



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

H04N 19/597 (2022.01); H04N 19/70 (2022.01); H04N 19/85 (2022.01); G06T 7/50 (2022.01)

(21)(22) Заявка: 2021102866, 10.07.2019

(24) Дата начала отсчета срока действия патента:
10.07.2019Дата регистрации:
21.03.2022

Приоритет(ы):

(30) Конвенционный приоритет:
11.07.2018 EP 18305930.2;
16.01.2019 EP 19305049.9

(45) Опубликовано: 21.03.2022 Бюл. № 9

(85) Дата начала рассмотрения заявки РСТ на
национальной фазе: 11.02.2021(86) Заявка РСТ:
US 2019/041145 (10.07.2019)(87) Публикация заявки РСТ:
WO 2020/014319 (16.01.2020)

Адрес для переписки:

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

(72) Автор(ы):

РИКАР, Жюльен (FR),
ГЕД, Селин (FR),
ОЛИВЬЕ, Янник (FR)

(73) Патентообладатель(и):

ИНТЕРДИДЖИТАЛ ВиСи ХОЛДИНГЗ,
ИНК. (US)(56) Список документов, цитированных в отчете
о поиске: US 2018268570 A1 - 2018.09.20. XU
YILING et al., Introduction to Point Cloud
Compression, published 2018-08-24. US
2017347120 A1 - 2017.11.30. RU 2431938 C2 -
2011.10.20. RICARDO L. DE QUEIROZ et al.,
Compression of 3D Point Clouds Using a Region-
Adaptive Hierarchical Transform, IEEE
Transactions on Image Processing, vol.25, Issue
8, Aug. 2016.

(54) ОБРАБОТКА ОБЛАКА ТОЧЕК

(57) Реферат:

Изобретение относится к области обработки данных, в частности, обработки облака точек путем удаления незначимых точек облака точек. Технический результат заключается в повышении эффективности кодирования или декодирования видео. Указанный результат достигается тем, что получают 2-D фрагмент облака точек посредством проецирования 3-D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей. Фрагмент разделяют

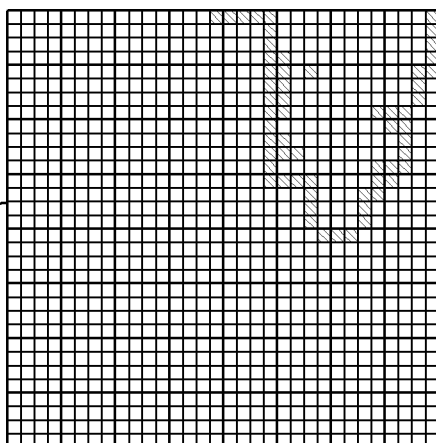
на множество меньших блоков. Определяют количество пикселей в каждом из множества меньших блоков. Получают обновленную карту заполненности на основе определенного количества пикселей в каждом из множества меньших блоков. Получают восстановленное облако точек на основе обновленной карты заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек. 3 н. и 12 з.п. ф-лы, 10 ил.

8000a

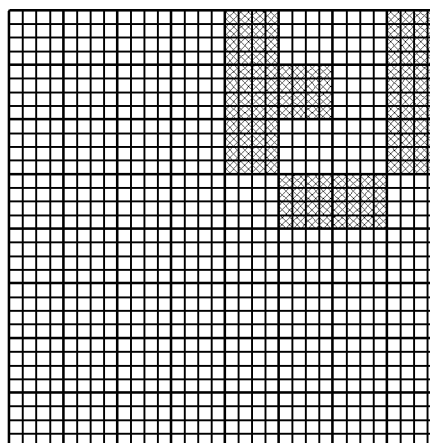
ФРАГМЕНТ ИСХОДНОГО ОБЛАКА
ТОЧЕК

ФРАГМЕНТ ВОССТАНОВЛЕННОГО
ОБЛАКА ТОЧЕК

8100a



8400a



8200a

0	0	0	0	1	0	0	1
0	0	0	0	1	1	0	1
0	0	0	0	1	0	0	1
0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

КАРТА ЗАПОЛНЕННОСТИ ПРИ
РАЗРЕШЕНИИ 4x4

8300a

0	1
0	0

ИНДЕКС БЛОКА К ФРАГМЕНТУ
ПРИ РАЗРЕШЕНИИ 16x16

ФИГ. 8a



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.
H04N 19/597 (2014.01)
H04N 19/70 (2014.01)
H04N 19/85 (2014.01)

(12) **ABSTRACT OF INVENTION**

(52) CPC
H04N 19/597 (2022.01); H04N 19/70 (2022.01); H04N 19/85 (2022.01); G06T 7/50 (2022.01)

(21)(22) Application: **2021102866, 10.07.2019**

(24) Effective date for property rights:
10.07.2019

Registration date:
21.03.2022

Priority:

(30) Convention priority:
11.07.2018 EP 18305930.2;
16.01.2019 EP 19305049.9

(45) Date of publication: **21.03.2022** Bull. № 9

(85) Commencement of national phase: **11.02.2021**

(86) PCT application:
US 2019/041145 (10.07.2019)

(87) PCT publication:
WO 2020/014319 (16.01.2020)

Mail address:
129090, Moskva, ul. B.Spasskaya, 25, stroenie 3,
OOO "Yuridicheskaya firma Gorodisskij i
Partnery"

(72) Inventor(s):

RICARD, Julien (FR),
GUEDE, Celine (FR),
OLIVIER, Yannick (FR)

(73) Proprietor(s):

INTERDIGITAL VC HOLDINGS, INC. (US)

(54) **POINT CLOUD PROCESSING**

(57) Abstract:

FIELD: physics.

SUBSTANCE: invention relates to data processing, in particular, to processing a point cloud by removing insignificant points of the point cloud. Said result is achieved by obtaining 2D fragment of a point cloud by projecting 3D points of the point cloud onto a projection plane, wherein the fragment has a plurality of pixels. Fragment is divided into multiple smaller units. Determining the number of pixels in each of the

plurality of smaller blocks. Updated occupancy map is obtained based on a certain number of pixels in each of the plurality of smaller blocks. Reconstructed point cloud is obtained based on updated occupancy map, wherein the reconstructed point cloud is a less dense representation of the point cloud.

EFFECT: high efficiency of encoding or decoding video.

15 cl, 10 dwg

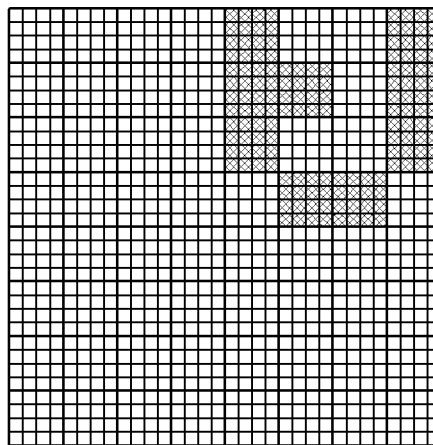
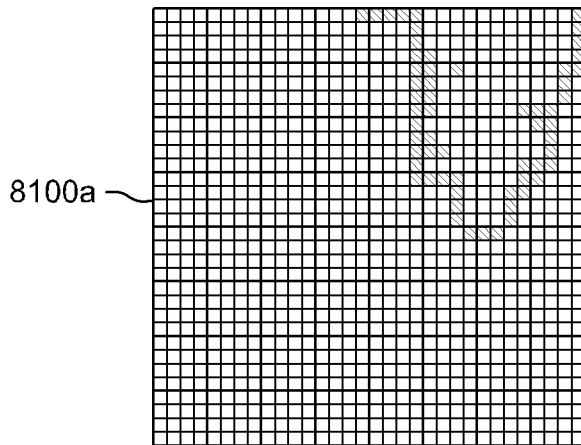
RU 2 767 775 C1

RU 2 767 775 C1

8000a

ФРАГМЕНТ ИСХОДНОГО ОБЛАКА
ТОЧЕК

ФРАГМЕНТ ВОССТАНОВЛЕННОГО
ОБЛАКА ТОЧЕК



8200a

0	0	0	0	1	0	0	1
0	0	0	0	1	1	0	1
0	0	0	0	1	0	0	1
0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

8300a

0	1
0	0

КАРТА ЗАПОЛНЕННОСТИ ПРИ
РАЗРЕШЕНИИ 4×4

ИНДЕКС БЛОКА К ФРАГМЕНТУ
ПРИ РАЗРЕШЕНИИ 16×16

ФИГ. 8а

Область техники

По меньшей мере один из настоящих вариантов относится, в общем, к обработке облака точек и, в частности, к способу и устройству для эффективной обработки облака точек путем удаления незначимых точек облака точек.

5 ПРЕдшествующий Уровень техники

Настоящий раздел предназначен, чтобы представить читателю различные аспекты уровня техники, которые могут быть связаны с различными аспектами по меньшей мере одного из настоящих вариантов осуществления, которые описаны и/или заявлены ниже. Это обсуждение представляется полезным для обеспечения читателя информацией
10 предшествующего уровня техники для облегчения лучшего понимания различных аспектов по меньшей мере одного варианта осуществления.

Облака точек могут использоваться для различных целей, например, в случае культурного наследия/зданий, в которых объекты, такие как изваяния или сооружения, сканируются в 3D, чтобы совместно использовать пространственную конфигурацию
15 объекта без отправки или посещения его. Кроме того, существует способ обеспечения сохранения знаний об объекте в случае, когда он может быть разрушен; например, как храм, разрушенный из-за землетрясения. Такие облака точек обычно являются статическими, цветными и очень большими.

Другим случаем использования является топография и картография, в которых
20 использование 3D представлений обеспечивает получение карт, которые не ограничиваются планом, а могут включать в себя рельеф. Google-карты теперь являются хорошим примером 3D карт, но использует сетки вместо облаков точек. Тем не менее, облака точек могут быть подходящим форматом данных для трехмерных карт, и такие облака точек обычно являются статическими, цветными и очень большими.

Автомобильная промышленность и автономный автомобиль (без водителя) также
25 представляют области, в которых могут использоваться облака точек. Автономные автомобили должны иметь возможность “зондировать” их окружающую среду, чтобы принимать правильные решения по вождению на основе реальности их непосредственных соседей. Типовые датчики, такие как лидары (оптические системы
30 обнаружения и определения дальности), формируют динамические облака точек, которые используются средством принятия решения. Эти облака точек не предназначены для просмотра человеком, и они обычно являются небольшими, не обязательно цветными и динамическими с высокой частотой захвата. Эти облака точек могут иметь другие атрибуты, такие как отражательная способность, обеспечиваемая лидаром,
35 поскольку этот атрибут обеспечивает надежную информацию о материале зондируемого объекта и может помочь в принятии решений.

Виртуальная реальность и иммерсивные миры стали в последнее время важными темами и прогнозируются многими как будущее 2D плоского видео. Основная идея
40 заключается в погружении зрителя в среду, которая окружает зрителя, в отличие от стандартного телевизора, в котором зритель может только смотреть на виртуальный мир перед зрителем. Существует несколько градаций в иммерсивности (степени погружения) в зависимости от свободы зрителя в окружающей среде. Облако точек является хорошим кандидатом формата для распространения миров виртуальной реальности (VR).

Во многих применениях важно иметь возможность распространять динамические
45 облака точек к конечному пользователю (или хранить их в сервере), потребляя только разумную величину битовой скорости (или пространства хранения для приложений хранения), при сохранении приемлемого (или предпочтительно очень хорошего) качества

восприятия. Эффективное сжатие этих динамических облаков точек является ключевым моментом для того, чтобы сделать практичной цепочку распространения многих иммерсивных миров.

По меньшей мере, один вариант осуществления был разработан с учетом
5 вышеизложенного.

Краткое описание Сущности изобретения

Ниже представлено упрощенное краткое описание по меньшей мере одного из
10 настоящих вариантов осуществления для обеспечения базового понимания некоторых аспектов настоящего раскрытия. Это краткое описание не является исчерпывающим обзором варианта осуществления. Оно не предназначено для идентификации ключевых или критических элементов варианта осуществления. Последующее краткое описание сущности изобретения просто представляет некоторые аспекты по меньшей мере одного из настоящих вариантов осуществления в упрощенной форме в качестве вводной части к более подробному описанию, предоставленному далее в настоящем документе.

15 В соответствии с общим аспектом по меньшей мере одного варианта осуществления, обеспечен способ сокращения облака точек, представляющего изображение, содержащий: получение 2-D фрагмента (патча) облака точек путем проецирования 3-D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей; разделение фрагмента на множество меньших блоков; определение количества пикселей
20 в каждом из множества меньших блоков; получение обновленной карты заполненности (занятости) на основе определенного количества пикселей в каждом из множества меньших блоков; и получение восстановленного облака точек на основе обновленной карты заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек.

25 В соответствии с другим общим аспектом по меньшей мере одного варианта осуществления, обеспечено устройство для сокращения облака точек, представляющего изображение, содержащее: средство для получения 2-D фрагмента облака точек посредством проецирования 3-D точек облака точек на плоскость проекции, причем
30 фрагмент имеет множество пикселей; средство для разделения фрагмента на множество меньших блоков; средство для определения количества пикселей в каждом из множества меньших блоков; средство для получения обновленной карты заполненности на основе определенного количества пикселей в каждом из множества меньших блоков; и средство для получения восстановленного облака точек на основе обновленной карты
35 заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек.

В соответствии с другим общим аспектом по меньшей мере одного варианта осуществления, обеспечено устройство для сокращения облака точек, представляющего изображение, содержащее один или более процессоров, причем один или более процессоров выполнены с возможностью: получать 2-D фрагмент облака точек
40 посредством проецирования 3-D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей; разделять фрагмент на множество меньших блоков; определять количество пикселей в каждом из множества меньших блоков; получать обновленную карту заполненности на основе определенного количества пикселей в каждом из множества меньших блоков; и получать восстановленное облако
45 точек на основе обновленной карты заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек.

В соответствии с другим вариантом осуществления, обеспечен битовый поток, содержащий восстановленное облако точек, при этом битовый поток формируется

посредством: получения 2-D фрагмента облака точек посредством проецирования 3-D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей; разделение фрагмента на множество меньших блоков; определение количества пикселей в каждом из множества меньших блоков; получение обновленной карты заполненности на основе определенного количества пикселей в каждом из множества меньших блоков; и получение восстановленного облака точек на основе обновленной карты заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек.

