



(51) МПК
H04N 19/503 (2014.01)
H04N 19/593 (2014.01)
H04N 19/91 (2014.01)

**ФЕДЕРАЛЬНАЯ СЛУЖБА
 ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ**

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
H04N 19/503 (2018.08); H04N 19/593 (2018.08); H04N 19/91 (2018.08)

(21)(22) Заявка: **2016152660, 21.07.2015**

(24) Дата начала отсчета срока действия патента:
21.07.2015

Дата регистрации:
05.07.2019

Приоритет(ы):
 (30) Конвенционный приоритет:
21.07.2014 GB 1412937.3

(43) Дата публикации заявки: **22.08.2018** Бюл. № 24

(45) Опубликовано: **05.07.2019** Бюл. № 19

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: **21.02.2017**

(86) Заявка РСТ:
EP 2015/025053 (21.07.2015)

(87) Публикация заявки РСТ:
WO 2016/012105 (28.01.2016)

Адрес для переписки:
**191036, Санкт-Петербург, а/я 24,
 "НЕВИНПАТ"**

(72) Автор(ы):
КАЛЕВО Осси (FI)

(73) Патентообладатель(и):
Гурулуджик Микросистемс Ой (FI)

(56) Список документов, цитированных в отчете о поиске: **D. SALOMON et al. "Data Compression", опубл. 20.09.2009 на 1118 страницах [найдено 24.12.2018], размещено в Интернет по адресу URL: <http://en.pudn.com/Download/item/id/769449.html>. EP 2723071 A1, 23.04.2014. EP 2611158 A1, 03.07.2013. US 2014/0173674 A1, 19.06.2014. RU 2236046 C2, 10.09.2004.**

(54) КОДЕР, ДЕКОДЕР И СПОСОБ

(57) Реферат:

Изобретение относится к вычислительной технике. Технический результат заключается в повышении эффективности кодирования данных. Кодер для кодирования входных данных для формирования соответствующих кодированных данных, выполненный с возможностью обработки входных данных и кодирования исходных значений по меньшей мере их части с использованием по меньшей мере одного алгоритма дельта-кодирования с получением значений разностей, которые выражены с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений

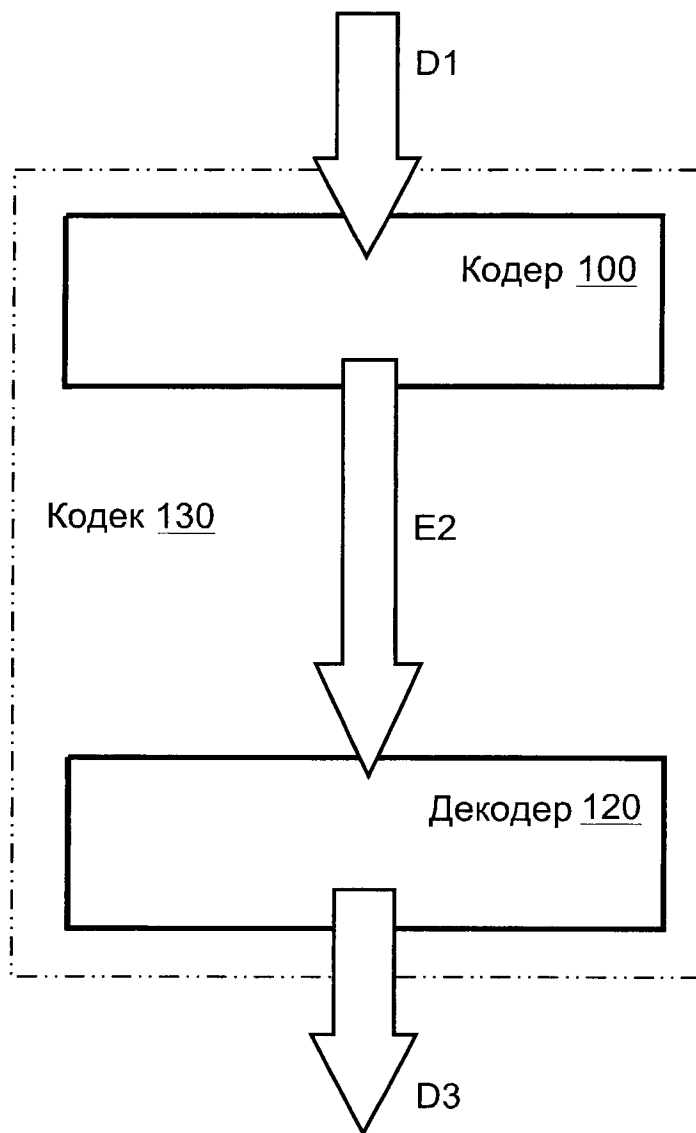
исходных значений, а также с возможностью формирования одного или более предикторов для использования при кодировании одной или более последующих частей входных данных, а также с возможностью кодирования данных, сформированных при помощи по меньшей мере одного алгоритма дельта-кодирования, такого как ODelta, DDelta, IDelta, PDelta, и одного или более предикторов с применением по меньшей мере одного алгоритма энтропийного кодирования, при этом один или более предикторов включают по меньшей мере одно из следующего: один или более временных

