV6-Y	MV7-X	MV7-Y	MV8-X	MV8-Y
MV9-X	MV9-Y	MV10-X	MV10-Y	MV11-X	MV11-Y	MV12-X	MV12-Y

Однако, когда каждая составляющая вектора движения использует только глубину в битах из 8 битов, могут быть сохранены всего двадцать четыре вектора движения, как показано в таблице 4.

Таблица 4							
MV1-X	MV1-Y	MV2-X	MV2-Y	MV3-X	MV3-Y	MV4-X	MV4-Y
MV5-X	MV5-Y	MV6-X	MV6-Y	MV7-X	MV7-Y	MV8-X	MV8-Y
MV9-X	MV9-Y	MV10-X	MV10-Y	MV11-X	MV11-Y	MV12-X	MV12-Y
MV13-X	MV13-Y	MV14-X	MV14-Y	MV15-X	MV15-Y	MV16-X	MV16-Y
MV17-X	MV17-Y	MV18-X	MV18-Y	MV19-X	MV19-Y	MV20-X	MV20-Y
MV21-X	MV21-Y	MV22-X	MV22-Y	MV23-X	MV23-Y	MV24-X	MV24-Y

Вследствие этого, согласно примерному варианту осуществления настоящего изобретения, когда изображение, реконструированное в кодере и/или декодере, подвергается обработке контурным фильтром, таким как фильтр блочности, адаптивный контурный фильтр, или подобным, и затем сохраняется в буфере декодированной картинки (DPB), динамический диапазон вектора движения ограничивается, так чтобы вектор движения опорной картинки был сохранен. Буфер декодированной картинки означает буфер опорной картинки согласно фиг.1 или Фиг.2.

I. Процесс усечения вектора движения

Процесс усечения каждой составляющей вектора движения задействуется в случае, в котором slice_type не равен 1. Процесс усечения вектора движения выполняется в древовидном блоке или наибольшей единице кодирования после завершения процесса фильтрации. Входными данными в процессе усечения вектора движения являются местоположение (xP, yP), задающее верхнюю левую выборку единицы предсказания относительно верхней левой выборки текущей картинки, и матрицы векторов движения $MvL0$ и $MvL1$. Выходными данными в данном процессе являются матрицы усеченных векторов движения $CMvL0$ и $CMvL1$.

По отношению к матрицам $MvL0$, $MvL1$, $CMvL0$ и $CMvL1$ выполняются операции уравнений с 4 по 7.

[Уравнение 4]

$$mvLX = MvLX[xP, yP]$$

[Уравнение 5]

$$cmvLX[0] = \text{Clip}(-1 \ll (\text{TMVBitWidth} - 1), 1 \ll (\text{TMVBitWidth} - 1) - 1, mvLX[0])$$

[Уравнение 6]

$$cmvLX[1] = \text{Clip3}(-1 \ll (\text{TMVBitWidth} - 1), 1 \ll (\text{TMVBitWidth} - 1) - 1, mvLX[1])$$

[Уравнение 7]

$$CMvLX[xP, yP] = cmvLX$$

где TMVBitWidth указывает глубину в битах вектора движения, $\text{Clip3}(a, b, c)$ означает

функцию усечения s , так чтобы существовать в диапазоне между a и b .

II. Процесс сохранения вектора движения

Фиг.5-8 являются схемами последовательностей операций, показывающими способ сохранения вектора движения опорной картинки.

5 Ссылаясь на фиг.5, вектор движения опорной картинки может быть сохранен с использованием как буфера изображения, хранящего реконструированное изображение, так и буфера вектора движения, хранящего вектор движения. Здесь, реконструированное изображение подвергается обработке контурным фильтром (S510), и вектор движения подвергается процессу ограничения динамического диапазона (S520) и затем сохраняется
10 (S540).

К тому же, ссылаясь на фиг.6, используются и буфер изображения, и буфер вектора движения, вектор движения подвергается процессу ограничения динамического диапазона (S620) и процессу уменьшения пространственного разрешения (S630) и затем сохраняется (S640).

15 К тому же, ссылаясь на фиг.7, реконструированное изображение подвергается обработке контурным фильтром (S710) и затем сохраняется в буфере изображения (S740), и вектор движения подвергается процессу ограничения динамического диапазона (S720) и затем сохраняется в буфере вектора движения (S540).

К тому же, ссылаясь на фиг.8, реконструированное изображение подвергается
20 обработке контурным фильтром (S810) и затем сохраняется в буфере изображения (S840), и вектор движения подвергается процессу ограничения динамического диапазона (S820) и процессу уменьшения пространственного разрешения (S830) и затем сохраняется (S850).

При этом в примерных вариантах осуществления согласно фиг.6 и 8,
25 последовательность из процесса ограничения динамического диапазона S620 или S820 и процесса уменьшения пространственного разрешения S630 и S830 не ограничивается, и может быть изменена.

К тому же, для того, чтобы дополнительно уменьшить полосу пропускания для доступа к памяти, динамические диапазоны каждой составляющей вектора движения
30 могут быть ограничены по-разному. Например, только один из динамического диапазона X-составляющей и динамического диапазона Y-составляющей может быть ограничен, или динамический диапазон Y-составляющей может быть дополнительно ограничен по сравнению с динамическим диапазоном X-составляющей.

Ограниченный динамический диапазон вектора движения передается посредством
35 набора параметров последовательности, набора параметров картинки (PPS), заголовка слайса (среза) и тому подобного, и декодер аналогично выполняет ограничение динамического диапазона временного вектора движения в последовательности, картинке или слайсе. В этом случае может также быть передана глубина в битах, которая является размером пространства памяти, требуемого для хранения вектора движения,
40 представленного в динамическом диапазоне. К тому же, возможно эффективно хранить временный вектор движения, с тем чтобы совпадал с характеристиками движения изображения, с использованием динамического диапазона, передаваемого посредством набора параметров последовательности, набора параметров картинки, заголовка слайса или подобного, вместо хранения вектора движения с использованием глубины
45 в битах, имеющей фиксированный размер.

При этом вектор движения может быть квантован и сохранен. В случае, в котором вектор движения квантован и сохранен, уменьшается точность вектора движения. В качестве способа квантования существуют равномерное квантование, при котором

размеры шага являются равномерными, неравномерное квантование, при котором размеры шага не являются равномерными, и подобные. Размер шага при квантовании задается в фиксированное значение, предварительно заданное между кодером или декодером, или передается из кодера в декодер посредством набора параметров последовательности, набора параметров картинки, заголовка слайса или подобного. Декодер использует квантованный вектор движения как он есть или деквантует и использует квантованный вектор движения. Фиг.9 показывает пример квантования вектора движения. Ссылаясь на Фиг.9, в случае, в котором вектор движения имеет значение составляющей от 32 до 48, вектор движения квантуется до 40.

К тому же, вектор движения может быть ограничен в разрешении представления и сохранен. Разрешение представления означает единицу целого пикселя (1 единица пикселя), единицу дробного пикселя (единица 1/2 пикселя, единица 1/4 пикселя или подобное). Например, разрешение вектора движения, обработанного в единице 1/4 пикселя, может быть сохранено как целый пиксель. Разрешение представления вектора движения задается в фиксированное значение, предварительно заданное между кодером или декодером, или передается из кодера в декодер посредством набора параметров последовательности, набора параметров картинки, заголовка слайса или подобного.

К тому же, только по отношению к некоторым векторам движения из числа временных векторов движения, хранящихся в памяти, могут быть выполнены процесс ограничения динамического диапазона, процесс уменьшения пространственного разрешения и процесс квантования вектора движения.

В случае, в котором динамический диапазон вектора движения ограничен и сохранен, информация о динамическом диапазоне вектора движения может быть добавлена и сохранена в памяти. Например, в случае, в котором динамический диапазон вектора движения составляет от -128 до +127, может быть дополнительно сохранен флаг со значением 1, и в случае, в котором динамический диапазон вектора движения составляет от -32 до +31, может быть дополнительно сохранен флаг со значением 0. В этом случае информация флага может быть сохранена вместе с вектором движения или может быть сохранена в памяти, отличной от памяти, в которой хранится вектор движения. В случае, в котором информация флага и вектор движения хранятся в разной памяти, когда распознано в каком динамическом диапазоне хранится конкретный вектор движения, может быть обеспечена возможность доступа к информации флага. К тому же, информация о том, в каком динамическом диапазоне хранятся некоторые векторы движения, передается посредством набора параметров последовательности, набора параметров картинки, заголовка слайса или подобного, тем самым делая возможным обеспечение возможности декодеру выполнить операцию, аналогичную операции кодера.