Один или более из по меньшей мере одного варианта осуществления также обеспечивают устройство, компьютерный программный продукт, долговременный машиночитаемый носитель и сигнал.

Конкретный характер по меньшей мере одного из настоящих вариантов осуществления, а также других объектов, преимуществ, признаков и применений по меньшей мере одного из настоящих вариантов осуществления станет очевидным из последующего описания примеров, рассматриваемых совместно с прилагаемыми чертежами.

Краткое Описание Чертежей

На чертежах проиллюстрированы примеры нескольких вариантов осуществления. На чертежах показано следующее:

Фиг. 1 иллюстрирует блок-схему примера двухуровневой структуры кодирования облака точек в соответствии с по меньшей мере одним из настоящих вариантов осуществления;

Фиг. 2 иллюстрирует блок-схему примера двухуровневой структуры декодирования облака точек в соответствии с по меньшей мере одним из настоящих вариантов осуществления;

Фиг. 3 иллюстрирует блок-схему примера кодера облака точек на основе изображения в соответствии с по меньшей мере одним из настоящих вариантов осуществления;

Фиг. 4 иллюстрирует блок-схему примера декодера облака точек на основе изображения в соответствии с по меньшей мере одним из настоящих вариантов осуществления;

Фиг. 5 иллюстрирует пример холста, содержащего 2 фрагмента и их 2D ограничивающие прямоугольники;

Фиг. 6 иллюстрирует пример двух промежуточных выборок, расположенных между двумя 3D выборками вдоль линии проекции;

Фиг. 7 схематично иллюстрирует пример синтаксиса битового потока, представляющего базовый уровень BL в соответствии с по меньшей мере одним из настоящих вариантов осуществления;

Фиг. 8 и фиг. 8А иллюстрируют два варианта осуществления того, как исходный фрагмент облака точек обрабатывается или восстанавливается с помощью карты заполненности и индекса блока к фрагменту; и

Фиг. 9 иллюстрирует блок-схему примера системы, в которой реализованы различные аспекты и варианты осуществления.

Подробное Описание

По меньшей мере один из настоящих вариантов осуществления описан более подробно ниже со ссылкой на прилагаемые чертежи, на которых показаны примеры по меньшей мере одного из настоящих вариантов осуществления. Однако вариант осуществления может быть воплощен во многих альтернативных формах и не должен рассматриваться как ограниченный примерами, изложенными в настоящем документе.

Соответственно, должно быть понятно, что не существует намерения ограничивать варианты осуществления конкретными раскрытыми формами. Напротив, раскрытие предназначено для охвата всех модификаций, эквивалентов и альтернатив, попадающих в сущность и объем этой заявки.

5 Когда на чертеже представлена блок-схема последовательности операций, следует понимать, что она также предоставляет блок-схему соответствующего устройства. Аналогично, когда на чертеже представлена блок-схема, следует понимать, что она также предоставляет блок-схему соответствующего способа/процесса.

10 Одинаковые или подобные элементы на чертежах обозначены одинаковыми ссылочными позициями.

Аспекты, описанные и рассматриваемые ниже, могут быть реализованы во многих различных формах. На фиг. 1-9 представлены некоторые варианты осуществления, но возможны другие варианты осуществления, и обсуждение, представленное на фиг. 1-9, не ограничивает широту вариантов осуществления.

15 По меньшей мере один из аспектов, в общем, относится к кодированию и декодированию облака точек, и по меньшей мере один другой аспект, в общем, относится к передаче сгенерированного или закодированного битового потока.

20 Более точно, различные способы и другие аспекты, описанные здесь, могут использоваться для модификации модулей, например основанных на изображениях кодера 3000 и декодера 4000, как показано на фиг. 3 и фиг. 4 соответственно.

Кроме того, настоящие аспекты не ограничиваются стандартами MPEG, такими как MPEG-1 часть 5, который относится к сжатию облака точек, и могут быть применены, например, к другим стандартам и рекомендациям, уже существующим или будущим, и расширениям любых таких стандартов и рекомендаций (включая MPEG-1 часть 5).

25 Если не указано иначе или технически не исключено, аспекты, описанные в настоящей заявке, могут быть использованы по отдельности или в комбинации.

В дальнейшем, данные изображения относятся к данным, например, к одному или нескольким массивам 2D выборок в конкретном формате изображения/видео.

30 Конкретный формат изображения/видео может специфицировать информацию, относящуюся к пиксельным значениям изображения (или видео). Конкретный формат изображения/видео может также специфицировать информацию, которая может использоваться дисплеем и/или любым другим устройством, например, для визуализации и/или декодирования изображения (или видео). Изображение обычно включает в себя первый компонент в форме первого 2D массива выборок, обычно представляющего сигнал яркости (или яркость) изображения. Изображение может также включать в себя 35 второй компонент и третий компонент в форме других 2D массивов выборок, обычно представляющих сигнал цветности (или цветность) изображения. Некоторые варианты осуществления представляют одну и ту же информацию с использованием набора 2D массивов цветовых выборок, таких как традиционное трехцветное RGB представление.

40 Пиксельное значение представлено в одном или нескольких вариантах осуществления вектором C значений, где C - количество компонентов. Каждое значение вектора обычно представляется количеством битов, которые могут определять динамический диапазон пиксельных значений.

Блок изображения означает набор пикселей, принадлежащих к изображению.

45 Пиксельные значения блока изображения (или данные блока изображения) относятся к значениям пикселей, которые относятся к этому блоку изображения. Блок изображения может иметь произвольную форму, хотя обычно это прямоугольники.

Облако точек может быть представлено набором 3D выборок в 3D объеме

пространстве, которые имеют уникальные координаты и которые также могут иметь один или более атрибутов.

3D выборка этого набора данных может быть определена ее пространственным положением (координатами X , Y и Z в 3D пространстве) и, возможно, одним или несколькими ассоциированными атрибутами, такими как цвет, представленный в цветовом пространстве RGB или YUV, например прозрачность, отражательная способность, двухкомпонентный нормальный вектор или любой признак, представляющий признак этой выборки. Например, 3D выборка может быть определена 6 компонентами (X , Y , Z , R , G , B) или эквивалентно (X , Y , Z , u , U , V), где (X , Y , Z) определяет координаты точки в 3D пространстве, и (R , G , B) или (u , U , V) определяет цвет этой 3D выборки. Один и тот же тип атрибута может присутствовать несколько раз. Например, множество цветовых атрибутов могут предоставлять цветовую информацию с различных точек зрения.

Облако точек может быть статическим или динамическим в зависимости от того, изменяется или нет облако в зависимости от времени. Статическое облако точек или экземпляр динамического облака точек обычно называют кадром облака точек. Следует отметить, что в случае динамического облака точек, количество точек, как правило, не является постоянным, а наоборот, обычно изменяется во времени. Более обобщенно, облако точек может рассматриваться как динамическое, если что-либо изменяется во времени, например, количество точек, положение одной или нескольких точек или любой атрибут любой точки.

В качестве примера, 2D выборка может быть определена 6 компонентами (u , v , Z , R , G , B) или эквивалентно (u , v , Z , u , U , V). (u , v) определяет координаты 2D выборки в 2D пространстве плоскости проекции. Z является значением глубины проецируемой 3D выборки на эту плоскость проекции. (R , G , B) или (u , U , V) определяет цвет этой 3D выборки.

Фиг. 1 иллюстрирует блок-схему примера двухуровневой структуры 1000 кодирования облака точек в соответствии с по меньшей мере одним из настоящих вариантов осуществления.

Двухуровневая структура 1000 кодирования облака точек может обеспечивать битовый поток B , представляющий входной кадр IPCF облака точек. Возможно, входной кадр IPCF облака точек представляет кадр динамического облака точек. Тогда кадр динамического облака точек может кодироваться двухуровневой структурой 1000 кодирования облака точек независимо от другого кадра.

В основном, двухуровневая структура 1000 кодирования облака точек может обеспечивать возможность структурировать битовый поток B как базовый уровень BL и расширенный уровень EL . Базовый уровень BL может обеспечивать представление с потерями входного кадра IPCF облака точек, и расширенный уровень EL может обеспечить представление более высокого качества (возможно без потерь) путем кодирования изолированных точек, не представленных базовым уровнем BL .

Базовый уровень BL может быть обеспечен основанным на изображении кодером 3000, как показано на фиг. 3. Основанный на изображении кодер 3000 может предоставлять геометрические/текстурные изображения, представляющие геометрию/атрибуты 3D выборок входного кадра IPCF облака точек. Это может позволить отбрасывать изолированные трехмерные выборки. Базовый уровень BL может быть декодирован основанным на изображении декодером 4000, как показано на фиг. 4, который может обеспечить промежуточный восстановленный кадр IRPCF облака точек.

Затем, возвращаясь к двухуровневому кодированию 1000 облака точек согласно

фиг. 1, компаратор COMP может сравнивать 3D выборки входного кадра IPCF облака точек с 3D выборками промежуточного восстановленного кадра IRPCF облака точек для того, чтобы обнаружить/локализовать пропущенные/изолированные 3D выборки. Затем кодер ENC может кодировать пропущенные 3D выборки и может обеспечивать расширенный уровень EL. Наконец, базовый уровень BL и расширенный уровень EL могут быть мультиплексированы вместе мультиплексором MUX так, чтобы сгенерировать битовый поток В.

В соответствии с одним вариантом осуществления, кодер ENC может содержать детектор, который может обнаруживать и ассоциировать 3D опорную выборку R промежуточного восстановленного кадра IRPCF облака точек с пропущенными 3D выборками M.

Например, 3D опорная выборка R, ассоциированная с пропущенной 3D выборкой M, может быть ближайшим соседом M в соответствии с заданной метрикой.

В соответствии с вариантом осуществления, кодер ENC может затем кодировать пространственные местоположения пропущенных 3D выборок M и их атрибуты как разности, определенные в соответствии с пространственными местоположениями и атрибутами 3D опорных выборок R.

В одном варианте, эти разности могут кодироваться отдельно.

Например, для пропущенной 3D выборки M с пространственными координатами x (M), y (M) и z (M), разность Dx (M) положения x -координаты, разность Dy (M) положения y -координаты, разность Dz (M) положения z -координаты, разность Dr (M) компонента R-атрибута, разность Dg (M) компонента G-атрибута и разность Db (M) компонента B-атрибута могут быть вычислены следующим образом:

$$Dx(M)=x(M)-x(R),$$

где x (M) представляет собой x -координату 3D выборки M, соответственно R в геометрическом изображении, представленном на фиг. 3,

$$Dy(M)=y(M)-y(R)$$

где y (M) представляет собой y -координату 3D выборки M, соответственно R в геометрическом изображении, представленном на фиг. 3,

$$Dz(M)=z(M)-z(R)$$

где z (M) представляет собой z -координату 3D выборки M, соответственно R в геометрическом изображении, представленном на фиг. 3,

$$Dr(M)=R(M)-R(R)$$

где R (M), соответственно R (R) представляет собой r -цветовой компонент атрибута цвета 3D выборки M, соответственно R,

$$Dg(M)=G(M)-G(R)$$

где G (M), соответственно G (R) представляет собой g -цветовой компонент атрибута цвета 3D выборки M, соответственно R,

$$Db(M)=B(M)-B(R)$$

где B (M), соответственно B (R) представляет собой b -цветовой компонент атрибута цвета 3D выборки M, соответственно R.

Фиг. 2 иллюстрирует блок-схему примера двухуровневой структуры 2000 декодирования облака точек в соответствии с по меньшей мере одним из настоящих вариантов осуществления.

Поведение двухуровневой структуры 2000 декодирования облака точек зависит от ее возможностей.

Двухуровневая структура 2000 декодирования облака точек с ограниченными возможностями может осуществлять доступ только к базовому уровню BL из битового

потока В с использованием демультимплексора DMUX, а затем может предоставлять верную (но с потерями) версию IRPCF входного кадра IPCF облака точек путем декодирования базового уровня BL с помощью декодера 4000 Облака точек, как показано на фиг. 4.

5 Двухуровневая структура 2000 декодирования облака точек с полными возможностями может осуществлять доступ как к базовому уровню BL, так и к расширенному уровню EL из битового потока В с использованием демультимплексора DMUX. Декодер 4000 облака точек, как показано на фиг. 4, может определять промежуточный восстановленный кадр IRPCF облака точек из основного уровня BL.
10 Декодер DEC может определять комплементарный кадр CPCF облака точек из расширенного уровня EL. Объединитель COM затем может объединять вместе промежуточный восстановленный кадр IRPCF облака точек и комплементарный кадр CPCF облака точек, чтобы, таким образом, обеспечить более высококачественное (возможно без потерь) представление (восстановление) CRPCF входного кадра IPCF
15 облака точек.

Фиг. 3 иллюстрирует блок-схему примера основанного на изображении кодера 3000 облака точек в соответствии с по меньшей мере одним из настоящих вариантов осуществления.

Основанный на изображении кодер 3000 облака точек использует существующие
20 кодеки видео для сжатия информации о геометрии и текстуре (атрибуте) динамического облака точек. Это достигается по существу путем преобразования данных облака точек в набор различных последовательностей видео.

В конкретных вариантах осуществления, два видео, одно для захвата геометрической информации данных облака точек и другое для захвата текстурной информации, могут
25 быть сгенерированы и сжаты с использованием существующих кодеков видео. Примером существующего кодека видео является кодер/декодер главного профиля HEVC (ITU-T H.265 Telecommunication standardization sector of ITU (02/2018), series H: audiovisual and multimedia systems, infrastructure of audiovisual services - coding of moving video, High efficiency video coding, Recommendation ITU-T H.265).

30 Дополнительные метаданные, которые используются для интерпретации двух видео, обычно также генерируются и сжимаются по отдельности. Такие дополнительные метаданные включают в себя, например, карту заполненности OM и/или вспомогательную информацию PI о фрагменте.

Сгенерированные битовые потоки видео и метаданные могут быть затем
35 мультимплексированы вместе для формирования комбинированного битового потока.

Следует отметить, что метаданные, как правило, представляют малый объем во всей информации. Основной объем информации приходится на битовые потоки видео.

Пример такого процесса кодирования/декодирования облака точек задается алгоритмом Тестовой модели Категории 2 (также обозначается как Сжатие облака на
40 основе видео, сокращенно как V-PCC), который реализует проект стандарта MPEG, как определено в ISO/IEC JTC1/SC29/WG1 1 MPEG2019/w18180 (January 2019, Marrakesh).