RU 2 693 902 C2

RU 2 693 902 C2

предикторов; один или более локальных пространственных предикторов, подвергаемых квантованию; и один или более локальных

пространственных предикторов, использующих заранее вычисленные значения. 6 н. и 29 з.п. ф-лы, 8 ил., 4 табл.



Фиг.5

RU 2693902 C2

RU 2693902 C2



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.
H04N 19/503 (2014.01)
H04N 19/593 (2014.01)
H04N 19/91 (2014.01)

(12) ABSTRACT OF INVENTION

(52) CPC

H04N 19/503 (2018.08); H04N 19/593 (2018.08); H04N 19/91 (2018.08)(21)(22) Application: **2016152660, 21.07.2015**(24) Effective date for property rights:
21.07.2015Registration date:
05.07.2019

Priority:

(30) Convention priority:
21.07.2014 GB 1412937.3(43) Application published: **22.08.2018 Bull. № 24**(45) Date of publication: **05.07.2019 Bull. № 19**(85) Commencement of national phase: **21.02.2017**(86) PCT application:
EP 2015/025053 (21.07.2015)(87) PCT publication:
WO 2016/012105 (28.01.2016)Mail address:
191036, Sankt-Peterburg, a/ya 24, "NEVINPAT"

(72) Inventor(s):

KALEVO Ossi (FI)

(73) Proprietor(s):

Gurulogic Microsystems Oy (FI)**(54) ENCODER, DECODER AND METHOD**

(57) Abstract:

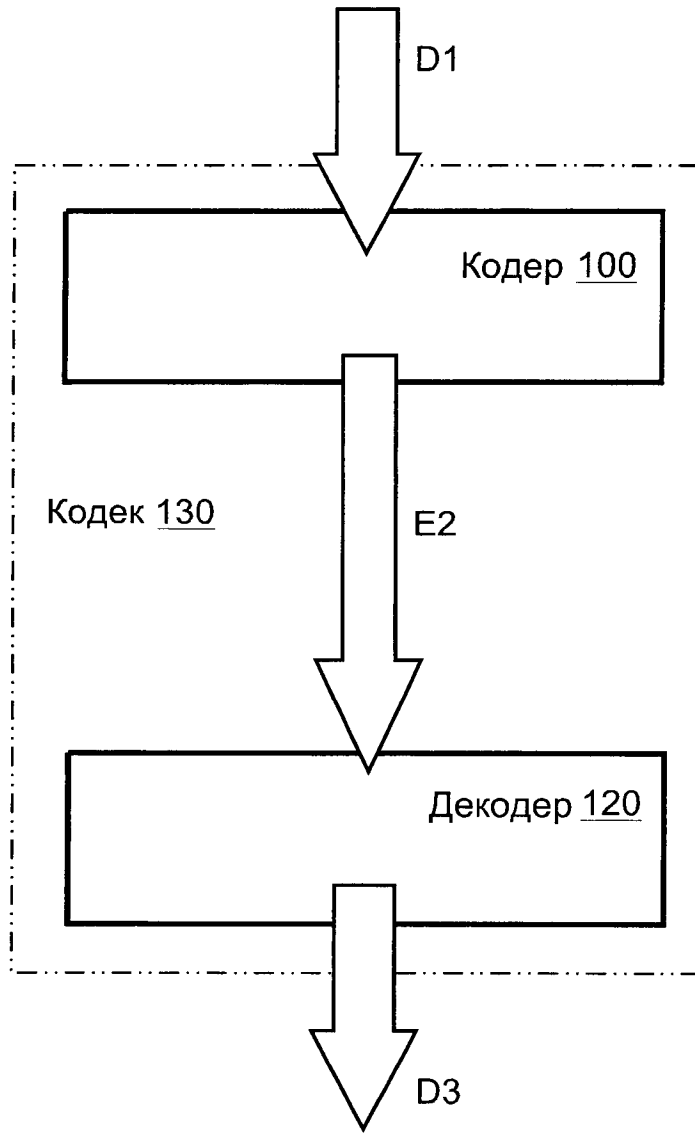
FIELD: calculating; counting.