В случае, в котором пространственное разрешение вектора движения уменьшено и сохранено, информация о размере блока вектора движения может быть добавлена и сохранена в памяти. Например, в случае, в котором размер блока вектора движения составляет 4×4, может быть дополнительно сохранен флаг со значением 1, и в случае, в котором размер блока вектора движения составляет 16×16, может быть дополнительно сохранен флаг со значением 0. В этом случае информация флага может быть сохранена вместе с вектором движения или может быть сохранена в памяти, отличной от памяти, в которой хранится вектор движения. В случае, в котором информация флага и вектор движения хранятся в разной памяти, когда распознано в каком размере блока хранится конкретный вектор движения, может быть обеспечена возможность доступа к информации флага. К тому же, информация о том, в каком размере блока хранятся

некоторые векторы движения, передается посредством набора параметров последовательности, набора параметров картинки, заголовка слайса или подобного, тем самым давая возможность декодеру выполнить операцию, аналогичную операции кодера.

5 В случае, в котором вектор движения квантован и сохранен, информация о точности вектора движения может быть добавлена и сохранена в памяти. Например, в случае, в котором размер шага квантования составляет 4, может быть дополнительно сохранен флаг со значением 1, и в случае, в котором размер шага квантования составляет 1, может быть дополнительно сохранен флаг со значением 0. В этом случае информация
10 флага может быть сохранена вместе с вектором движения или может быть сохранена в памяти, отличной от памяти, в которой хранится вектор движения. В случае, в котором информация флага и вектор движения хранятся в разной памяти, когда распознано с каким размером шага квантован и сохранен конкретный вектор движения, может быть обеспечена возможность доступа к информации флага. К тому же, информация о том,
15 с каким размером шага квантованы и сохранены некоторые векторы движения, передается посредством набора параметров последовательности, набора параметров картинки, заголовка слайса или подобного, тем самым делая возможным обеспечение возможности декодеру выполнить операцию, аналогичную операции кодера.

К тому же, в случае, в котором информация движения хранится в памяти,
20 пространственное разрешение вектора движения может быть уменьшено и сохранено. Здесь, информация движения включает в себя по меньшей мере одно из информации режима внешнего предсказания, указывающей индекс опорной картинки, вектор движения, однонаправленный, двунаправленный или подобное, требуемой в момент внешнего предсказания, списка опорных картинок и информации режима предсказания
25 о том, выполняется ли режим внутреннего предсказания или режим внешнего предсказания.

Например, информация единицы предсказания, имеющей наибольший размер сегмента среди множества информации движения конкретной области, может быть сохранена в памяти как репрезентативная информация движения. Здесь, конкретная область может
30 включать в себя область в древовидном блоке кодирования и области соседних блоков древовидного блока кодирования. К тому же, конкретная область может быть областью, включающей в себя блок, в котором хранится информация движения, в случае, в котором вся картинка, или слайс, разделена на предварительно определенный размер.

Например, после того как информация движения, которая закодирована в способе
35 слияния информации движения, способе пропуска информации кодирования или подобном, исключена из множества информации движения, включенной в конкретную область, репрезентативная информация движения может быть сохранена в памяти.

Например, наиболее часто генерируемая информация движения среди множества информации движения, включенной в конкретную область, может быть сохранена как
40 репрезентативная информация движения в памяти. В этом случае может быть вычислено число генераций информации движения для каждого размера блока или подобное.

Например, информация движения в конкретном положении среди множества информации движения, включенной в конкретную область, может быть сохранена. Здесь, конкретное положение, которое является положением, включенным в конкретную
45 область, может быть фиксированным положением конкретной области. К тому же, конкретное положение может быть выбрано как одно из множества положений. Когда используются множество положений, может быть определен приоритет для каждого из положений, и информация движения может быть сохранена в памяти согласно

приоритету.

Например, когда множество информации движения, включенной в конкретную область, хранится в памяти, так как информация движения не существует снаружи границы блока, кодированного в режиме внутреннего предсказания, блока, кодированного в режиме импульсно-кодовой модуляции, слайса или картинки, информация движения соответствующего положения может быть не сохранена в памяти.

В вышеупомянутых примерах, когда информация движения конкретного положения сохранена, в случае, в котором информация движения соответствующего положения не существует, информация положения совмещенного блока, информация положения уже кодированного блока или информация движения соседнего блока может быть использована в качестве информации движения соответствующего положения. Здесь, конкретное положение может быть положением выборки в соседнем блоке или положением данного блока. Например, в случае, в котором информация движения конкретного положения не существует, промежуточное значение или среднее значение среди информации движения соседних блоков, которые кодированы с внешним предсказанием, может быть сохранено в памяти. Например, в случае, в котором информация движения конкретного положения не существует, среднее значение информации движения соседних блоков может быть сохранено в памяти. Когда вычислены промежуточное значение и среднее значение, в случае, в котором информация движения соседних блоков отличается от по меньшей мере одного из индекса опорной картинки, списка опорных картинок и информации режима внешнего предсказания, размер вектора движения может быть отрегулирован согласно индексу опорной картинки, списку опорных картинок, информации режима внешнего предсказания, порядковому номеру картинки и тому подобному.

III. Процесс получения вектора движения

В случае, в котором информация движения сохраняется в памяти с использованием вышеупомянутых способов информации движения, и информация движения опорной картинки используется в способе предсказания вектора движения, способе усовершенствованного предсказания вектора движения или способе слияния информации движения, сохраненная информация может быть извлечена.

Например, может быть извлечена информация движения положения, соответствующего положению древовидного блока кодирования в опорной картинке. В этом случае положение, соответствующее положению древовидного блока кодирования в опорной картинке, может быть фиксированным положением в конкретной области или относительном положении исходя из положения древовидного блока кодирования.

Фиг.10-13 показывают примеры извлечения информации движения из опорной картинки.

На фиг.10-13, блок X указывает древовидный блок кодирования в картинке 1010, 1110, 1210 или 1310 объектов декодирования, и блок А, блок В, блок С, блок D и блок Е указывает реконструированные соседние блоки. К тому же, блок Т в опорной картинке 1020, 1120, 1220 и 1320 указывает совмещенный блок, соответствующий древовидному блоку кодирования. Блок Y в опорной картинке 1320 согласно фиг.13 указывает блок, соответствующий положению, отличному от блока объектов кодирования/декодирования.

Ссылаясь на фиг.10, может быть извлечена информация движения, соответствующая положению, соответствующему положению верхнего левого пикселя, среди положений древовидного блока X кодирования в опорной картинке.

Ссылаясь на фиг.11, может быть извлечена информация движения, соответствующая положению, соответствующему положению центрального пикселя, среди положений древовидного блока X кодирования в опорной картинке.

Ссылаясь на фиг.12, может быть извлечена информация движения, соответствующая положению, соответствующему положению правого нижнего пикселя, среди положений древовидного блока X кодирования в опорной картинке.

Ссылаясь на фиг.13, может быть извлечена информация движения, соответствующая положению, соответствующему положению пикселя, отличного от древовидного блока X кодирования в опорной картинке.

Способ кодирования/декодирования, такой как предсказание вектора движения, усовершенствованное предсказание вектора движения, слияние информации движения, пропуск слияния информации движения или подобный, может быть выполнен с использованием информации движения, хранящейся в памяти, то есть, информации движения опорной картинки.

Вектор движения может быть сохранен в памяти с использованием по меньшей мере одного из способа ограничения динамического диапазона вектора движения, способа уменьшения пространственного разрешения вектора движения, способа квантования вектора движения и способа уменьшения разрешения представления вектора движения, и сохраненный вектор движения может быть использован для предсказания вектора движения древовидного блока кодирования и слияния их информации движения.

Процесс извлечения вектора движения опорной картинки из памяти называется процессом получения временного вектора движения. В процессе получения временного вектора движения, TMVbitWidth указывает глубину в битах временного вектора движения, хранящегося в памяти.

Входными данными в процессе получения временного вектора движения являются местоположение (xP, yP), задающее верхнюю левую выборку яркости текущей единицы предсказания относительно верхней левой выборки текущей картинки, переменные, задающие ширину и высоту единицы предсказания для яркости, nPSW и nPSH, опорный индекс текущего сегмента единицы предсказания refIdxLX (с X, составляющим 0 или 1). Выходными данными в процессе являются предсказание вектора движения mvLXCol и флаг доступности availableFlagLXCol.