На этапе 3100, модуль PGM может генерировать по меньшей мере один фрагмент путем разложения 3D выборки набора данных, представляющего входной кадр IRPCF облака точек, на 2D выборки на плоскости проекции, используя стратегию, которая
45 обеспечивает наилучшее сжатие.

Фрагмент может быть определен как набор 2D выборок.

Например, в V-PCC, нормаль в каждой 3D выборке сначала оценивается, как описано, например, в Hoppe et al. (Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, Werner

Stuetzle. Surface reconstruction from unorganized points. *ACM SIGGRAPH 1992 Proceedings*, 71-78). Далее, начальную кластеризацию входного кадра IPCF облака получают путем ассоциирования каждой 3D выборки с одной из шести ориентированных плоскостей 3D ограничивающего прямоугольника, окружающего 3D выборки входного кадра IPCF облака точек. Более точно, каждая 3D выборка кластеризуется и ассоциируется с ориентированной плоскостью, которая имеет ближайшую нормаль (т.е. максимизирует скалярное произведение нормали точки и нормали плоскости). Затем 3D выборки проецируются на ассоциированные с ними плоскости. Набор 3D выборок, который образует связанную область в своей плоскости, упоминается как связанный компонент. Связанный компонент представляет собой набор из по меньшей мере одной 3D выборки, имеющей одинаковую нормаль и ту же самую ассоциированную ориентированную плоскость. Начальная кластеризация затем уточняется путем итерационного обновления кластера, ассоциированного с каждой 3D выборкой, на основе ее нормали и кластеров ее ближайших соседних выборок. Последний этап состоит в генерации одного фрагмента из каждого связанного компонента, что осуществляется путем проецирования 3D выборок каждого связанного компонента на ориентированную плоскость, ассоциированную со связанным компонентом. Фрагмент ассоциируется с вспомогательной информацией PI фрагмента, которая представляет вспомогательную информацию фрагмента, определенную для каждого фрагмента, для интерпретации спроецированных 2D выборок, которые соответствуют информации о геометрии и/или атрибуте.

В V-PCC, например, вспомогательная информация PI фрагмента включает в себя 1) информацию, указывающую одну из шести ориентированных плоскостей 3D ограничивающего прямоугольника, охватывающего 3D выборки связанного компонента; 2) информацию относительно нормали к плоскости; 3) информацию, определяющую 3D местоположение связанного компонента относительно фрагмента, представленного в терминах глубины, тангенциального сдвига и би-тангенциального сдвига; и 4) информацию, такую как координаты (u_0, v_0, u_1, v_1) в плоскости проекции, определяющей 2D ограничивающий прямоугольник, охватывающий фрагмент.

На этапе 3200, модуль компоновки фрагмента, PPM, может отображать (помещать) по меньшей мере один сгенерированный фрагмент на 2D сетку (также называемую холстом) без какого-либо перекрытия способом, который обычно минимизирует неиспользуемое пространство, и может гарантировать, что каждый блок T×T (например, 16×16) 2D сетки ассоциирован с уникальным фрагментом. Заданный минимальный размер блока T×T 2D сетки может специфицировать минимальное расстояние между отдельными фрагментами, помещенными на эту 2D сетку. Разрешение 2D сетки может зависеть от размера входного облака точек, и его ширина W и высота H и размер T блока могут передаваться как метаданные в декодер.

Вспомогательная информация PI фрагмента может дополнительно включать в себя информацию относительно ассоциации между блоком 2D сетки и фрагментом.

В V-PCC, вспомогательная информация PI может включать в себя информацию индекса блока к фрагменту (BlockToPatch), которая определяет ассоциацию между блоком 2D сетки и индексом фрагмента.

На фиг. 5 показан пример холста C, содержащего 2 фрагмента P1 и P2 и ассоциированные с ними 2D ограничивающие прямоугольники B1 и B2. Следует отметить, что два ограничивающих прямоугольника могут перекрываться в холсте C, как показано на фиг. 5. 2D сетка (разделение холста) представлена только внутри ограничивающего прямоугольника, но разделение холста также происходит за

пределами этих ограничивающих прямоугольников. Ограничивающий прямоугольник, ассоциированный с фрагментом, может быть разделен на блоки $T \times T$, как правило, $T=16$.

Блоки $T \times T$, содержащие 2D выборки, принадлежащие фрагменту, могут рассматриваться как занятые (заполненные) блоки. Каждый заполненный блок холста представлен конкретным пиксельным значением (например 1) в карте заполненности ОМ, и каждый незаполненный блок холста представлен другим конкретным значением, например 0. Тогда пиксельное значение карты заполненности ОМ может указывать, заполнен ли блок $T \times T$ холста, то есть содержит 2D выборки, которые принадлежат фрагменту.

На фиг. 5, заполненный блок представлен белым блоком, а светло-серые блоки представляют незаполненные блоки. Процессы генерации изображения (этапы 3300 и 3400 на фиг. 3) используют отображение по меньшей мере одного сгенерированного фрагмента на 2D сетку, вычисленную на этапе 3200, для сохранения геометрии и текстуры входного кадра IPCF облака точек как изображений.

На этапе 3300, генератор геометрического изображения, GIG, может генерировать по меньшей мере одно геометрическое изображение G1 из входного кадра IPCF облака точек, карты заполненности ОМ и вспомогательной информации PI фрагмента. Генератор GIG геометрического изображения может использовать информацию карты заполненности для того, чтобы обнаружить (локализовать) заполненные блоки и, таким образом, непустые пиксели в геометрическом изображении G1.

Геометрическое изображение G1 может представлять геометрию входного кадра IPCF облака точек и может представлять собой монохроматическое изображение $W \times H$ пикселей, представленных, например, в YUV420-8-битовом формате.

Для того чтобы лучше обрабатывать случай, когда множество 3D выборок проецируется (отображается) на одну и ту же 2D выборку плоскости проекции (вдоль одного и того же направления (линии) проекции), может генерироваться множество изображений, упоминаемых как уровни. Таким образом, различные значения глубины D_1, \dots, D_n могут быть ассоциированы с 2D выборкой фрагмента, и затем может быть сгенерировано множество геометрических изображений.

В V-PCC, 2D выборки фрагмента проецируются на два уровня. Первый уровень, также называемый ближним уровнем, может хранить, например, значения D_0 глубины, ассоциированные с 2D выборками с меньшими глубинами. Вторым уровнем, упоминаемым как дальний уровень, может хранить, например, значения D_1 глубины, ассоциированные с 2D выборками с большими глубинами. Альтернативно, второй уровень может сохранять значения разности между значениями D_1 и D_0 глубины. Например, информация, сохраненная вторым изображением глубины, может находиться в интервале $[0, \Delta]$, соответствующем значениям глубины в диапазоне $[D_0, D_0 + \Delta]$, где Δ является параметром, определяемым пользователем, который описывает толщину поверхности.

Таким образом, второй уровень может содержать существенные контурно-подобные высокочастотные признаки. Таким образом, очевидно, что изображение второй глубины может быть трудным для кодирования с использованием унаследованного кодера видео, и поэтому значения глубины могут плохо восстанавливаться из декодированного изображения второй глубины, что приводит к плохому качеству геометрии восстановленного кадра облака точек.

Согласно варианту осуществления модуль GIG генерации геометрического изображения может кодировать (получать) значения глубины, ассоциированные с 2D

выборками первого и второго уровней, с использованием вспомогательной информации PI фрагмента.

В V-РСС, местоположение 3D выборки во фрагменте с соответствующим связанным компонентом может быть выражено в терминах глубины $\delta(u, v)$, тангенциального сдвига $s(u, v)$ и би-тангенциального сдвига $r(u, v)$ следующим образом:

$$\delta(u, v) = \delta_0 + g(u, v)$$

$$s(u, v) = s_0 - u_0 + u$$

$$r(u, v) = r_0 - v_0 + v$$

где $g(u, v)$ представляет собой компонент яркости геометрического изображения, (u, v) представляет собой пиксел, ассоциированный с 3D выборкой на плоскости проекции, (δ_0, s_0, r_0) представляет собой 3D местоположение соответствующего фрагмента связанного компонента, которому принадлежит 3D выборка, и (u_0, v_0, u_1, v_1) являются координатами в плоскости проекции, определяющей 2D ограничивающий прямоугольник, охватывающий проекцию фрагмента, ассоциированного со связанным компонентом.

Таким образом, модуль GIG генерации геометрического изображения может кодировать (получать) значения глубины, ассоциированные с 2D выборками уровня (первого или второго или обоих) в качестве компонента яркости $g(u, v)$, заданного посредством: $g(u, v) = \delta(u, v) - \delta_0$. Следует отметить, что эта взаимосвязь может использоваться для восстановления местоположений 3D выборки (δ_0, s_0, r_0) из восстановленного геометрического изображения $g(u, v)$ с сопутствующей вспомогательной информацией PI фрагмента.

В соответствии с вариантом осуществления, режим проекции может использоваться для указания того, может ли первое геометрическое изображение GI0 сохранять значения глубины 2D выборок либо первого, либо второго уровня, и второе геометрическое изображение GI1 может сохранять значения глубины, ассоциированные с 2D выборками либо второго, либо первого уровня.

Например, когда режим проекции равен 0, то первое геометрическое изображение GI0 может сохранять значения глубины 2D выборок первого уровня, и второе геометрическое изображение GI1 может сохранять значения глубины, ассоциированные с 2D выборками второго уровня. Взаимно, когда режим проекции равен 1, первое геометрическое изображение GI0 может сохранять значения глубины 2D выборок второго уровня, и второе геометрическое изображение GI1 может сохранять значения глубины, ассоциированные с 2D выборками первого уровня.

В соответствии с вариантом осуществления, режим проекции кадра может использоваться для указания того, используется ли режим фиксированной проекции для всех фрагментов, или используется ли режим переменной проекции, в котором каждый фрагмент может использовать другой режим проекции.

Режим проекции и/или режим проекции кадра могут быть переданы как метаданные. Алгоритм принятия решения о режиме проекции кадра может быть обеспечен, например, в разделе 2.2.1 V-РСС.

В соответствии с вариантом осуществления, когда проекция кадра указывает, что может использоваться режим переменной проекции, может использоваться режим проекции фрагмента для указания соответствующего режима для использования, чтобы (де-) проецировать фрагмент.

Режим проекции фрагмента может быть передан как метаданные и может представлять собой, возможно, информацию, включенную во вспомогательную информацию PI фрагмента.

Алгоритм принятия решения о режиме проекции фрагмента обеспечен, например, в разделе 2.2.1.3.2 V-PCC.

В соответствии с вариантом осуществления этапа 3300, пиксельное значение в первом геометрическом изображении, например, G_{i0} , соответствующее 2D выборке (u, v) фрагмента, может представлять значение глубины по меньшей мере одной промежуточной 3D выборки, определенной вдоль линии проекции, соответствующей 2D выборке (u, v) . Более точно, промежуточные 3D выборки располагаются вдоль линии проекции и совместно используют те же координаты 2D выборки (u, v) , значение D_{i1} глубины которой закодировано во втором геометрическом изображении, например G_{i1} . Кроме того, промежуточные 3D выборки могут иметь значения глубины между значением D_{i0} глубины и значением D_{i1} глубины. Назначенный бит может быть ассоциирован с каждой из промежуточных 3D выборок и установлен в 1, если существует промежуточная 3D выборка, и 0 в противном случае.

Фиг. 6 иллюстрирует пример двух промежуточных 3D выборок P_{i1} и P_{i2} , расположенных между двумя 3D выборками P_0 и P_1 вдоль линии PL проекции. 3D выборки P_0 и P_1 имеют соответственно значения глубины, равные D_0 и D_1 . Значения глубины D_{i1} и D_{i2} соответственно двух промежуточных 3D выборок P_{i1} и P_{i2} больше чем D_0 , и меньше чем D_1 .

Затем все назначенные биты по линии проекции могут быть конкатенированы для формирования кодового слова, обозначаемого далее как кодовое слово расширенной карты заполненности (EOM). Как показано на фиг. 6, в предположении кодового слова EOM длиной 8 битов, 2 бита равны 1 для указания местоположения двух 3D выборок P_{i1} и P_{i2} . Наконец, все кодовые слова EOM могут быть упакованы в изображение, например карту заполненности OM. В этом случае, по меньшей мере один фрагмент холста может содержать по меньшей мере одно кодовое слово EOM. Такой фрагмент обозначен как опорный фрагмент, и блок опорного фрагмента обозначается как опорный блок EOM. Таким образом, пиксельное значение карты заполненности OM может быть равно первому значению, например 0, для указания незаполненного блока холста или другому значению, например больше 0, для указания либо заполненного блока холста, например, когда $D_1 - D_0 \leq 1$, либо для указания опорного блока EOM холста, когда, например, $D_1 - D_0 > 1$.

Местоположения пикселей в карте заполненности OM, которая указывает опорные блоки EOM, и значения битов кодового слова EOM, которые получаются из значений этих пикселей, указывают 3D координаты промежуточных 3D выборок.

На этапе 3400, генератор TIG текстурного изображения может генерировать по меньшей мере одно текстурное изображение TIG из входного кадра IPCF облака точек, карты заполненности OM, вспомогательной информации PI фрагмента и геометрии восстановленного кадра облака точек, полученного из по меньшей мере одного декодированного геометрического изображения DGI, выведенного из декодера видео VDEC (этап 4200 на фиг. 4).

Текстурное изображение TI может представлять текстуру входного кадра IPCF облака точек и может быть изображением $W \times H$ пикселей, представленным, например, в YUV420-8-битовом формате.

Генератор TG текстурного изображения может использовать информацию карты заполненности для того, чтобы обнаруживать (локализовать) заполненные блоки и, таким образом, непустые пиксели в текстурном изображении.

Генератор TIG текстурного изображения может быть приспособлен, чтобы

генерировать и ассоциировать текстурное изображение T1 с каждым геометрическим изображением/уровнем DGI.

В соответствии с вариантом осуществления, генератор TIG текстурного изображения может кодировать (сохранять) значения T0 текстуры (атрибута), ассоциированные с 2D выборками первого уровня, как пиксельные значения первого текстурного изображения T10 и значения T1 текстуры, ассоциированные с 2D выборками второго уровня, как пиксельные значения второго текстурного изображения T11.

Альтернативно, модуль TIG генерации текстурного изображения может кодировать (сохранять) значения T1 текстуры, ассоциированные с 2D выборками второго уровня, как пиксельные значения первого текстурного изображения T10 и значения D0 текстуры, ассоциированные с 2D выборками первого уровня, как пиксельные значения второго геометрического изображения G11.