SUBSTANCE: invention relates to the computer equipment. Encoder for encoding input data to generate corresponding encoded data, configured to process input data and encode original values of at least a portion thereof using at least one delta encoding algorithm with obtaining values of differences, which are expressed using a range of values, which is not increased compared to the range of values of initial values, as well as with possibility of generating one or more predictors for use in encoding one or more subsequent parts of input data, as well as with possibility of

encoding data generated by means of at least one delta-coding algorithm, such as ODelta, DDelta, IDelta, PDelta, and one or more predictors using at least one entropy coding algorithm, wherein one or more predictors include at least one of: one or more time predictors; one or more local spatial predictors subjected to quantisation; and one or more local spatial predictors using predetermined values.

EFFECT: technical result is higher efficiency of encoding data.

35 cl, 8 dwg, 4 tbl



Фиг.5

Область техники

Настоящее изобретение относится к способам кодирования данных, например, к способу кодирования данных при помощи дельта-кодирования с применением одного или более предикторов (предсказателей). Также настоящее изобретение относится к способам декодирования данных, например, к способам декодирования данных при помощи дельта-декодирования с применением одного или более предикторов. Также настоящее изобретение относится к системам, приспособлениям и устройствам для реализации упомянутых выше способов.

Также настоящее изобретение относится к компьютерным программным продуктам, включающим машиночитаемый носитель, на котором хранят машиночитаемые инструкции, при этом машиночитаемые инструкции могут быть исполнены на компьютеризованном устройстве, включающем процессорную аппаратуру, для исполнения упомянутых выше способов.

Предпосылки создания изобретения

На существующем уровне техники, как правило, во многих видеокодеках, например, в кодеках MPEG-4, H.264, VC-1, HEVC и VP9, для предсказания движения блоков в изображении могут использоваться предшествующие кадры (следует отметить, что перечисленные примеры кодеков включают зарегистрированные фирменные наименования). Предсказание движения и компенсация движения может выполняться поблочно для каждого кадра видеоизображения. Аналогично, при кодировании заданного блока данных или пакета данных, в целях повторного использования уже закодированных блоков данных или пакетов данных, могут применяться методы дедупликации или методы обработки с использованием баз данных. Дельта-кодирование может применяться для снижения энтропии информационных символов, присутствующих в видеоконтенте, или контенте сходного типа. О-дельта-кодирование, которое будет более подробно описано ниже, также, опционально, применяют для еще большего снижения энтропии информационных символов. Кроме того, О-дельта-кодирование позволяет снизить энтропию отдельных битов. Способы, относящиеся к О-дельта-кодированию, описаны более подробно в приложении 1. В этих двух способах, аналогичных DPCM-модуляции, то есть, в дельта-кодировании и О-дельта-кодировании, при формировании кодированных значений данных для энтропийного кодера используют предшествующие значения данных.

В настоящее время объемы хранимых и передаваемых данных быстро растут, и соответственно, растет потребность в сжатии данных, то есть, нужны новые, более совершенные методы, которые позволят повысить эффективность сжатия данных. Данные, например, могут захватываться одним или более датчиков, например, это могут быть данные изображений, видеоданные, аудиоданные, данные измерений или различные типы двоичных данных, ASCII-данных и т.п. Также могут применяться комбинации из захваченных данных с датчиков и абстрактных данных.

На существующем уровне техники имеются множество различных способов кодирования, которое могут применяться для кодирования данных, однако ни один из них не обеспечивает достаточно высокой степени сжатия для всех возможных типов данных. При кодировании заданного текущего канала, кадра, блока данных или пакета данных часто необходимо кодировать различные каналы или кадры данных, к примеру, каналы различных данных, каналы изображения, аудиоканалы, параллельные данные изображений, отдельные изображения в видеоданных, отдельные пакеты в аудиоданных, 3D изображения, 3D аудиоданные или другие аналогичные данные, по отдельности, но с использованием при этом информации об уже кодированных каналах, кадрах, блоках

данных или пакетах данных. Существующие способы кодирования данных недостаточно универсальны для работы с входными данными, содержащими в себе столь разнообразные информационные структуры.