Функция RefPicOrderCnt(pic, refIdx, LX) задается значением PicOrderCnt картинки, которая является опорной картинкой RefPicListX[refIdx] из pic с X, составляющим 0 или 1. PicOrderCnt из опорной картинки должно быть обеспечено пока картинка не отмечена как "несуществующая". Clip3(a, b, c) означает функцию усечения c, так чтобы существовать в диапазоне между a и b.

Если slice_type равен B и collocated_from_10_flag равен 0, переменная colPic определяет картинку, которая содержит совмещенный сегмент, как задано посредством RefPicList1[0]. Иначе (slice_type равен B и collocated_from_10_flag равен 1 или slice_type равен P), переменная colPic определяет картинку, которая содержит совмещенный сегмент, как задано посредством RefPicList0[0].

Переменная colPu и ее положение (xPCol, yPCol) получают на следующих упорядоченных этапах:

1. Правое нижнее положение яркости (xPRb, yPRb) текущей единицы предсказания задается как представлено уравнениями 8 и 9.

[Уравнение 8]

$$xPRb = xP + nPSW$$

[Уравнение 9]

$yPRb = yP + nPSH$

2. Если colPu кодируется в режиме внутреннего предсказания или colPu недоступна, (1) Центральное положение яркости текущей единицы предсказания задается как представлено уравнениями 10 и 11.

5 [Уравнение 10]

$xPCtr = (xP + (nPSW \gg 1) - 1$

[Уравнение 11]

$yPCtr = (yP + (nPSH \gg 1) - 1$

(2) Переменная colPu задается как единица предсказания, покрывающая измененное положение, заданное посредством $((xPCtr \gg 4) \ll 4, (yPCtr \gg 4) \ll 4)$ внутри colPic.

3. $(xPCol, yPCol)$ задается равным верхней левой выборке яркости для colPu относительно верхней левой выборки яркости для colPic.

Переменные mvLXCol и availableFlagLXCol получают как следует ниже.

1. Если colPu кодируется в режиме внутреннего предсказания или colPu недоступна, 15 обе составляющие mvLXCol задаются равными 0, и availableFlagLXCol задается равной 0.

2. Иначе (colPu не кодируется в режиме внутреннего предсказания и colPu доступна), переменные mvCol и refIdxCol получают как следует ниже,

(1) Если PredFlagL0[xPCol][yPCol] равна 0, вектор движения mvCol и опорный индекс 20 refIdxCol задаются равными MvL1[xPCol][yPCol] и RefIdxL1 [xPCol][yPCol], соответственно.

(2) Иначе (PredFlagL0[xPCol][yPCol] равна 1), применяется следующее.

1) Если PredFlagL1[xPCol][yPCol] равна 0, вектор движения mvCol и опорный индекс 25 refIdxCol задаются равными MvL0[xPCol][yPCol] и RefIdxL0[xPCol][yPCol], соответственно.

2) Иначе (PredFlagL1[xPCol][yPCol] равна 1), применяется следующее.

а. Следующие назначения сделаны с X, составляющим 0 или 1.

i. $RefIdxColLX = RefIdxLX[xPCol][yPCol]$

ii. Если PicOrderCnt(colPic) меньше, чем PicOrderCnt(currPic), и RefPicOrderCnt(colPic, 30 RefIdxColLX, LX) больше, чем PicOrderCnt(currPic), или PicOrderCnt(colPic) больше, чем PicOrderCnt(currPic), и RefPicOrderCnt(colPic, RefIdxColLX, LX) меньше, чем PicOrderCnt(currPic), переменная MvXCross равна 1.

iii. Иначе (PicOrderCnt(colPic) меньше, чем PicOrderCnt(currPic), и RefPicOrderCnt(colPic, 35 RefIdxColLX, LX) меньше, чем PicOrderCnt(currPic) или равна ей, или PicOrderCnt(colPic) больше, чем PicOrderCnt(currPic), и RefPicOrderCnt(colPic, RefIdxColLX, LX) больше, чем PicOrderCnt(currPic) или равна ей), переменная MvXCross равна 0.

б. Если одно из следующих условий верно, вектор движения mvCol, опорный индекс refIdxCol и ListCol задаются равными MvL1[xPCol][yPCol], RefIdxColL1 и L1, 40 соответственно.

i. Mv0Cross равна 0 и Mv1Cross равна 1.

ii. Mv0Cross равна Mv1Cross и список опорных индексов равен L1

с. Иначе, вектор движения mvCol, опорный индекс refIdxCol и ListCol задаются равными MvL0[xPCol][yPCol], RefIdxColL0 и L0, соответственно.

3) переменная availableFlagLXCol задается равной 1, и применяются операции 45 уравнения 12 и уравнений с 13 по 18.

а. Если $PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, ListCol)$ равна $PicOrderCnt(currPic) - RefPicOrderCnt(currPic, refIdxLX, LX)$,

[Уравнение 12]

$mvLXCol = Clip3(-1 << (TMVBitWidth - 1), 1 << (TMVBitWidth - 1) - 1, mvCol)$

b. Иначе

[Уравнение 13]

$tx = (16384 + Abs(td/2))/td$

5 [Уравнение 14]

$DistScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6)$

[Уравнение 15]

$mvLXCol = Clip3(-1 << (TMVBitWidth - 1), 1 << (TMVBitWidth - 1) - 1, mvCol)$

[Уравнение 16]

10 $mvLXCol = ClipMv((DistScaleFactor * mvLXCol + 128) >> 8)$ где td и tb получают как в уравнениях 17 и 18.

[Уравнение 17]

$td = Clip3(-128, 127, PicOrderCnt(colPic) - RefPicOrderCnt(colPic, refIdxCol, ListCol))$

[Уравнение 18]

15 $tb = C / i > 3(-128, 127, PicOrderCnt(currPic) - RefPicOrderCnt(currPic, refIdxLX, LX))$

То есть ссылаясь на уравнения 13-16, $mvLXCol$ получают как масштабированную версию вектора движения $mvCol$.

При этом даже если вектор движения усечен в динамическом диапазоне, в случае, в котором усеченный вектор движения масштабирован, усеченный вектор движения
20 может быть снова вне динамического диапазона. Вследствие этого, после получения масштабированного вектора движения, динамический диапазон вектора движения может быть ограничен. В этом случае каждое из уравнений 15 и 16 может быть заменено уравнениями 19 и 20.

[Уравнение 19]

25 $mvLXCol = ClipMv(DistScaleFactor * mvLXCol + 128) >> 8$

[Уравнение 20]

$mvLXCol = Clip3(-1 << (TMVBitWidth - 1), 1 << (TMVBitWidth - 1) - 1, mvCol)$

IV. Способ передачи информации для усечения временного вектора движения в декодере.

30 В дальнейшем будет описан способ передачи информации, требуемой для усечения временного вектора движения в декодере, с использованием того же способа, как и способ кодера.

$TMVBitWidth$ в процессе получения временного вектора движения может быть передана из кодера в декодер посредством набора параметров последовательности,
35 набора параметров картинки, заголовка слайса или подобного.

Таблица 5

seq_parameter_set_rbsp() {	C'	Дескриптор
.....		
bit_width_temporal_motion_vector_minus8	0	se(v)
.....		
rbp_trailing_bits()	0	
}		

40 $bit_width_temporal_motion_vector_minus8$ по таблице 5 задает ширину в битах составляющей временного вектора движения. Когда $bit_width_temporal_motion_vector_minus8$ не присутствует, следует предполагать, что она равна 0. Ширина в битах составляющей временного вектора движения задается как следует ниже:

[Уравнение 21]

$TMVBitWidth = bit_width_temporal_motion_vector_minus8 + 8(?)$

1. Способ 1 передачи информации - в случае, в котором вектор движения сжат и глубина в битах вектора движения ограничена

Таблица 6

seq_parameter_set_rbsp() {	C'	Дескриптор
.....		
motion_vector_buffer_comp_flag	0	u(1)
if (motion_vector_buffer_comp_flag)		
motion_vector_buffer_comp_ratio_log2	0	u(8)
bit_depth_temporal_motion_vector_constraint_flag	0	u(1)
if (bit_depth_temporal_motion_vector_constraint_flag)		
bit_depth_temporal_motion_vector_minus8	0	se(v)
.....		
rbsp_trailing_bits()	0	
}		

Ссылаясь на таблицу 6, в случае, в котором motion_vector_buffer_comp_flag равняется 1, применяется процесс сжатия буфера вектора движения.