Например, цвета 3D выборок могут быть получены, как описано в разделе 2.2.3, 2.2.4, 2.2.5, 2.2.8 или 2.5 V-PCC.

Значения текстуры двух 3D выборок хранятся либо в первом, либо во втором текстурных изображениях. Но значения текстуры промежуточных 3D выборок не могут быть сохранены ни в этом первом текстурном изображении T10, ни во втором текстурном изображении T11, потому что местоположения проецируемых промежуточных 3D выборок соответствуют заполненным блокам, которые уже используются для хранения значения текстуры другой 3D выборки (P0 или P1), как показано на фиг. 6. Таким образом, значения текстуры промежуточных 3D выборок сохраняются в текстурных блоках EOM, расположенных где-то либо в первом, либо во втором текстурном изображении в местоположениях, определенных согласно процедуре (раздел 9.4.5 V-PCC). Вкратце, этот процесс определяет местоположения незаполненных блоков в текстурном изображении и сохраняет значения текстуры, ассоциированные с промежуточными 3D выборками в качестве пиксельных значений незаполненных блоков текстурного изображения, обозначенных текстурными блоками EOM.

В соответствии с вариантом осуществления, процесс заполнения может быть применен к геометрическому и/или текстурному изображению. Процесс заполнения может использоваться для заполнения пустого пространства между фрагментами для генерации кусочно-гладкого изображения, подходящего для сжатия видео.

Пример заполнения изображения предоставлен в секциях 2.2.6 и 2.2.7 V-PCC.

На этапе 3500, кодер видео VENC может кодировать сгенерированные изображения/уровни T1 и G1.

На этапе 3600, кодер OMENC может кодировать карту заполненности в качестве изображения, как подробно описано, например, в разделе 2.2.2 V-PCC. Может использоваться кодирование с потерями или без потерь.

В соответствии с вариантом осуществления, кодер видео ENC и/или OMENC может представлять собой HEVC-кодер.

На этапе 3700, кодер PIENC может кодировать вспомогательную информацию PI фрагмента и, возможно, дополнительные метаданные, такие как размер T блока, ширина W и высота H геометрического/текстурного изображений.

В соответствии с вариантом осуществления, вспомогательная информация фрагмента может быть дифференциально кодирована (как определено, например, в разделе 2.4.1 V-PCC).

На этапе 3800, мультиплексор может быть применен к сгенерированным выходам этапов 3500, 3600 и 3700, и в результате эти выходы могут быть мультиплексированы

вместе для генерации битового потока, представляющего базовый уровень BL. Следует отметить, что информация метаданных представляет малую часть всего битового потока. Объем информации сжимается с использованием кодеков видео.

На фиг. 4 показана блок-схема примера основанного на изображении декодера 4000 облака точек в соответствии по меньшей мере с одним из настоящих вариантов осуществления.

На этапе 4100, демультиплексор DMUX может применяться для демультиплексирования кодированной информации битового потока, представляющего базовый уровень BL.

На этапе 4200, декодер видео VDEC может декодировать кодированную информацию, чтобы получить по меньшей мере одно декодированное геометрическое изображение DGI и по меньшей мере одно декодированное текстурное изображение DTI.

На этапе 4300, декодер OMDEC может декодировать кодированную информацию, чтобы получить декодированную карту заполненности DOM.

В соответствии с вариантом осуществления, декодер видео VDEC и/или OMDEC может представлять собой HEC-декодер.

На этапе 4400, декодер PIDEС может декодировать кодированную информацию, чтобы получить вспомогательную информацию DPI фрагмента.

Возможно, метаданные также могут быть получены из битового потока BL.

На этапе 4500, модуль генерации геометрии, GGM, может получать геометрию RG восстановленного кадра IRPCF облака точек из по меньшей мере одного декодированного геометрического изображения DGI, декодированной карты заполненности DOM, декодированной вспомогательной информации DPI фрагмента и возможных дополнительных метаданных.

Модуль GGM генерации геометрии может использовать декодированную информацию карты заполненности DOM для того, чтобы определить местоположение непустых пикселей в по меньшей мере одном декодированном геометрическом изображении DGI.

Непустые пиксели принадлежат либо заполненным блокам, либо опорным блокам EOM в зависимости от пиксельных значений декодированной информации заполненности DOM и значения D1-D0, как объяснено выше.

В соответствии с вариантом осуществления этапа 4500, модуль GGM генерации геометрии может получать две из 3D координат промежуточных 3D выборок из координат непустых пикселей.

В соответствии с вариантом осуществления этапа 4500, когда непустые пиксели принадлежат опорному блоку EOM, модуль GGM генерации геометрии может получить третью из 3D координат промежуточных 3D выборок из битовых значений кодовых слов EOM.

Например, согласно примеру, показанному на фиг. 6, кодовое слово EOM, EOMC, используется для определения 3D координат промежуточных 3D выборок P_{i1} и P_{i2} .

Третья координата промежуточной 3D выборки P_{i1} может быть получена, например, из D0 посредством $D_{i1}=D0+3$, и третья координата восстановленной 3D выборки P_{i2} может быть получена, например, из D0 посредством $D_{i2}=D0+5$. Значение смещения (3 или 5) представляет собой количество интервалов между D0 и D1 вдоль линии проекции.

В соответствии с вариантом осуществления, когда непустые пиксели принадлежат заполненному блоку, модуль GGM генерации геометрии может получить 3D координаты восстановленных 3D выборок из координат непустых пикселей, значений непустых

пикселей одного из по меньшей мере одного декодированного геометрического изображения DGI, декодированной вспомогательной информации фрагмента и, возможно, из дополнительных метаданных.

Использование непустых пикселей основано на соотношении 2D пикселей с 3D выборками. Например, при проецировании в V-PCC, 3D координаты восстановленных 3D выборок могут быть выражены в терминах глубины $\delta(u, v)$, тангенциального сдвига $s(u, v)$ и би-тангенциального сдвига $r(u, v)$ следующим образом:

$$\delta(u, v) = \delta_0 + g(u, v)$$

$$s(u, v) = s_0 - u_0 + u$$

$$r(u, v) = r_0 - v_0 + v$$

где $g(u, v)$ представляет собой компонент яркости декодированного геометрического изображения DGI, (u, v) представляет собой пиксел, ассоциированный с восстановленной 3D выборкой, (δ_0, s_0, r_0) представляет собой 3D местоположение связанного компонента, которому принадлежит восстановленная 3D выборка, и (u_0, v_0, u_1, v_1) являются координатами в плоскости проекции, определяющей 2D ограничивающий прямоугольник, охватывающий проекцию фрагмента, ассоциированного со связанным компонентом.

На этапе 4600, модуль TGM генерации текстуры может получить текстуру восстановленного кадра IRPCF облака точек из геометрии RG и по меньшей мере одного декодированного текстурного изображения DTI.

В соответствии с вариантом осуществления этапа 4600, модуль TGM генерации текстуры может получить текстуру непустых пикселей, которые принадлежат опорному блоку EOM, из соответствующего текстурного блока EOM. Местоположения текстурных блоков EOM в текстурном изображении определяются согласно процедуре (раздел 9.4.5 V-PCC).

В соответствии с вариантом осуществления этапа 4600, модуль TGM генерации текстуры может вывести текстуру непустых пикселей, которые принадлежат заполненному блоку, непосредственно как пиксельные значения либо первого, либо второго текстурного изображения.

Фиг. 7 схематично иллюстрирует примерный синтаксис битового потока 7000, представляющего базовый уровень BL в соответствии с по меньшей мере одним из настоящих вариантов осуществления.

Битовый поток содержит заголовок SH 7100 битового потока и по меньшей мере одну группу потока кадров GOFs 7110, ... 7120, ... 7130 и т.д.

Группа потока кадров GOFs содержит заголовок HS 7121, по меньшей мере один синтаксический элемент OMS 7122, представляющий карту заполненности OM, по меньшей мере один синтаксический элемент GVS 7123, представляющий по меньшей мере одно геометрическое изображение (или видео), по меньшей мере один синтаксический элемент TVS 7125, представляющий по меньшей мере одно текстурное изображение (или видео), и по меньшей мере один синтаксический элемент PIS 7124, представляющий вспомогательную информацию фрагмента и другие дополнительные метаданные.

В одном варианте, группа потоков кадров GOFs содержит по меньшей мере один поток кадров.

В V-PCC, метаданные могут быть разделены на две категории:

- Метаданные на фрагмент описывают координаты каждого фрагмента в 2D изображениях глубины и текстуры (цвета) (U_0 и V_0) и в 3D пространстве (U_1, V_1 и D_1), а также ширину и высоту фрагмента ($\delta\text{Size}U_0$ и $\delta\text{Size}V_0$).

- Метаданные на блок обеспечивают, для каждого блока $N \times N$ изображений глубины и текстуры, информацию, которая указывает:

- какому фрагменту принадлежит текущий блок (информация индекса блока к фрагменту); и

5 - какие пикселы в текущем блоке соответствуют спроецированным точкам (карта заполненности).

Метаданные на фрагмент, требуемые, чтобы де-проецировать каждый 2D фрагмент в 3D пространство, относительно невелики. Имеется только несколько параметров для отправки на фрагмент (7 в текущей версии V-PCC), и количество фрагментов обычно
10 мало (сотни).

Карта заполненности и метаданные индекса блока к фрагменту, однако, требуются для всех пикселов мозаичного изображения. Для облака точек, которое использует 10 битов на пространственную координату, размер изображения фрагмента обычно составляет 1280×1280 пикселов, приводя к намного большему количеству данных для
15 кодирования.

В V-PCC, способ пытается уменьшить размер кодированной карты заполненности и метаданных отображения блока на фрагмент посредством (1) смешивания кодирования карты заполненности и метаданных индекса блока к фрагменту и (2) снижения точности
20 обоих, причем вместо кодирования информации для каждого пиксела она кодируется только один раз для блоков $N \times N$ пикселов. Хотя это несколько снижает эффективность кодирования изображений фрагмента глубины и цвета (поскольку они больше, чем они были бы в противном случае), это значительно уменьшает размер метаданных. Для индекса блока к фрагменту, N обычно равно 16, что уменьшает объем данных в 256 раз. Для карты заполненности, N может быть 1, 2, 4, 8 или 16; 4 является типичным,
25 что уменьшает объем данных в 16 раз.

Определим следующее:

1 - Карта заполненности с полной точностью: она содержит информацию для каждого пиксела. Карта заполненности доступна только на уровне кодера.

2 - Карта заполненности с блочной точностью (4 в типовом случае): она содержит
30 информацию для блока с точностью карты заполненности. Эта точность называется "малым блоком". Она доступна на стороне кодера и передается на декодер.

3 - Индекс блока к фрагменту при блочном разрешении (16 в типовом случае): это содержит информацию для блока с разрешением карты заполненности. Она доступна на стороне кодера и передается на декодер.

35 Настоящие варианты осуществления обеспечивают способ и/или устройство для удаления незначимых или менее значимых точек облака точек путем изменения некоторых пиксельных значений в карте заполненности и/или в карте блока на индекс поблочных метаданных, предусмотренных для блока изображения глубины и текстуры, полученного проецированием точек облака точек на плоскость проекции.

40 Первый вариант осуществления

Фиг. 8 иллюстрирует первый вариант осуществления того, как исходное облако точек (8100 на фиг. 8, 32×32 - полное разрешение) может обрабатываться и восстанавливаться в восстановленное облако точек (8400 на фиг. 8 - уменьшенное разрешение) с использованием карты заполненности (8200 на фиг. 8 - уменьшенное разрешение с использованием блоков 4×4) и индекса блока к фрагменту (8300 на фиг. 8 - дополнительно уменьшенное разрешение с использованием блоков 16×16).

Вариант осуществления предлагает способ, который сокращает данные карты заполненности и фрагмента на индекс, когда несколько точек присутствуют в блоке

на стороне кодера, чтобы избежать создания множества бесполезных точек (3D выборов) в восстановленном облаке точек. Этот способ уменьшает данные, подлежащие кодированию посредством:

Обновление карты заполненности для малых блоков.

5 Для каждого малого блока (обычно 4×4), выполнить:

a. отсчитать количество точек $NP_{\text{small_block}}$ в исходном облаке точек с использованием полной точности карты заполненности,

10 b. если $NP_{\text{small_block}} \leq Th_{\text{small_block}}$, где $Th_{\text{small_block}}$ является заданным значением, установить карту заполненности на незаполненную, в противном случае, установить заполненность блока на единицу. Типовое число для $Th_{\text{small_block}}$ равно 1.

Пусть OM_{reduce} будет обновленной картой заполненности исходного облака точек при полном разрешении.

15 В соответствии с одним вариантом, этот первый вариант осуществления способа дополнительно содержит:

Из OM_{reduce} , обновить индекс блока к фрагменту при блочном разрешении. Для каждого блока, выполнить:

c. отсчитать количество заполненных блоков: NB_{block}

20 d. если $NB_{\text{block}} = 0$, маркировать блок как незаполненный в индекс блока к фрагменту.

В противном случае, маркировать блок как заполненный.

25 Однако, как показано на фиг. 8, когда блок заполнен, следствием является то, что многочисленные точки будут восстанавливаться во время процесса декодирования (если точность заполненности установлена на 4, количество восстанавливаемых точек на блок равно 16). В частности, если одна точка заполняет малый блок 4×4 на стороне кодера, то индекс блока к фрагменту указывает блок как заполненный. Во время процесса декодирования, 16 точек будут генерироваться вместо 1.

Соответственно, фиг. 8a иллюстрирует другую альтернативу первого варианта осуществления, из исходного облака точек (8100a на фиг. 8a, полная точность карты заполненности), выполнить следующее:

30 a. отсчитать количество точек в блоке при разрешении блока: NP_{block}

b. если $NP_{\text{block}} \leq Th_{\text{block}}$, где Th_{block} является заданным значением, удалить блок из индекса блока к фрагменту. Типовое значение для Th_{block} равно 4.

35 Наконец, получить обновленную карту заполненности для малых блоков (8200 на фиг. 8a) и обновленный фрагмент к индексу для блочного разрешения (8300 на фиг. 8a). Они указывают данные, которые будут переданы на сторону декодера.