motion_vector_buffer_comp_ratio_log2 задает коэффициент сжатия в процессе сжатия буфера вектора движения. Когда motion_vector_buffer_comp_ratio_log2 не присутствует, следует предполагать, что он равен 0. Коэффициент сжатия буфера вектора движения задается как следует ниже:

[Уравнение 22]

$$MVBufferCompRatio = 1 << \text{motion_vector_buffer_comp_ratio_log2}$$

Снова ссылаясь на таблицу 6, в случае, в котором bit_depth_temporal_motion_vector_constraint_flag равняется 1, применяется процесс ограничения глубины в битах временного вектора движения.

bit_depth_temporal_motion_vector_minus8 задает глубину в битах временного вектора движения. Когда bit_depth_temporal_motion_vector_minus8 не присутствует, следует предполагать, что она равна 0. Глубина в битах временного вектора движения задается как следует ниже:

[Уравнение 23]

$$TMVBitDepth = \text{bit_depth_temporal_motion_vector_minus8} + 8$$

2. Способ 2 передачи информации - в случае, в котором глубина в битах вектора движения ограничена

Таблица 7

seq_parameter_set_rbsp() {	C'	Дескриптор
.....		
bit_depth_temporal_motion_vector_constraint_flag	0	u(1)
if (bit_depth_temporal_motion_vector_constraint_flag)		
bit_depth_temporal_motion_vector_minus8	0	se(v)
.....		
rbsp_trailing_bits()	0	
}		

Ссылаясь на таблицу 7, в случае, в котором bit_depth_temporal_motion_vector_constraint_flag равняется 1, применяется процесс ограничения глубины в битах временного вектора движения.

bit_depth_temporal_motion_vector_minus8 задает глубину в битах временного вектора движения. Когда bit_depth_temporal_motion_vector_minus8 не присутствует, следует предполагать, что она равна 0. Глубина в битах временного вектора движения задается как следует ниже:

[Уравнение 24]

$TMVBitDepth = bit_depth_temporal_motion_vector_minus8 + 8$

3. Способ 3 передачи информации - в случае, в котором глубина в битах вектора движения ограничена

Таблица 8

seq_parameter_set_rbsp() {	C'	Дескриптор
.....		
bit_depth_temporal_motion_vector_minus8	0	se(v)
.....		
rbp_trailing_bits()	0	
}		

bit_depth_temporal_motion_vector_minus8 задает глубину в битах временного вектора движения. Когда bit_depth_temporal_motion_vector_minus8 не присутствует, следует предполагать, что она равна 0. Глубина в битах временного вектора движения задается как следует ниже:

[Уравнение 25]

$TMVBitDepth = bit_depth_temporal_motion_vector_minus8 + 8$

4. Способ 4 передачи информации - в случае, в котором глубина в битах ограничена по отношению к каждому из X- и Y-составляющих вектора движения

Таблица 9

seq_parameter_set_rbsp() {	C'	Дескриптор
.....		
bit_depth_temporal_motion_vector_constraint_flag	0	u(1)
if(bit_depth_temporal_motion_vector_constraint_flag) {		
bit_depth_temporal_motion_vector_x_minus8	0	se(v)
bit_depth_temporal_motion_vector_y_minus8	0	se(v)
}		
.....		
rbp_trailing_bits()	0	
}		

Ссылаясь на таблицу 9, в случае, в котором bit_depth_temporal_motion_vector_constraint_flag равняется 1, применяется процесс ограничения глубины в битах временного вектора движения.

bit_depth_temporal_motion_vector_x_minus8 задает глубину в битах составляющей x временного вектора движения. Когда bit_depth_temporal_motion_vector_x_minus8 не присутствует, следует предполагать, что она равна 0. Глубина в битах составляющей x временного вектора движения задается как следует ниже:

[Уравнение 26]

$TMVXBitDepth = bit_depth_temporal_motion_vector_x_minus8 + 8$

bit_depth_temporal_motion_vector_y_minus8 задает глубину в битах составляющей y временного вектора движения. Когда bit_depth_temporal_motion_vector_y_minus8 не присутствует, следует предполагать, что она равна 0. Глубина в битах составляющей y временного вектора движения задается как следует ниже:

[Уравнение 27]

$TMVXBitDepth = bit_depth_temporal_motion_vector_y_minus8 + 8$

5. Способ 5 передачи информации - в случае, в котором вектор движения сжат и глубина в битах вектора движения ограничена

Таблица 10

seq_parameter_set_rbsp() {	C	Дескриптор
.....		
motion_vector_buffer_comp_flag	0	u(1)
if(motion_vector_buffer_comp_flag) {		
motion_vector_buffer_comp_ratio_log2	0	u(8)
bit_depth_temporal_motion_vector_minus8	0	se(v)
}		
.....		
rbsp_trailing_bits()	0	
}		

Ссылаясь на таблицу 10, в случае, в котором motion_vector_buffer_comp_flag равняется 1, применяется процесс сжатия буфера вектора движения.

motion_vector_buffer_comp_ratio_log2 задает коэффициент сжатия в процессе сжатия буфера вектора движения. Когда motion_vector_buffer_comp_ratio_log2 не присутствует, следует предполагать, что она равна 0. Коэффициент сжатия буфера вектора движения задается как следует ниже:

[Уравнение 28]

$MVBufferCompRatio = 1 \ll motion_vector_buffer_comp_ratio_log2$

V. Задание динамического диапазона посредством уровней видеокодека

Динамический диапазон временного вектора движения может быть задан посредством уровня видеокодека, вместо того, чтобы передаваться посредством набора параметров последовательности, набора параметров картинки или заголовка слайса. Кодер и декодер могут определять ограниченный диапазон вектора движения с использованием информации уровня.

К тому же, даже в уровнях, динамические диапазоны и/или глубины в битах каждой из X- и Y-составляющих вектора движения могут быть заданы по-разному, и могут быть заданы минимальные значения и максимальные значения каждой из составляющих.

Таблицы 11 и 12 показывают пример случая, в котором TMVBitWidth в процессе получения временного вектора движения, описанного выше, задается в уровнях.

Таблица 11	
Номер уровня	MaxTMVBitWidth (Максимальная ширина в битах составляющей временного MV)
1	8
1b	8
1,1	8
1,2	8
1,3	8
2	8
2,1	8
2,2	8
3	8
3,1	10
3,2	10
4	10
4,1	10
4,2	10
5	10
5,1	10

Ссылаясь на таблицу 11, TMVBitWidth задается как MaxTMVBitWidth, заданная в уровнях. Здесь, MaxTMVBitWidth указывает максимальную ширину в битах временного

вектора движения, когда временный вектор движения хранится в памяти.

При этом TMVBitWidth может также быть задана в уровнях, и отличие от заданного значения (дельта значение) может быть передано посредством набора параметров последовательности, набора параметров картинки или заголовка слайса. То есть TMVBitWidth может быть задана как значение, полученное посредством прибавления отличия, переданного посредством набора параметров последовательности, набора параметров картинки или заголовка, к MaxTMVBitWidth, заданной в уровнях. Здесь, TMVBitWidth указывает ширину в битах временного вектора движения, когда временный вектор движения хранится в памяти.

Таблица 12	
Номер уровня	MaxTMVBitWidth (Максимальная глубина в битах составляющей временного MV)
1	8
1b	8
1,1	8
1,2	8
1,3	8
2	8
2,1	8
2,2	8
3	8
3,1	10
3,2	10
4	10
4,1	10
4,2	10
5	10
5,1	10

Таблица 13

seq_parameter_set_rbsp() {	C	Дескриптор
.....		
delta_bit_width_temporal_motion_vector_minus8	0	se(v)
.....		
rbp_trailing_bits()	0	
}		

delta_bit_width_temporal_motion_vector_minus8 задает дельта ширину в битах составляющей временного вектора движения. Когда delta_bit_width_temporal_motion_vector_minus8 не присутствует, следует предполагать, что она равна 0. Ширина в битах составляющей временного вектора движения задается как следует ниже:

[Уравнение 29]

$$\text{TMVBitWidth} = \text{delta_bit_width_temporal_motion_vector_minus8} + \text{MaxTMVBitWidth}$$

К тому же, как показано в таблице 14, динамический диапазон каждой составляющей временного вектора движения может также быть задан в уровнях.