40 Соответственно, дальнейшие улучшения в эффективности показаны на фиг. 8a по сравнению с вариантом осуществления на фиг. 8. Как можно видеть, дальнейшее улучшение на фиг. 8a приводит к дальнейшему подавлению малых областей исходного облака точек, которые являются незначимыми или менее значимыми. Даже если некоторая информация пропущена, восстановленное облако точек является менее плотным, чем то, которое показано на фиг. 8 и более соответствующим исходному.

Второй вариант осуществления

45 Второй вариант осуществления основан на вычислении расстояния для каждого блока для того, чтобы оценить, является ли восстановленное облако точек близким или отдаленным относительно исходного облака точек. Расстояние представляет собой расстояние, от точки до точки, между двумя облаками точек.

Пусть A и B представляют собой два набора точек в 3D пространстве. Расстояние

от A до B определяется как

$$dist_{point}(A,B) = \sum_{p \in A} \|p - q_{closest}(p,B)\|_2^2$$

где норма представляет евклидово расстояние, и ближайшее ($closest$) B от точки p в A определяется как

$$q_{closest}(p,B) = \operatorname{argmin}_{q \in B} \|p - q\|_2^2$$

Этот вариант осуществления избегает отправки восстановленного облака точек, которое имеет вычисленное расстояние, которое больше, чем пороговое расстояние, и, таким образом, избегает слишком плотного восстановления облака точек. Поэтому этот вариант осуществления также уменьшает объем данных, подлежащих сжатию.

Что касается выигрыша: мы наблюдали выигрыш 1% в метриках (но это было сделано для другого алгоритма и должно быть сделано только на V-PCC).

На фиг. 9 показана блок-схема, иллюстрирующая пример системы, в которой реализованы различные аспекты и варианты осуществления.

Система 9000 может быть воплощена как одно или более устройств, включающих в себя различные компоненты, описанные ниже, и выполнена с возможностью выполнения одного или нескольких аспектов, описанных в этом документе. Примеры оборудования, которое может формировать всю или часть системы 9000, включают в себя персональные компьютеры, ноутбуки, смартфоны, планшетные компьютеры, цифровые мультимедийные телевизионные приставки, цифровые телевизионные приемники, персональные системы записи видео, соединенные домашние электронные приборы, соединенные транспортные средства и ассоциированные с ними системы обработки, наголовные устройства отображения (HMD, прозрачные очки), проекторы (излучатели), “caves” (система, включающая в себя множество дисплеев), серверы, кодеры видео, декодеры видео, пост-процессоры, обрабатывающие вывод из декодера видео, пре-процессоры, обеспечивающие ввод в кодер видео, web-серверы, телевизионные приставки и любое другое устройство для обработки облака точек, видео или изображения или другие коммуникационные устройства. Элементы системы 9000, отдельно или в комбинации, могут быть реализованы в одной интегральной схеме, множестве ИС и/или на дискретных компонентах. Например, по меньшей мере в одном варианте осуществления, элементы обработки и кодера/декодера системы 9000 могут быть распределены по нескольким ИС и/или дискретным компонентам. В различных вариантах осуществления, система 9000 может быть коммуникативно связана с другими подобными системами или с другими электронными устройствами, например, через коммуникационную шину или через специализированные порты ввода и/или вывода. В различных вариантах осуществления, система 9000 может быть выполнена с возможностью реализации одного или нескольких аспектов, описанных в этом документе.

Система 9000 может включать в себя по меньшей мере один процессор 9010, выполненный с возможностью исполнения инструкций, загружаемых в него для реализации, например, различных аспектов, описанных в этом документе. Процессор 9010 может включать в себя встроенную память, интерфейс ввода/вывода и различные другие схемы, известные в данной области техники. Система 9000 может включать в себя по меньшей мере одну память 9020 (например, энергозависимое устройство памяти и/или энергонезависимое устройство памяти).

Система 9000 может включать в себя устройство хранения 9040, которое может включать в себя энергонезависимую память и/или энергозависимую память, включая,

но без ограничения, электрически стираемую программируемую постоянную память (EEPROM), постоянную память (ROM), программируемую постоянную память (PROM), память с произвольным доступом (RAM), динамическую память с произвольным доступом (DRAM), статическую память с произвольным доступом (SRAM), флэш-память, накопитель на магнитных дисках и/или накопитель на оптических дисках. Устройство хранения 9040 может включать в себя внутреннее устройство хранения, присоединенное устройство хранения и/или сетевое устройство хранения, в качестве неограничивающих примеров.

Система 9000 может включать в себя модуль 9030 кодера/декодера, выполненный, например, с возможностью обработки данных для обеспечения кодированных данных или декодированных данных, и модуль 9030 кодера/декодера может включать в себя свой собственный процессор и память. Модуль 9030 кодера/декодера может представлять модуль(и), который может быть включен в устройство для выполнения функций кодирования и/или декодирования. Как известно, устройство может включать в себя один или оба из модуля кодирования и декодирования. Кроме того, модуль 9030 кодера/декодера может быть реализован как отдельный элемент системы 9000 или может быть встроен в процессор 9010 как комбинация аппаратных средств и программного обеспечения, как известно специалистам в данной области техники.

Программный код для загрузки в процессор 9010 или кодер/декодер 9030 для выполнения различных аспектов, описанных в этом документе, может храниться в устройстве хранения 9040 и затем загружаться в память 9020 для исполнения процессором 9010. В соответствии с различными вариантами осуществления, одно или более из процессора 9010, памяти 9020, устройства хранения 9040 и модуля 9030 кодера/декодера могут хранить один или более различных элементов во время выполнения процессов, описанных в этом документе. Такие сохраненные элементы могут включать в себя, но без ограничения, кадр облака точек, кодированные/декодированные геометрические/текстурные видео/изображения или части кодированных/декодированных геометрических/текстурных видео/изображений, битовый поток, матрицы, переменные и промежуточные или конечные результаты из обработки уравнений, формул, операций и операционной логики.

В некоторых вариантах осуществления, память в процессоре 9010 и/или модуле 9030 кодера/декодера может использоваться для хранения инструкций и обеспечения рабочей памяти для обработки, которая может выполняться во время кодирования или декодирования.

В других вариантах осуществления, однако, память, внешняя по отношению к устройству обработки (например, устройство обработки может быть либо процессором 9010, либо модулем 9030 кодера/декодера), может использоваться для одной или нескольких из этих функций. Внешняя память может представлять собой память 9020 и/или устройство хранения 9040, например, динамическую энергозависимую память и/или энергонезависимую флэш-память. В нескольких вариантах осуществления, внешняя энергонезависимая флэш-память может использоваться для хранения операционной системы телевизора. По меньшей мере в одном варианте осуществления, быстродействующая внешняя динамическая энергозависимая память, такая как RAM, может использоваться в качестве рабочей памяти для операций кодирования и декодирования видео, таких как для MPEG-2 часть 2 (также известно как рекомендация H.261 ITU-T и ISO/IEC 13818-2, также известно как MPEG-2 Video), HEVC (Высокоэффективное кодирование видео) или VVC (Универсальное кодирование видео).

Ввод в элементы системы 9000 может обеспечиваться через различные устройства

ввода, как показано в блоке 9130. Такие устройства ввода включают в себя, но без ограничения, (i) RF-часть, которая может принимать RF-сигнал, передаваемый, например, посредством беспроводной связи вещательной станцией, (ii) терминал полного ввода, (iii) USB-терминал ввода и/или (iv) HDMI-терминал ввода.

5 В различных вариантах осуществления, устройства ввода блока 9130 могут иметь ассоциированные соответствующие элементы обработки ввода, как известно в данной области техники. Например, RF-часть может быть ассоциирована с элементами, необходимыми для (i) выбора требуемой частоты (также упоминается как выбор сигнала или ограничение по полосе сигнала до некоторой полосы частот), (ii) понижающего преобразования выбранного сигнала, (iii) ограничения по полосе повторно до более узкой полосы частот, чтобы выбрать (например) полосу частот сигнала, которая может упоминаться как канал в некоторых вариантах осуществления, (iv) демодуляции преобразованного с понижением и ограниченного по полосе сигнала, (v) выполнения коррекции ошибок и (vi) демультиплексирования для выбора требуемого потока пакетов данных. RF-часть различных вариантов осуществления может включать в себя один или более элементов для выполнения этих функций, например селекторы частоты, селекторы сигнала, ограничители полосы, селекторы канала, фильтры, понижающие преобразователи, демодуляторы, корректоры ошибок и демультиплексоры. RF-часть может включать в себя тюнер, который выполняет различные из этих функций, включая, например, понижающее преобразование принятого сигнала на более низкую частоту (например, промежуточную частоту или частоту, близкую к базовой полосе) или в базовую полосу.

В одном варианте осуществления с телевизионной приставкой, RF-часть и ассоциированный с ней элемент обработки ввода могут принимать RF-сигнал, передаваемый по проводному (например, кабельному) носителю. Тогда, RF-часть может выполнять выбор частоты путем фильтрации, понижающего преобразования и повторной фильтрации до требуемой полосы частот.

Различные варианты осуществления переупорядочивают очередность описанных выше (и других) элементов, удаляют некоторые из этих элементов и/или добавляют другие элементы, выполняющие аналогичные или различные функции.

Добавление элементов может включать в себя вставку элементов между существующими элементами, например, вставку усилителей и аналого-цифрового преобразователя. В различных вариантах осуществления, RF-часть может включать в себя антенну.

35 Кроме того, USB- и/или HDMI-терминалы могут включать в себя соответствующие процессоры интерфейса для соединения системы 9000 с другими электронными устройствами через USB- и/или HDMI-соединения. Следует понимать, что различные аспекты обработки ввода, например коррекции ошибок Рида-Соломона, могут быть реализованы, например, в отдельной ИС обработки ввода или в процессоре 9010 по мере необходимости. Аналогично, аспекты обработки USB- или HDMI-интерфейса могут быть реализованы в отдельных ИС интерфейса или в процессоре 9010 по мере необходимости. Демодулированный, скорректированный и демультиплексированный поток может быть предоставлен в различные элементы обработки, включая, например, процессор 9010 и кодер/декодер 9030, работающие в комбинации с элементами памяти и хранения для обработки потока данных, как это необходимо для представления на устройстве вывода.

Различные элементы системы 9000 могут быть предусмотрены в интегрированном корпусе. Внутри интегрированного корпуса различные элементы могут быть соединены

между собой и могут передавать данные между собой с использованием подходящего соединительного устройства 9140, например внутренней шины, как известно в данной области техники, включая шину I2C, разводку и печатные платы.

5 Система 9000 может включать в себя интерфейс 9050 связи, который обеспечивает связь с другими устройствами по каналу 9060 связи. Интерфейс 9050 связи может включать в себя, но без ограничения, приемопередатчик, выполненный с возможностью передачи и приема данных по каналу 9060 связи. Интерфейс 9050 связи может включать в себя, но без ограничения, модем или сетевую карту, и канал 9060 связи может быть реализован, например, в проводной и/или беспроводной среде.

10 Данные могут передаваться в потоковом режиме в систему 9000, в различных вариантах осуществления, используя сеть Wi-Fi, такую как IEEE 802.11. Сигнал Wi-Fi этих вариантов осуществления может приниматься по каналу 9060 связи и интерфейсу 9050 связи, которые адаптированы для связи Wi-Fi. Канал 9060 связи этих вариантов осуществления может быть, как правило, соединен с точкой доступа или маршрутизатором, который обеспечивает доступ к внешним сетям, включая Интернет, для обеспечения возможности потоковой передачи приложений и других передач доставки контента на пользовательский терминал поверх сетей провайдеров связи.

15 Другие варианты осуществления могут предоставлять потоковые данные в систему 9000 с использованием телевизионной приставки, которая доставляет данные по HDMI-соединению блока 9130 ввода.

Другие варианты осуществления могут предоставлять потоковые данные в систему 9000 с использованием RF-соединения блока 9130 ввода.

20 Следует понимать, что сигнализация может быть выполнена множеством способов. Например, один или более синтаксических элементов, флагов и т.д. могут использоваться, чтобы сигнализировать информацию на соответствующий декодер в различных вариантах осуществления.

30 Система 9000 может выдавать выходной сигнал на различные устройства вывода, включая дисплей 9100, громкоговорители 9110 и другие периферийные устройства 9120. Другие периферийные устройства 9120 могут включать в себя, в различных примерах вариантов осуществления, один или более из автономного DVR, проигрывателя дисков, стереосистемы, системы освещения и других устройств, которые обеспечивают функцию на основе вывода системы 9000.

В различных вариантах осуществления, управляющие сигналы могут передаваться между системой 9000 и дисплеем 9100, громкоговорителями 9110 или другими периферийными устройствами 9120, используя сигнализацию, такую как AV.Link (Аудио/видео линия), СЕС (Управление бытовыми электронными приборами) или другие протоколы связи, которые обеспечивают управление от устройства к устройству с вмешательством пользователя или без него.

40 Устройства вывода могут быть коммуникативно связаны с системой 9000 по выделенным соединениям через соответствующие интерфейсы 9070, 9080 и 9090.

Альтернативно, устройства вывода могут быть соединены с системой 9000 с использованием канала 9060 связи через интерфейс 9050 связи. Дисплей 9100 и громкоговорители 9110 могут быть интегрированы в один блок с другими компонентами системы 9000 в электронном устройстве, таком как, например, телевизор.

45 В различных вариантах осуществления, интерфейс 9070 дисплея может включать в себя драйвер дисплея, такой как, например, микросхема контроллера синхронизации (T Con).

Дисплей 9100 и громкоговоритель 9110 могут альтернативно быть отдельными от

одного или нескольких других компонентов, например, если RF-часть ввода 9130 является частью отдельной телевизионной приставки. В различных вариантах осуществления, в которых дисплей 9100 и громкоговорители 9110 могут быть внешними компонентами, выходной сигнал может предоставляться через выделенные выходные соединения, включая, например, HDMI-порты, USB-порты или COM-выводы.

В соответствии с другим вариантом осуществления, предложен способ сокращения облака точек, представляющего изображение, содержащий: получение 2-D фрагмента облака точек путем проецирования 3-D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей; разделение фрагмента на множество меньших блоков; определение количества пикселей в каждом из множества меньших блоков; получение обновленной карты заполненности на основе определенного количества пикселей в каждом из множества меньших блоков; и получение восстановленного облака точек на основе обновленной карты заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек.

В соответствии с другим вариантом осуществления, предложено устройство для сокращения облака точек, представляющего изображение, содержащее: средство для получения 2-D фрагмента облака точек посредством проецирования 3-D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей; средство для разделения фрагмента на множество меньших блоков; средство для определения количества пикселей в каждом из множества меньших блоков; средство для получения обновленной карты заполненности на основе определенного количества пикселей в каждом из множества меньших блоков; и средство для получения восстановленного облака точек на основе обновленной карты заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек.

В соответствии с другим вариантом осуществления, предложено устройство для сокращения облака точек, представляющего изображение, содержащее один или более процессоров, причем один или более процессоров выполнены с возможностью: получать 2-D фрагмент облака точек посредством проецирования 3-D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей; разделять фрагмент на множество меньших блоков; определять количество пикселей в каждом из множества меньших блоков; получать обновленную карту заполненности на основе определенного количества пикселей в каждом из множества меньших блоков; и получать восстановленное облако точек на основе обновленной карты заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек.

В соответствии с другим вариантом осуществления, предлагается битовый поток, содержащий восстановленное облако точек, при этом битовый поток формируется посредством: получения 2-D фрагмента облака точек посредством проецирования 3-D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей; разделения фрагмента на множество меньших блоков; определения количества пикселей в каждом из множества меньших блоков; получения обновленной карты заполненности на основе определенного количества пикселей в каждом из множества меньших блоков; и получения восстановленного облака точек на основе обновленной карты заполненности, при этом восстановленное облако точек является менее плотным представлением облака точек.

В соответствии с другим вариантом осуществления, вариант осуществления дополнительно содержит получение обновленного индекса блока к фрагменту на основе обновленной карты заполненности, причем разрешение обновленной карты

заполненности выше, чем разрешение обновленного индекса блока к фрагменту.

В соответствии с другим вариантом осуществления, вариант осуществления дополнительно содержит сравнение количества пикселей в каждом из множества меньших блоков с некоторым значением.

5 В соответствии с другим вариантом осуществления, вариант осуществления дополнительно содержит установку соответствующего меньшего блока на незаполненный, если количество пикселей в соответствующем меньшем блоке меньше, чем пороговое значение.

В соответствии с другим вариантом осуществления, значение равно 1 или 4.

10 В соответствии с другим вариантом осуществления, множество меньших блоков являются блоками 4×4.

В соответствии с другим вариантом осуществления, разрешение обновленного индекса блока к фрагменту равно 16×16.

Кроме того, вариант осуществления обеспечивает компьютерную программу, содержащую инструкции, которые при исполнении одним или несколькими процессорами побуждают один или более процессоров выполнять способ кодирования или способ декодирования в соответствии с любым из вариантов осуществления, описанных выше. Один или более из настоящих вариантов осуществления также обеспечивают машиночитаемый носитель хранения, имеющий сохраненные на нем инструкции для кодирования или декодирования данных видео в соответствии со способами, описанными выше. Один или более вариантов осуществления также обеспечивают машиночитаемый носитель хранения, имеющий сохраненный на нем битовый поток, сгенерированный в соответствии со способами, описанными выше. Один или более вариантов осуществления также обеспечивают способ и устройство для передачи или приема битового потока, сгенерированного в соответствии со способами, описанными выше.

Здесь описаны различные способы, и каждый из способов включает в себя один или более этапов или действий для реализации описанного способа. Если только конкретный порядок этапов или действий не требуется для надлежащей работы способа, порядок и/или использование конкретных этапов и/или действий могут быть изменены или объединены.

Некоторые примеры описаны в отношении блок-схем и операционных блок-схем. Каждый блок представляет схемный элемент, модуль или часть кода, которая включает в себя одну или более исполняемых инструкций для реализации указанной логической функции(й). Следует также отметить, что в других реализациях, функция(и), указанная в блоках, может выполняться не в указанном порядке. Например, два блока, показанные последовательно, фактически могут исполняться по существу одновременно, или блоки могут иногда исполняться в обратном порядке в зависимости от используемых функциональных возможностей.

Реализации и аспекты, описанные здесь, могут быть реализованы, например, в виде способа или процесса, устройства, компьютерной программы, потока данных, битового потока или сигнала. Даже если обсуждается только в контексте одной формы реализации (например, обсуждается только как способ), реализация обсуждаемых признаков также может быть осуществлена в других формах (например, как устройство или компьютерная программа).

45 Способы могут быть реализованы, например, в процессоре, который относится к устройствам обработки в общем, включая, например, компьютер, микропроцессор, интегральную схему или программируемое логическое устройство. Процессоры также включают в себя устройства связи.

Кроме того, способы могут быть реализованы посредством инструкций, выполняемых процессором, и такие инструкции (и/или значения данных, созданные реализацией) могут храниться на машиночитаемом носителе хранения. Машиночитаемый носитель хранения может принимать форму машиночитаемого программного продукта, реализованного на одном или нескольких машиночитаемых носителях и имеющего машиночитаемый программный код, реализованный на нем, который может исполняться компьютером. Машиночитаемый носитель хранения, используемый здесь, может рассматриваться как долговременный (энергонезависимый) носитель хранения с присущей ему способностью сохранять в себе информацию, а также присущей ему способностью предоставлять извлечение из него информации. Машиночитаемый носитель хранения может представлять собой, например, но без ограничения, электронную, магнитную, оптическую, электромагнитную, инфракрасную или полупроводниковую систему, устройство или прибор или любую подходящую комбинацию вышеупомянутого. Следует понимать, что следующее, хотя и обеспечивает более конкретные примеры машиночитаемых носителей хранения, к которым могут быть применены настоящие варианты осуществления, является просто иллюстративным и не исчерпывающим перечнем, как легко понять специалисту в данной области техники: переносная компьютерная дискета; жесткий диск; постоянная память (ROM); стираемая программируемая постоянная память (EPROM или флэш-память); переносная постоянная память на компакт-диске (CD-ROM); оптическое устройство хранения; магнитное устройство хранения или любая подходящая комбинация вышеупомянутого.

Инструкции могут формировать прикладную программу, материально воплощенную на машиночитаемом носителе.

Инструкции могут представлять собой, например, аппаратные средства, микропрограммное обеспечение, программное обеспечение или комбинацию. Инструкции могут быть, например, в операционной системе, отдельном приложении или их комбинации. Таким образом, процессор может быть охарактеризован, например, как устройство, выполненное с возможностью исполнения процесса, и устройство, которое включает в себя машиночитаемый носитель (такой как устройство хранения), имеющий инструкции для выполнения процесса. Кроме того, машиночитаемый носитель может хранить, в дополнение или вместо инструкций, значения данных, созданные реализацией.

Устройство может быть реализовано, например, в соответствующих аппаратных средствах, программном обеспечении и микропрограммном обеспечении. Примеры таких устройств включают в себя персональные компьютеры, ноутбуки, смартфоны, планшетные компьютеры, цифровые мультимедийные телевизионные приставки, цифровые телевизионные приемники, персональные системы записи видео, соединенные домашние электронные приборы, наголовные устройства отображения (HMD, прозрачные очки), проекторы (излучатели), "caves" (система, включающая в себя множество дисплеев), серверы, кодеры видео, декодеры видео, пост-процессоры, обрабатывающие вывод из декодера видео, пре-процессоры, обеспечивающие ввод в кодер видео, web-серверы, телевизионные приставки и любое другое устройство для обработки облака точек, видео или изображения или другие коммуникационные устройства. Как должно быть ясно, оборудование может быть мобильным и даже установленным в подвижном транспортном средстве.

Компьютерное программное обеспечение может быть реализовано процессором 9010 или аппаратными средствами, или комбинацией аппаратных средств и программного обеспечения. В качестве неограничивающего примера, варианты

осуществления могут быть также реализованы посредством одной или нескольких интегральных схем. Память 9020 может быть любого типа, соответствующего технического окружения, и может быть реализована с использованием любой подходящей технологии хранения данных, такой как оптические устройства хранения, магнитные устройства хранения, полупроводниковые устройства памяти, стационарная память и сменная память, в качестве неограничивающих примеров. Процессор 9010 может быть любого типа, пригодного для технического окружения, и может включать в себя один или более микропроцессоров, компьютеров общего назначения, компьютеров специального назначения и процессоров на основе многоядерной архитектуры, в качестве неограничивающих примеров.

Как будет очевидно специалисту в данной области техники, реализации могут формировать множество сигналов, отформатированных для переноса информации, которая может быть, например, сохранена или передана. Информация может включать в себя, например, инструкции для выполнения способа или данные, созданные одной из описанных реализаций. Например, сигнал может быть отформатирован для переноса битового потока описанного варианта осуществления. Такой сигнал может быть отформатирован, например, как электромагнитная волна (например, с использованием радиочастотной части спектра) или как сигнал базовой полосы. Форматирование может включать в себя, например, кодирование потока данных и модуляцию носителя закодированным потоком данных. Информация, которая переносит сигнал, может представлять собой, например, аналоговую или цифровую информацию. Сигнал может передаваться по множеству различных проводных или беспроводных линий связи, как известно. Сигнал может храниться на считываемом процессором носителе.

Используемая здесь терминология предназначена только для описания конкретных вариантов осуществления и не предназначена для ограничения. Как используется здесь, формы единственного числа могут включать в себя также формы множественного числа, если контекст ясно не указывает иное. Кроме того, следует понимать, что термины "включает в себя/содержит" и/или "включающий в себя/содержащий" при использовании в спецификации, могут определять присутствие указанных, например, признаков, целых чисел, этапов, операций, элементов и/или компонентов, но не исключают присутствия или добавления одного или нескольких других признаков, целых чисел, этапов, операций, элементов, компонентов и/или их групп. Кроме того, когда элемент упоминается как "реагирующий" или "соединенный" с другим элементом, он может непосредственно реагировать на другой элемент или быть соединенным с другим элементом, или могут присутствовать промежуточные элементы. В отличие от этого, когда элемент упоминается как "непосредственно реагирующий" или "непосредственно соединенный" с другим элементом, промежуточные элементы отсутствуют.

Следует принимать во внимание, что использование любого из символов/терминов "/", "и/или" или "по меньшей мере одно из", например, в случае "A/B" и "по меньшей мере одно из A и B", может быть предназначено для того, чтобы охватывать выбор только первой упомянутой опции (A) или выбор только второй упомянутой опции (B) или выбор обеих опций (A и B). В качестве дополнительного примера, в случаях "A, B и/или C" и "по меньшей мере одного из A, B и C", такая фраза предназначена, чтобы охватывать выбор только первой упомянутой опции (A) или выбор только упомянутой опции (B) или выбор только третьей упомянутой опции (C), или выбор только первой и второй упомянутых опций (A и B), или только выбор первой и третьей упомянутых опций (A и C), или только выбор второй и третьей упомянутых опций (B и C), или выбор всех трех упомянутых опций (A и B и C). Это может быть расширено, как очевидно

специалисту в данной области техники, на многие элементы, которые перечислены.

В настоящей заявке могут использоваться различные числовые значения, например пороговое значение 1 или 4 для сравнения с количеством пикселей в соответствующем меньшем блоке фрагмента. Конкретные значения могут быть приведены для иллюстративных целей, и описанные аспекты не ограничиваются этими конкретными значениями.

Следует понимать, что хотя термины "первый", "второй" и т.д. могут использоваться здесь для описания различных элементов, эти элементы не ограничиваются этими терминами. Эти термины используются только для различения одного элемента от другого. Например, первый элемент может быть назван вторым элементом, и, аналогично, второй элемент может быть назван первым элементом, не выходя за рамки объема настоящей заявки. Не подразумевается никакого упорядочения между первым элементом и вторым элементом.

Ссылки на "один вариант осуществления" или "вариант осуществления", или "одну реализацию", или "реализацию", а также другие их вариации, часто используются для указания того, что конкретный признак, структура, характеристика и т.д. (описанные в связи с вариантом осуществления/реализацией) включены в по меньшей мере один вариант осуществления/реализацию. Таким образом, появление фразы "в одном варианте осуществления" или "в варианте осуществления", или "в одной реализации", или "в реализации", а также любые другие вариации, появляющиеся в различных местах в этой заявке, не обязательно все ссылаются на один и тот же вариант осуществления.

Подобным образом, ссылка здесь на "в соответствии с вариантом осуществления/примером/реализацией" или "в варианте осуществления/примере/реализации", а также другие их вариации часто используется для указания того, что конкретный признак, структура или характеристика (описанные в связи с вариантом осуществления/примером/реализацией) могут быть включены по меньшей мере в один вариант осуществления/пример/реализацию. Таким образом, появление выражения "в соответствии с вариантом осуществления/примером/реализацией" или "в варианте осуществления/примере/реализации" в различных местах спецификации не обязательно все относятся к одному и тому же варианту осуществления/примеру/реализации, и не являются отдельными или альтернативными вариантами осуществления/ примерами/реализацией, обязательно взаимно исключающими другие варианты осуществления/примеры/реализацию.

Ссылочные позиции, появляющиеся в формуле изобретения, представлены только в качестве иллюстрации и не должны ограничивать объем формулы изобретения. Хотя это и не описано явно, настоящие варианты осуществления/примеры и варианты могут использоваться в любой комбинации или подкомбинации.

Когда на чертеже представлена блок-схема последовательности операций, следует понимать, что она также предоставляет блок-схему соответствующего устройства. Аналогично, когда на чертеже представлена блок-схема устройства, следует понимать, что она также предоставляет блок-схему последовательности операций соответствующего способа/процесса.

Хотя некоторые из схем включают в себя стрелки на линиях связи для того, чтобы показать основное направление связи, должно быть понятно, что связь может происходить в противоположном направлении по отношению к показанным стрелкам.

Различные реализации включают в себя декодирование. "Декодирование", как используется в настоящей заявке, может охватывать все или часть процессов, выполняемых, например, над принятым кадром облака точек (включая, возможно, принятый битовый поток, в котором закодирован один или более кадров облака точек)

для того, чтобы сформировать конечный выходной сигнал, подходящий для отображения или для дальнейшей обработки в области восстановленного облака точек. В различных вариантах осуществления, такие процессы включают в себя один или более процессов, обычно выполняемых основанным на изображении декодером. В различных вариантах осуществления, такие процессы также, или альтернативно, включают в себя процессы, выполняемые декодером различных реализаций, описанных в этой заявке, например, декодером 2000 согласно фиг. 2 или декодером 4000 согласно фиг. 4.