Таблица 14								
Номер уровня	MaxMBPS (Максимальная скорость обработки макроблока) (МБ/с)	MaxFS (Максимальный размер картинки) (МБ)	MaxDpbMbs (Максимальный размер буфера декодированного изображения) (МБ)	MaxBR (Максимальная битовая скорость видео) (1000 бит/с, 1200 бит/с, cpbBrVclFactor бит/с, или cpbBrNalFactor бит/с)	MaxCPB (Максимальный размер CPB) (1000 битов, 1200 битов, cpbBrVclFactor битов, или cpbBrNalFactor битов)	MaxTmvR (Максимальный диапазон составляющей временного MV) (выборки картинки по яркости)	MinCR (Минимальный коэффициент сжатия)	MaxMvsPer2Mb (Максимальное число векторов движения на два последующих МБ)

5	1	1485	99	396	64	175	[-64, +63,75]	2	-
	1b	1485	99	396	128	350	[-64, +63,75]	2	-
	1,1	3000	396	900	192	500	[-128, +127,75]	2	-
	1,2	6000	396	2376	384	1000	[-128, +127,75]	2	-
	1,3	11880	3%	2376	768	2000	[-128, +127,75]	2	-
10	2	11880	396	2376	2000	2000	[-128, +127,75]	2	-
	2,1	19800	792	4752	4000	4000	[-256, +255,75]	2	-
	2,2	20250	1620	8100	4000	4000	[-256, +255,75]	2	-
	3	40500	1620	8100	10000	10000	[-256, +255,75]	2	32
	3,1	106000	3600	18000	14000	14000	[-512, +511,75]	4	16
15	3,2	216000	5120	20480	20000	20000	[-512, +511,75]	4	16
	4	245760	8192	32768	20000	25000	[-512, +511,75]	4	16
	4,1	245760	8192	32768	50000	62500	[-512, +511,75]	2	16
	4,2	522240	8704	34816	50000	62500	[-512, +511,75]	2	16
	5	589824	22080	110400	135000	135000	[-512, +511,75]	2	16
	5,1	983040	36864	184320	240000	240000	[-512, +511,75]	2	16

К тому же, как показано в таблице 15-17, ширина в битах каждой составляющей временного вектора движения может также быть задана в уровнях.

Таблица 15

Номер уровня	MaxMBPS (Максимальная скорость обработки макроблока) (МБ/с)	MaxFS (Максимальный размер картинки) (МБ)	MaxDpbMbs (Максимальный размер буфера декодированного изображения) (МБ)	MaxBR (Максимальная битовая скорость видео) (1000 бит/с, 1200 бит/с, cpbBrVclFactor бит/с, или cpbBrNalFactor бит/с)	MaxCPB (Максимальный размер CPB) (1000 битов, 1200 битов, cpbBrVclFactor битов, или cpbBrNalFactor битов)	MaxTMVBitWidth (Максимальная ширина в битах составляющей временного MV)	MinCR (Минимальный коэффициент сжатия)	MaxMvsPer2Mb (Максимальное число векторов движения на два последующих МБ)
1	1485	99	396	64	175	8	2	-
1b	1485	99	3%	128	350	8	2	-
1,1	3000	396	900	192	500	8	2	-
1,2	6000	396	2376	384	1000	8	2	-
1,3	11880	396	2376	768	2000	8	2	-
2	11880	396	2376	2000	2000	8	2	-
2,1	19800	792	4752	4000	4000	8	2	-
2,2	20250	1620	8100	4000	4000	8	2	-
3	40500	1620	8100	10000	10000	8	2	32
3,1	108000	3600	18000	14000	14000	10	4	16
3,2	216000	5120	20480	20000	20000	10	4	16
4	245760	8192	32768	23000	25000	10	4	16
4,1	245760	8192	32768	50000	62500	10	2	16
4,2	522240	8704	34816	50000	62500	10	2	16
5	589824	22080	110400	135000	135000	10	2	16
5,1	983040	36864	184320	240000	240000	10	2	16

Таблица 16

Номер уровня	MaxMBPS (Максимальная скорость обработки макроблока) (МБ/с)	MaxFS (Максимальный размер картинки) (МБ)	MaxDpbMbs (Максимальный размер буфера декодированного изображения) (МБ)	MaxBR (Максимальная битовая скорость видео) (1000 бит/с, 1200 бит/с, cpbBrVclFactor бит/с, или cpbBrNalFactor бит/с)	MaxCPB (Максимальный размер CPB) (1000 битов, 1200 битов, cpbBrVclFactor битов, или cpbBrNalFactor битов)	MaxTMVBitWidth (Максимальная ширина в битах составляющей временного MV)	MinCR (Минимальный коэффициент сжатия)	MaxMvsPer2Mb (Максимальное число векторов движения на два последующих МБ)
1	1485	99	396	64	175	6	2	-
1b	1485	99	396	128	350	6	2	-
1,1	3000	396	900	192	500	7	2	-
1,2	6000	396	2376	384	1000	7	2	-
1,3	11880	396	2376	768	2000	7	2	-

2	11880	396	2376	2000	2000	7	2	-
2,1	19800	792	4752	4000	4000	8	2	-
2,2	20250	1620	8100	4000	4000	8	2	-
3	40500	1620	8100	10000	10000	8	2	32
3,1	108000	3600	18000	14000	14000	10	4	16
3,2	216000	5120	20480	28900	20000	10	4	16
4	245760	8192	32768	20000	25000	10	4	16
4,1	245760	8192	32768	50000	62500	10	2	16
4,2	522240	8704	34816	50000	62500	10	2	16
5	589824	22080	110400	135000	135000	10	2	16
5,1	983040	36864	184320	240000	240000	10	2	16

Таблица 17	
Номер уровня	MaxTMVBitWidth (Максимальная ширина в битах составляющей временного MV)
1	8
1b	8
1,1	8
1,2	8
1,3	8
2	8
2,1	8
2,2	8
3	8
3,1	10
3,2	10
4	10
4,1	10
4,2	10
5	10
5,1	10

К тому же, как показано в таблице 18, ширина в битах Y-составляющей временного вектора движения может также быть задана в уровнях.

Таблица 18

Номер уровня	MaxMBPS (Максимальная скорость обработки макроблока) (МБ/с)	MaxFS (Макси- мальный размер картинки) (МБ)	MaxDpbMbs (Максимальный размер буфера декодированного изображения) (МБ)	MaxBR (Максимальная битовая скорость видео) (1000 бит/с, 1200 бит/с, или сrbBrVclFactor бит/с, или сrbBrNalFactor бит/с)	MaxCPB (Максимальный размер CPB) (1000 битов, 1200 битов, сrbBrVclFactor битов, или сrbBrNalFactor битов)	MaxTMVYBitWidth (Максимальная ширина в битах вертикальной составляющей временного MV)	MinCR (Минималь- ный коэффи- циент сжатия)	MaxMvsPer2Mb (Максимальное число векторов движения на два последующих МБ)
1	1485	99	396	64	175	8	2	-
1b	1485	99	396	128	350	8	2	-
1,1	3000	396	900	192	500	8	2	-
1,2	6000	396	2376	384	1000	8	2	-
1,3	11883	396	2376	768	2000	8	2	-
2	11880	396	2376	2000	2000	8	2	-
2,1	19800	792	4752	4000	4000	8	2	-
2,2	20250	1620	8100	4000	4000	8	2	-
3	40500	1620	8100	10000	10000	8	2	32
3,1	108000	3600	18000	14000	14000	10	4	16
3,2	216000	5120	20480	20000	20000	10	4	16
4	245760	8192	32768	20000	25000	10	4	16
4,1	245760	8192	32768	50000	62500	10	2	16
4,2	522240	8704	34816	50000	62000	10	2	16
5	589824	22080	110400	135000	135000	10	2	16
5,1	983040	36864	184320	240000	240000	10	2	16

К тому же, динамический диапазон временного вектора движения может быть задан как фиксированное значение, предварительно заданное между кодером и декодером, без передачи информации об ограничении вектора движения или сохранен в виде фиксированной глубины в битах.

В случае, в котором TMVBitWidth привязана к тому же значению и используется в кодере и декодере, TMVBitWidth может быть положительным целым числом, таким как 4, 6, 8, 10, 12, 14, 16 или подобным. Здесь, TMVBitWidth указывает ширину в битах временного вектора движения, когда временный вектор движения хранится в памяти.

Фиг.14 является схемой последовательности операций, показывающей способ кодирования изображения согласно примерному варианту осуществления настоящего изобретения. Ссылаясь на фиг.14, способ кодирования изображения включает в себя этап усечения (S1410), этап сохранения (S1420) и этап кодирования (S1430).

Устройство кодирования изображения и/или устройство декодирования изображения отсекают вектор движения опорной картинки в предварительно определенном динамическом диапазоне (S1410). Как описано выше посредством "I. Процесс усечения вектора движения", вектор движения, который выходит за динамический диапазон, представляется минимальным значением или максимальным значением соответствующего динамического диапазона. Вследствие этого, как описано выше посредством "IV. Способ передачи информации для усечения временного вектора движения в декодере" и "V. Задание динамического диапазона посредством уровней видеокodeка", глубина в битах ограничивается посредством уровня видеокodeка, набора параметров последовательности и подобного, или динамический диапазон ограничивается посредством уровня видеокodeка, тем самым делая возможным усечение вектора движения опорной картинки в предварительно определенном динамическом диапазоне.

Устройство кодирования изображения и/или устройство декодирования изображения сохраняет усеченный вектор движения опорной картинки в буфере, как описано выше посредством "II. Процесс сохранения вектора движения" (S1420). Вектор движения может быть сохранен в буфере вместе с реконструированным изображением или отдельно от него.

Устройство кодирования изображения кодирует вектор движения древовидного блока кодирования с использованием сохраненного вектора движения опорной картинки (S1430). Как описано выше посредством "III. Процесс получения вектора движения", в способе усовершенствованного предсказания вектора движения, используемом в HEVC, может быть использован вектор движения блока, существующий в положении, которое является таким же или соответствует положению древовидного блока кодирования в опорной картинке, также как и векторы движения реконструированных блоков, расположенных вокруг древовидного блока кодирования. Вследствие этого вектор движения древовидного блока кодирования может также быть вектором движения опорной картинки, то есть временным вектором движения, также как и векторы движения соседних блоков, смежных с древовидным блоком кодирования.

При этом, так как динамический диапазон X-составляющей и Y-составляющей вектора движения опорной картинки могут быть заданы по-разному, каждая составляющая вектора движения опорной картинки может быть усечена в каждом динамическом диапазоне.

К тому же, может быть использован способ сжатия вектора опорной картинки, также как и способ ограничения динамического диапазона вектора движения опорной картинки. В случае ограничения динамического диапазона вектора движения опорной

картинки или сжатия вектора движения опорной картинки, флаг, указывающий динамический диапазон, и вектор движения и параметр, относящийся к нему, могут быть заданы в уровне видеокодека, наборе параметров последовательности или подобном.

5 К тому же, способ кодирования, такой как предсказание вектора движения, усовершенствованное предсказание вектора движения, слияние информации движения, пропуск слияния информации движения или подобный, может быть выполнен с использованием информации движения, хранящейся в памяти, то есть, информации движения опорной картинки.

10 Фиг.15 является схемой последовательности операций, показывающей способ декодирования изображения согласно примерному варианту осуществления настоящего изобретения. Ссылаясь на фиг.15, способ декодирования изображения включает в себя этап усечения (S1510), этап сохранения (S1520), этап получения (S1530) и этап декодирования (S1540).

15 Этап усечения (S1510) и этап сохранения (S1520) согласно фиг.15 аналогичны этапу усечения (S1410) и этапу сохранения (S1420) согласно фиг.14, использующим "I. Процесс усечения вектора движения" и "II. Процесс сохранения вектора движения", описанные выше. К тому же, этап получения (S1530) согласно фиг.15 использует "III. Процесс получения вектора движения", описанный выше, и является симметричным этапу кодирования (S1430) согласно фиг.14. Вследствие этого их подробное описание будет опущено.

20 Устройство декодирования изображения выполняет декодирование с внешним предсказанием с использованием вектора движения древовидного блока кодирования (S1540). Устройство декодирования изображения может сохранять вектор движения в 25 памяти с использованием по меньшей мере одного из способа ограничения динамического диапазона вектора движения, способа уменьшения пространственного разрешения вектора движения, способа квантования вектора движения и способа уменьшения разрешения представления вектора движения, и использовать сохраненный вектор движения для предсказания вектора движения древовидного блока кодирования 30 и слияния их информации движения.

К тому же устройство декодирования изображения может выполнять способ декодирования, такой как предсказание вектора движения, усовершенствованное предсказание вектора движения, слияние информации движения, пропуск слияния информации движения, или подобный, с использованием информации движения, 35 сохраненной в памяти, то есть, информации движения опорной картинки.

Хотя вышеупомянутые примерные варианты осуществления были описаны посредством схем последовательностей операций, представленных сериями этапов или блоков, настоящее изобретение не ограничено последовательностью этапов, описанных 40 выше. То есть, некоторые этапы могут быть осуществлены в другой последовательности, чем другие этапы или одновременно с ними. К тому же, может быть понятно специалистам в данной области техники, к которой принадлежит настоящее изобретение, что этапы, показанные в схемах последовательностей операций являются неисключительными, так что другие этапы могут быть включены или некоторые этапы могут быть удалены.

45 К тому же, вышеупомянутые примерные варианты осуществления включают в себя примеры различных аспектов. Хотя все возможные комбинации для показа различных аспектов не описаны, специалистам в данной области техники понятно, что могут быть реализованы другие комбинации. Вследствие этого, настоящее изобретение должно

толковаться как включающее в себя все другие замены, изменения и модификации, принадлежащие к следующей ниже формуле изобретения.

Формула изобретения

- 5 1. Устройство декодирования изображения, содержащее:
буфер опорной картинки для хранения опорной картинки; и
блок компенсации движения для генерирования блока предсказания с использованием
опорной картинки и вектора движения опорной картинки,
при этом вектор движения опорной картинки усекается в предварительно
10 определенном фиксированном диапазоне, и при этом устройство декодирования
изображения декодирует битовый поток в режиме внутреннего предсказания или в
режиме внешнего предсказания для вывода реконструированного изображения.
2. Устройство декодирования изображения по п. 1, в котором вектор движения
усекается в предварительно определенном диапазоне с фиксированными значениями.
- 15 3. Устройство декодирования изображения по п. 1,
в котором вектор движения опорной картинки сохраняется в предварительно
определенной единице блока, и
в котором блок компенсации движения генерирует блок предсказания с
использованием вектора движения опорной картинки, сохраненного в предварительно
20 определенной единице блока.
4. Устройство декодирования изображения по п. 2, в котором X- и Y-составляющие
вектора движения усекаются в одном и том же диапазоне с фиксированными значениями.
5. Устройство декодирования изображения по п. 1, в котором вектор движения
является вектором движения блока, декодированного в режиме внешнего предсказания.