В качестве дополнительных примеров, в одном варианте осуществления "декодирование" может относиться только к энтропийному декодированию, в другом варианте осуществления "декодирование" может относиться только к дифференциальному декодированию, и в другом варианте осуществления "декодирование" может относиться к комбинации энтропийного декодирования и дифференциального декодирования. То, предполагается ли, что фраза "процесс декодирования" относится конкретно к поднабору операций или, в общем, к более широкому процессу декодирования, будет ясно на основе контекста конкретных описаний и считается хорошо понятным специалистам в данной области техники.

Различные реализации включают в себя кодирование. Аналогичным образом, как описано выше для "декодирования", "кодирование", как используется в этой заявке, может охватывать все или часть процессов, выполняемых, например, над входным кадром облака точек для того, чтобы сформировать кодированный битовый поток. В различных вариантах осуществления, такие процессы включают в себя один или более процессов, обычно выполняемых основанным на изображении декодером. В различных вариантах осуществления, такие процессы также, или альтернативно, включают в себя процессы, выполняемые кодером различных реализаций, описанных в этой заявке, например, кодером 1000 согласно фиг. 1 или кодером 3000 согласно фиг. 3.

Кроме того, эта заявка ссылается на "определение" различных частей информации. Определение информации может включать в себя одно или более из, например, оценки информации, вычисления информации, предсказания информации или извлечения информации из памяти.

Кроме того, эта заявка может ссылаться на "осуществление доступа" к различным частям информации. Осуществление доступа к информации может включать в себя одно или более из, например, приема информации, извлечения информации (например, из памяти), сохранения информации, перемещения информации, копирования информации, вычисления информации, определения информации, предсказания информации или оценки информации.

Кроме того, эта заявка может ссылаться на "прием" различных частей информации. Прием, как и в случае "осуществления доступа", предполагается в качестве широкого термина. Прием информации может включать в себя одно или более из, например, осуществления доступа к информации или извлечения информации (например, из памяти). Кроме того, "прием" обычно осуществляется, тем или иным образом, при выполнении операций, таких как, например, хранение информации, обработка информации, передача информации, перемещение информации, копирование информации, стирание информации, вычисление информации, определение информации, предсказание информации или оценка информации.

Было описано несколько реализаций. Тем не менее, следует понимать, что могут выполняться различные модификации. Например, элементы различных реализаций могут быть объединены, дополнены, модифицированы или удалены для создания других

реализаций. Кроме того, специалисту в данной области техники будет понятно, что другие структуры и процессы могут заменять раскрытые, и полученные реализации будут выполнять по меньшей мере по существу ту же функцию(и) по меньшей мере по существу таким же образом, чтобы достичь по меньшей мере, по существу, того же результата(ов), что и раскрытые реализации. Соответственно, эти и другие реализации предполагаются настоящей заявкой.

(57) Формула изобретения

1. Способ, содержащий кодирование облака точек, причем кодирование облака точек содержит этапы, на которых:
 - получают карту заполненности, содержащую данные, указывающие, является ли пиксел изображения глубины или текстурного изображения заполненным или нет, для по меньшей мере одного блока карты заполненности, который имеет первый размер, определяют количество заполненных пикселов,
 - в ответ на определение того, что количество заполненных пикселов в этом по меньшей мере одном блоке меньше или равно заданному значению, обновляют карту заполненности, кодируют обновленную карту заполненности.
2. Способ по п.1, в котором упомянутое обновление карты заполненности содержит этап, на котором устанавливают пиксели блока в карте заполненности как незаполненные.
3. Способ по п.1 или 2, дополнительно содержащий этап, на котором обновляют данные индекса блока к фрагменту из обновленной карты заполненности, причем данные индекса блока к фрагменту указывают, какому фрагменту принадлежит блок, при этом фрагмент получается посредством проецирования 3D точек облака точек на плоскость проекции.
4. Способ по п.3, в котором упомянутое обновление данных индекса блока к фрагменту при блочном разрешении содержит этапы, на которых:
 - для по меньшей мере одного второго блока карты заполненности, который имеет второй размер больший, чем первый размер, определяют второе количество заполненных пикселов в обновленной карте заполненности,
 - в ответ на определение того, что второе количество заполненных пикселов в этом по меньшей мере одном втором блоке равно 0, устанавливают данные индекса блока к фрагменту для данного по меньшей мере одного второго блока как незаполненные.
5. Способ по п.4, в котором упомянутое обновление данных индекса блока к фрагменту при блочном разрешении дополнительно содержит этап, на котором, в ответ на определение того, что второе количество заполненных пикселов в упомянутом по меньшей мере одном втором блоке меньше, чем заданное второе значение, дополнительно обновляют карту заполненности.
6. Устройство, содержащее один или более процессоров, выполненный с возможностью кодировать облако точек, причем кодирование облака точек содержит:
 - получение карты заполненности, содержащей данные, указывающие, является ли пиксел изображения глубины или текстурного изображения заполненным или нет, для по меньшей мере одного блока карты заполненности, который имеет первый размер, определение количества заполненных пикселов,
 - в ответ на определение того, что количество заполненных пикселов в этом по меньшей мере одном блоке меньше или равно заданному значению, обновление карты заполненности,

кодирование обновленной карты заполненности.

7. Устройство по п.6, в котором упомянутое обновление карты заполненности содержит установку пикселей блока в карте заполненности как незаполненных.

8. Устройство по п.6, в котором один или более процессоров дополнительно выполнены с возможностью обновления данных индекса блока к фрагменту из обновленной карты заполненности, причем данные индекса блока к фрагменту указывают, какому фрагменту принадлежит блок, при этом фрагмент получается посредством проецирования 3D точек облака точек на плоскость проекции.

9. Устройство по п.8, в котором упомянутое обновление данных индекса блока к фрагменту при блочном разрешении содержит:

для по меньшей мере одного второго блока карты заполненности, который имеет второй размер больший, чем первый размер, определение второго количества заполненных пикселей в обновленной карте заполненности,

в ответ на определение того, что второе количество заполненных пикселей в этом по меньшей мере одном втором блоке равно 0, установку данных индекса блока к фрагменту для данного по меньшей мере одного второго блока как незаполненных.

10. Устройство по п.9, в котором упомянутое обновление данных индекса блока к фрагменту при блочном разрешении дополнительно содержит, в ответ на определение того, что второе количество заполненных пикселей в упомянутом по меньшей мере одном втором блоке меньше, чем заданное второе значение, дополнительное обновление карты заполненности.

11. Устройство по любому одному из пп.8-10, в котором один или более процессоров дополнительно выполнены с возможностью кодирования обновленных данных индекса блока к фрагменту.

12. Устройство по любому одному из пп.6-11, в котором упомянутое получение карты заполненности содержит получение 2D фрагмента облака точек посредством проецирования 3D точек облака точек на плоскость проекции, причем фрагмент имеет множество пикселей.

13. Устройство по п.12, при этом фрагмент имеет более высокое разрешение, чем обновленная карта заполненности.

14. Устройство по любому одному пп.8-13, при этом разрешение обновленной карты заполненности выше, чем разрешение обновленного индекса блока к фрагменту.

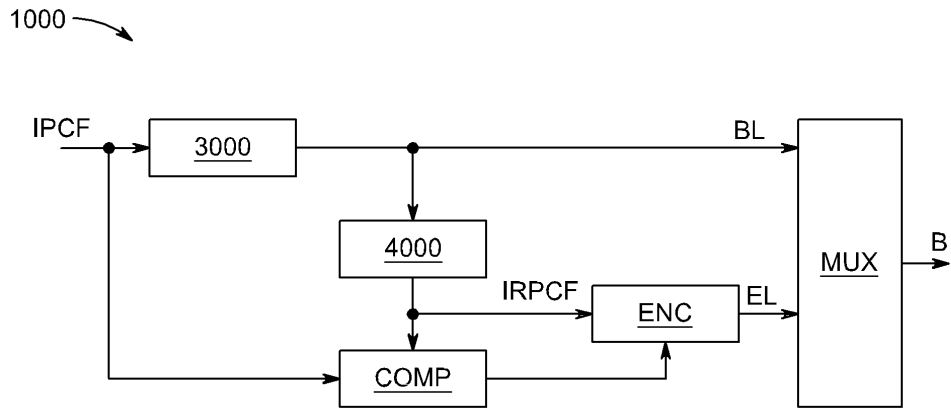
15. Долговременный машиночитаемый носитель, заключающий в себе содержимое данных, сгенерированное согласно способу по любому одному из пп.1-5 или устройством по любому одному из пп.6-14.

40

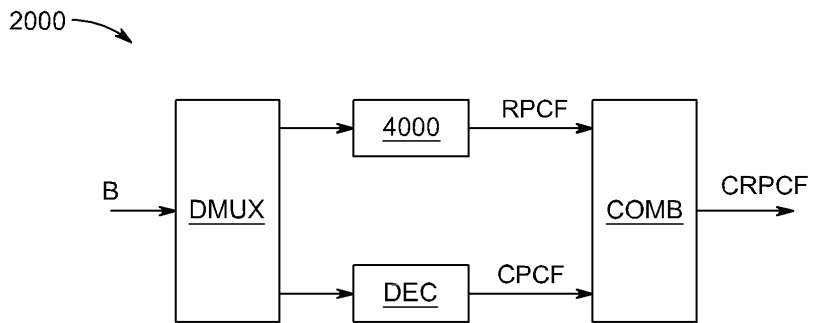
45

1

1/9

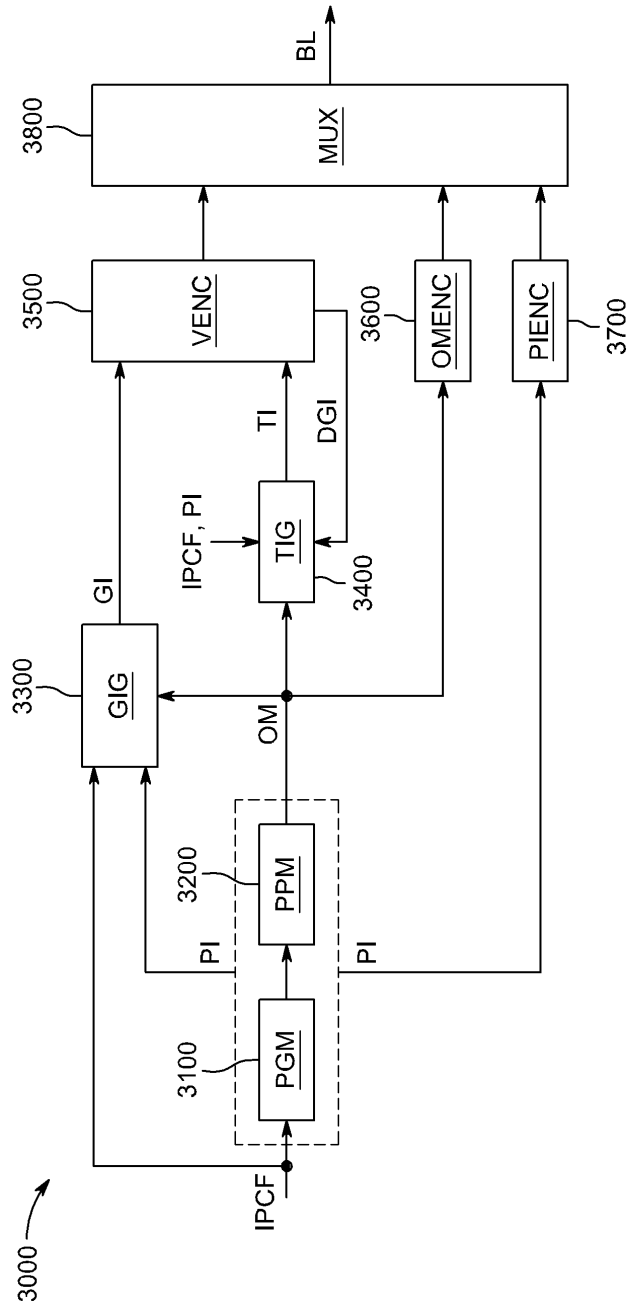


ФИГ. 1

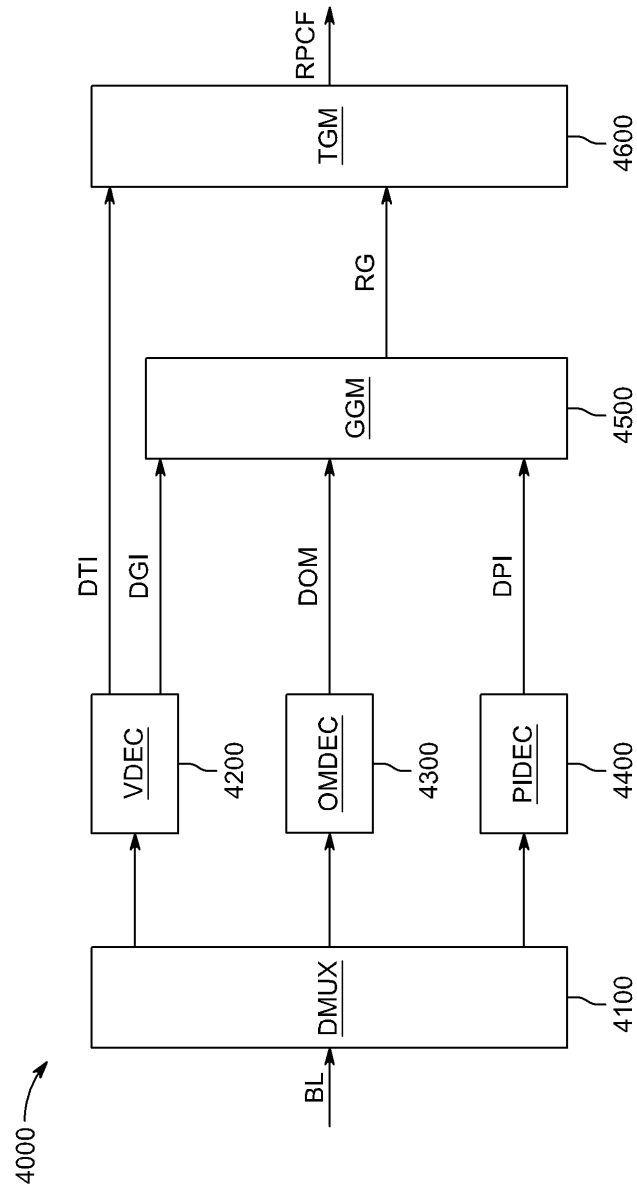


ФИГ. 2

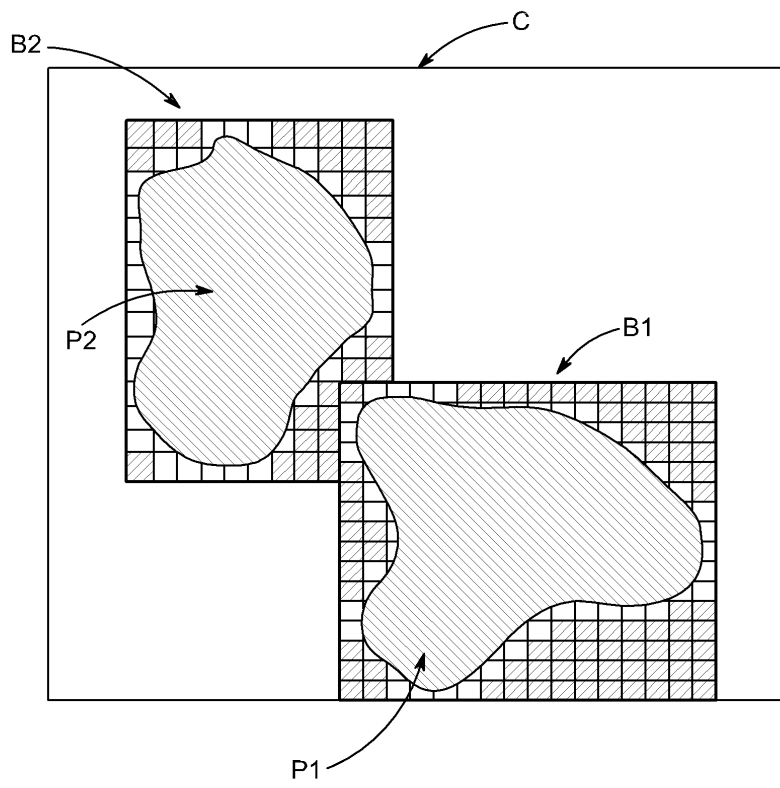
2



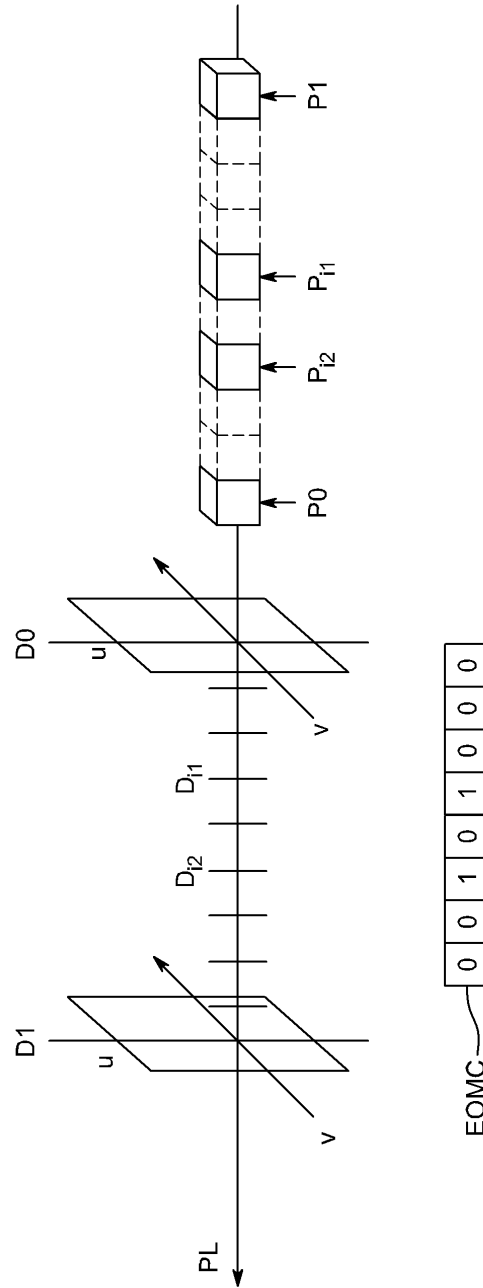
ФИГ. 3



ФИГ. 4

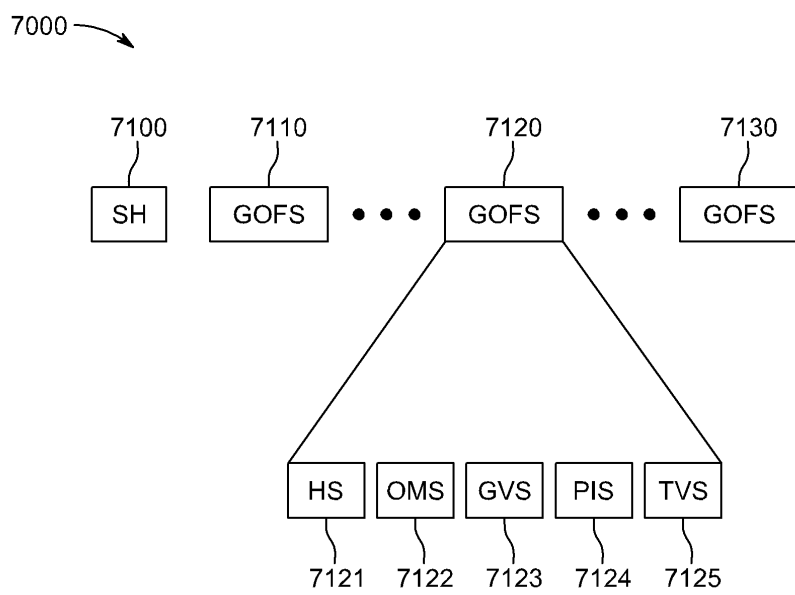


ФИГ. 5

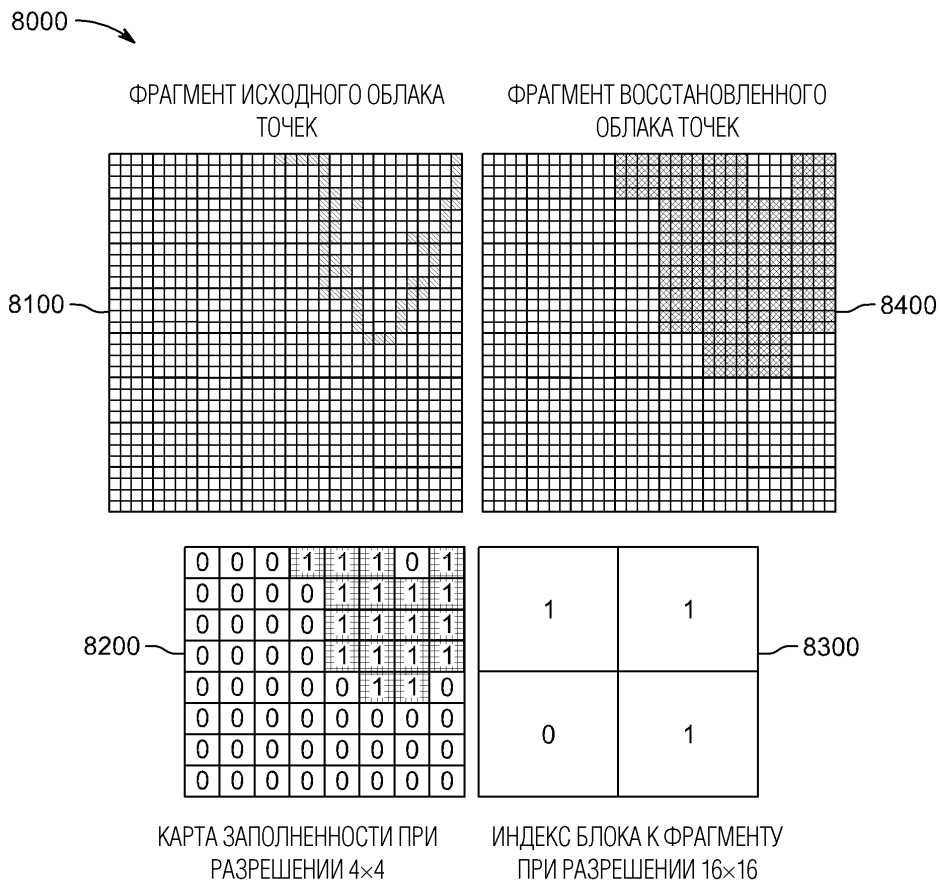


ФИГ. 6

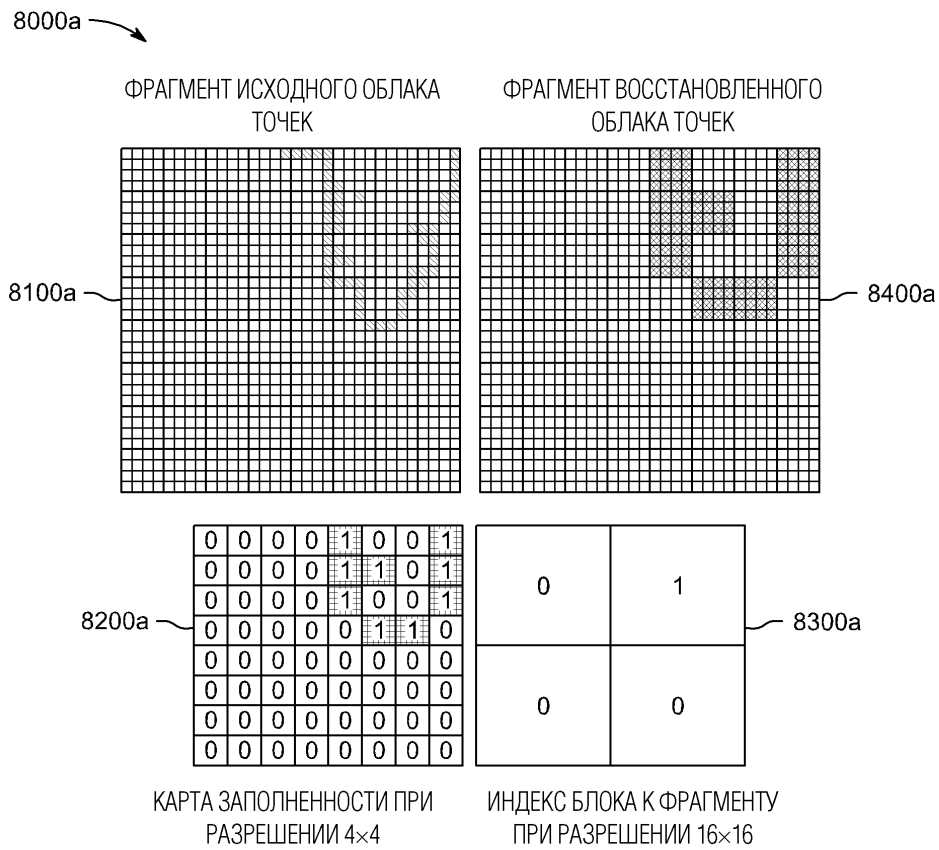
6/9



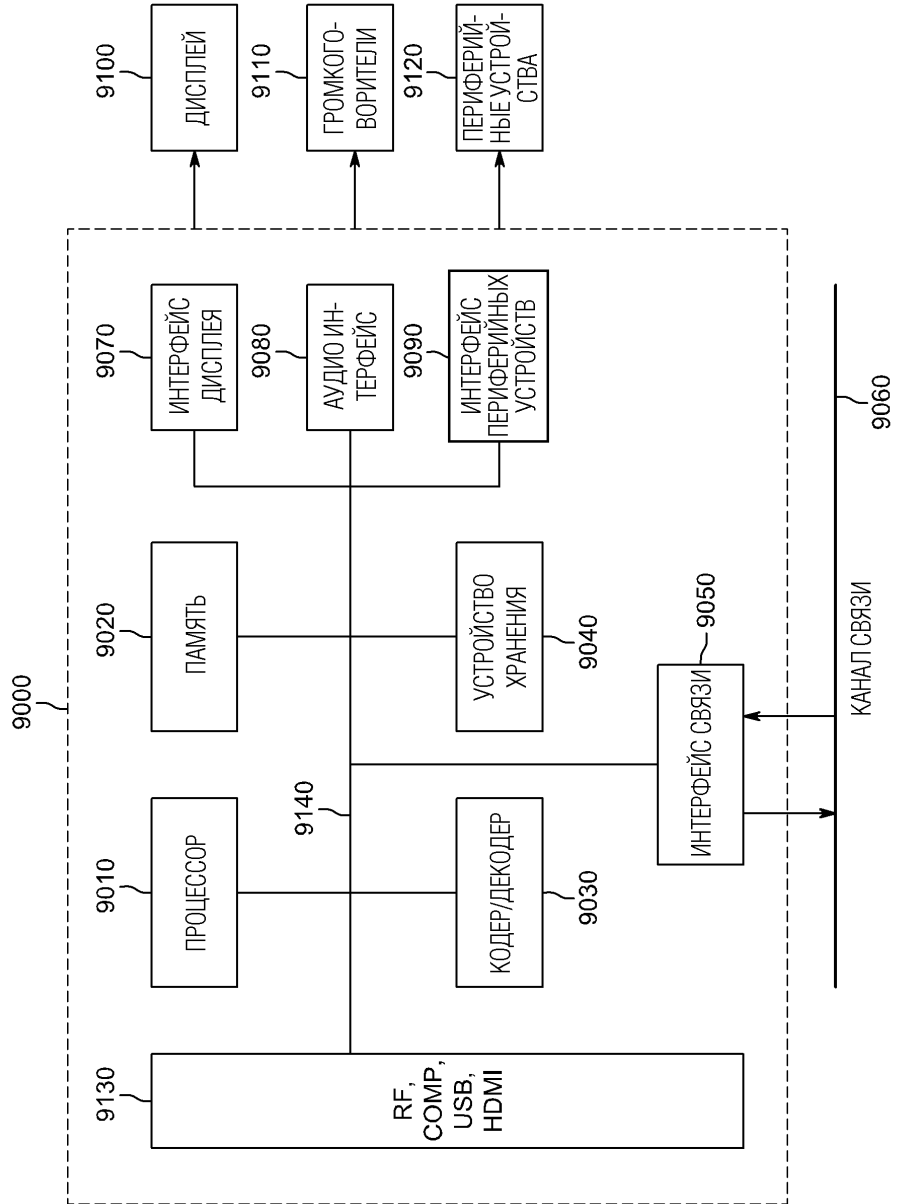
ФИГ. 7



ФИГ. 8



ФИГ. 8a



ФИГ. 9