25

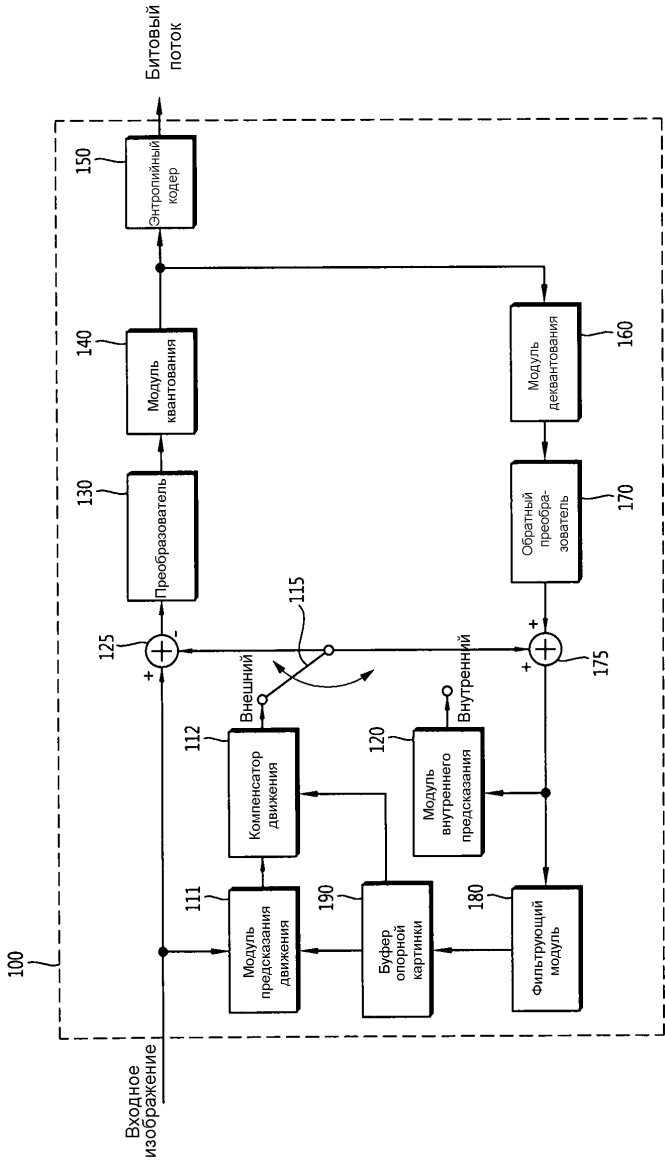
30

35

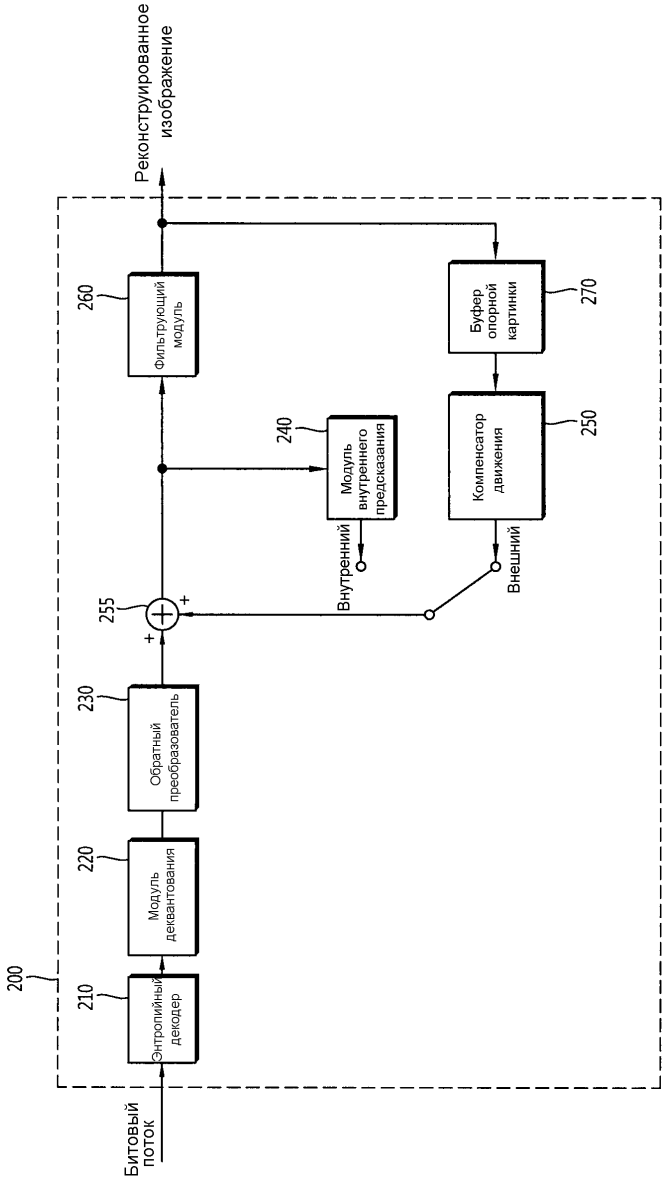
40

45

Фиг.1

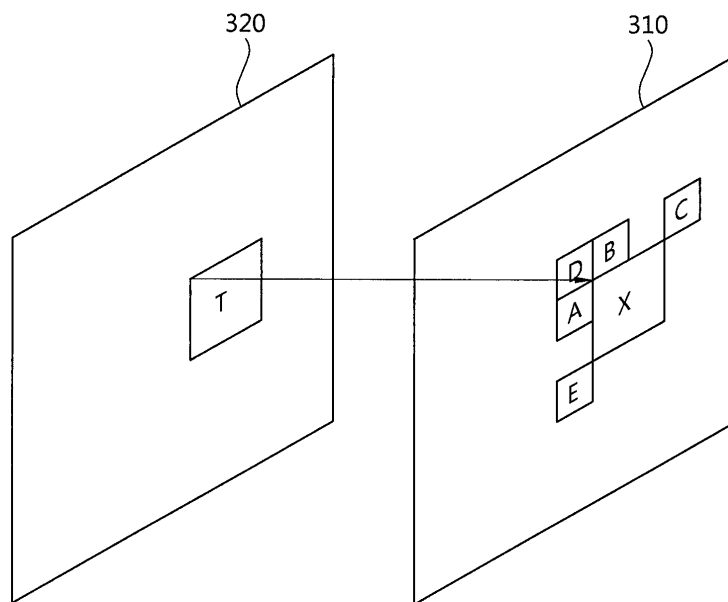


Фиг.2



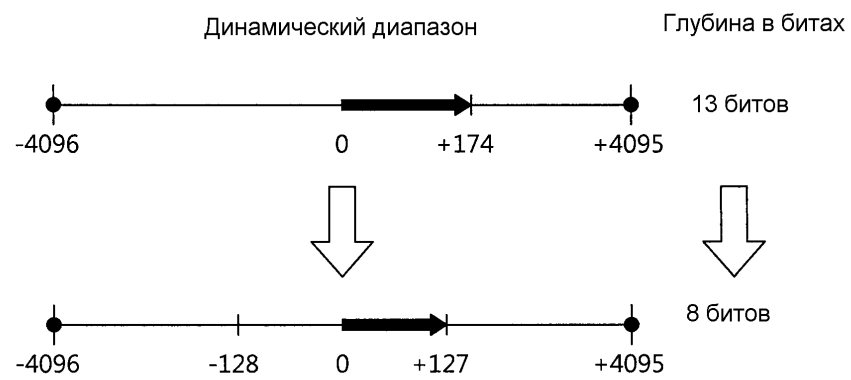
3/15

Фиг.3



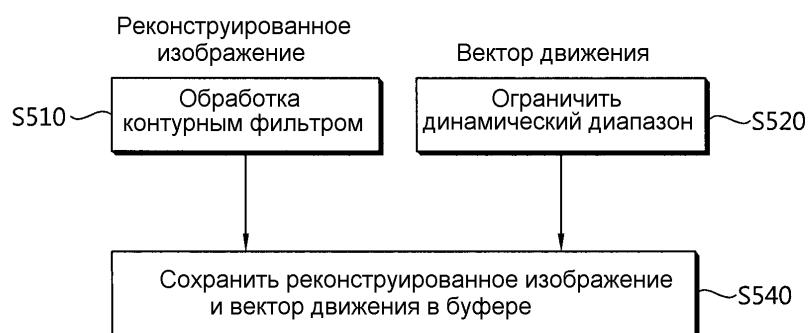
4/15

Фиг.4



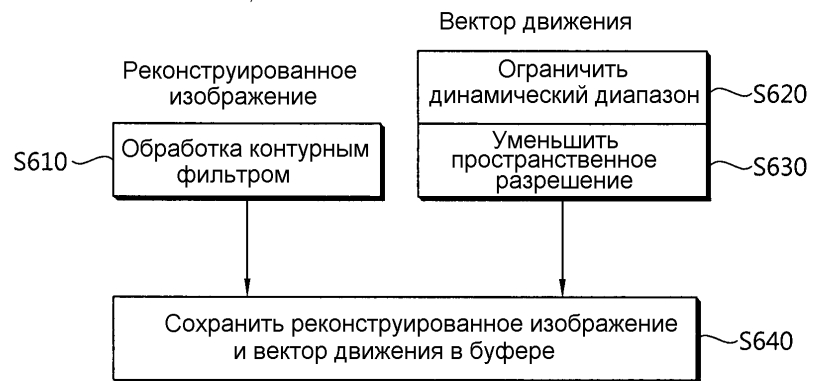
5/15

Фиг.5



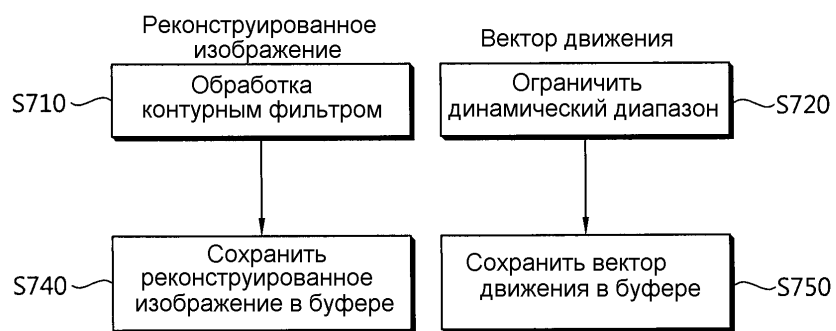
6/15

Фиг.6

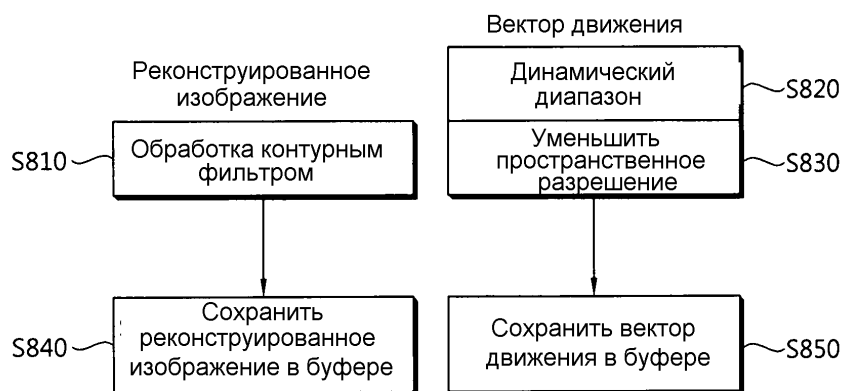


7/15

Фиг.7

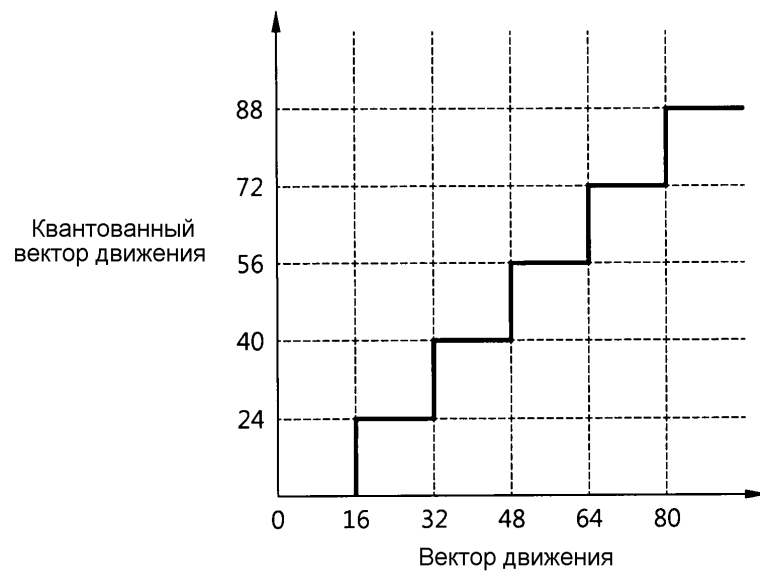


Фиг.8



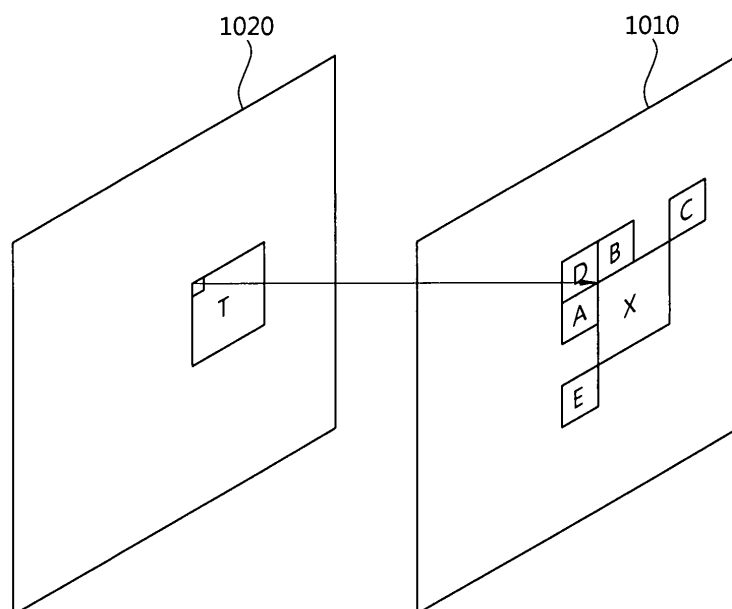
9/15

Фиг.9



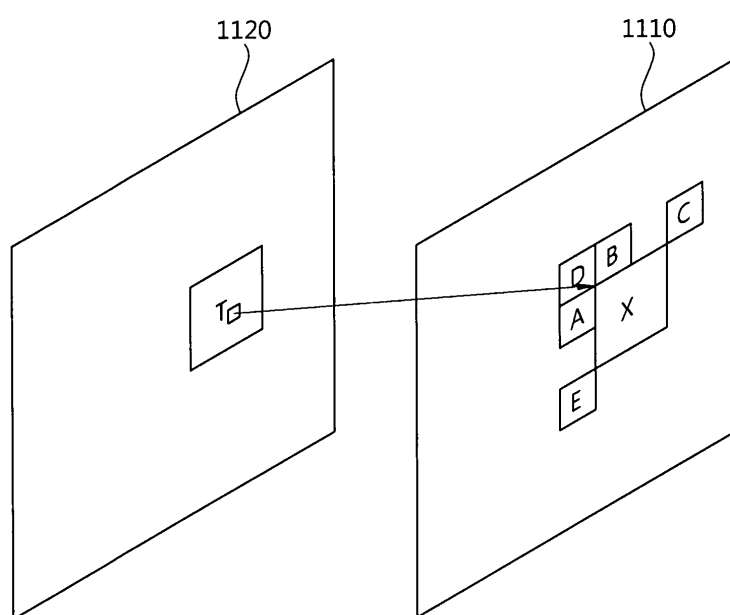
10/15

Фиг.10



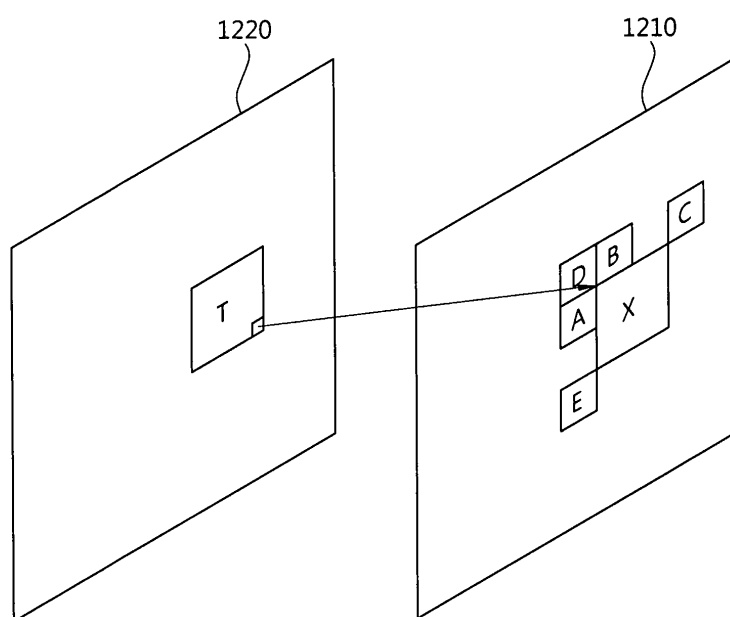
11/15

Фиг.11



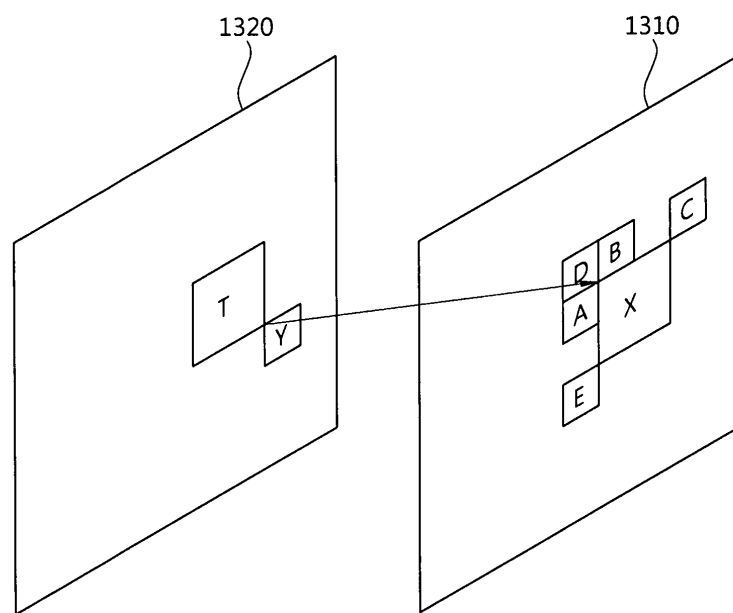
12/15

Фиг.12



13/15

Фиг.13



14/15

Фиг.14



15/15

Фиг.15