Аналогично, уже кодированная пространственная информация может при кодировании быть использована более эффективно, чем в существующих способах, например, таких как DPCM-модуляция (<http://en.wikipedia.org/wiki/DPCM>). Существует также потребность в простых, но в тоже время эффективных способах кодирования и декодирования, позволяющих выполнять кодирование данных как без потерь, так и с потерями. Известен способ О-дельта-кодирования, который описан более подробно ниже в приложении 1, однако необходимо более эффективное применение способов О-дельта-кодирования, к примеру, необходимо модифицировать существующие способы дельта-кодирования таким образом, чтобы дельта-кодирование на основе DPCM-модуляции допускало применение также и с различными предикторами. Такой подход позволит обеспечить большее снижение энтропии для отрицательных и положительных разностей, или сумм, значений для символов, чем традиционные существующие способы дельта-кодирования, как, например, DPCM-модуляция.

На сегодняшний день не известны способы, которые бы сочетали в себе оба этих свойства, а именно, повторное использование данных за счет применения современных методов предсказания и эффективное снижение энтропии в данных остатка с квантованием или без квантования за счет передачи и хранения только информации о выбранном способе кодирования, например, для каждого кадра, канала, блока данных или пакета данных, и кодированных значений остатка, то есть без передачи векторов движения, символов выбора или указателей баз данных. В некоторых случаях кодирование остатка не требуется, по следующим причинам:

- (i) предсказание является абсолютно точным;
- (ii) остаток одинаков для всех значений данных, и достаточного одного значения для его передачи; или
- (iii) остаток, с квантованием или без квантования, не превышает порога погрешности, заданного на основе некоторого параметра качества.

Сущность изобретения

Цель настоящего изобретения - предложить усовершенствованный способ кодирования данных на основе применения алгоритмов дельта-кодирования. Еще одна цель настоящего изобретения - предложить усовершенствованный кодер для кодирования данных на основе применения алгоритмов дельта-кодирования.

Еще одна цель настоящего изобретения - предложить усовершенствованный способ декодирования данных на основе применения алгоритмов, обратных дельта-кодирования.

Еще одна цель настоящего изобретения - предложить усовершенствованный декодер для декодирования данных на основе применения алгоритмов, обратных дельта-кодированию.

В соответствии с первым аспектом настоящего изобретения предложен кодер для кодирования входных данных (D1) для формирования соответствующих кодированных данных (E2), выполненный с возможностью обработки входных данных (D1) и кодирования исходных значений по меньшей мере их части с использованием по меньшей мере одного алгоритма дельта-кодирования с получением значений разностей, которые выражены с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений исходных значений, а также с возможностью формирования одного или более предикторов для использования при кодировании одной или более

последующих частей входных данных (D1), при этом кодер также выполнен с возможностью кодирования данных, сформированных при помощи упомянутого по меньшей мере одного алгоритма дельта-кодирования (ODelta, DDelta, IDelta, PDelta), и упомянутого одного или более предикторов с применением по меньшей мере одного алгоритма энтропийного кодирования для формирования кодированных данных (E2), при этом упомянутые один или более предикторов включают по меньшей мере одно из следующего:

- (i) один или более временных предикторов;
- (ii) один или более локальных пространственных предикторов, подвергаемых квантованию; и
- (iii) один или более локальных пространственных предикторов, использующих заранее вычисленные значения.

Настоящее изобретение обладает тем преимуществом, что применение дельта-кодирования, формирование одного или более локальных пространственных предикторов, использование различных способов передачи дельта-значений и энтропийное кодирование, в комбинации, позволяют обеспечить высокоэффективное кодирование данных.

«Локальный пространственный предиктор», опционально, может кратко именоваться «локальным предиктором» или «пространственным предиктором».

Упомянутые заранее вычисленные значения могут представлять собой любые значения, которые могут быть определены заранее, до обработки данных на основе их пространственного местоположения; то есть, они не являются локальными пространственными предикторами, используемыми в О-дельта-кодировании, которое описано в приложении 1. Примеры подобных заранее вычисленных значений включают предыдущий блок слева или сверху относительно текущего кодируемого блока данных, сорасположенный блок в предыдущем канале, ракурсе или кадре, или внутреннее предсказание движения в заданном кадре данных.

Предсказание, в соответствии с предложенным способом, как правило, определяют заранее один раз и для всего. То есть, значение предсказания для заданного блока определяют заранее один раз при помощи временного или пространственного предсказания и для всего. Помимо упомянутых выше блоков, каналов, ракурсов и кадров, предсказание движения также удобно рассматривать как один из видов временного предсказания. Предсказание движения может также выполняться пространственно, и в таком случае оно будет относиться к пространственному предиктору, описанному выше в пункте (iv). То есть предсказания (оценки) движения не обязательно должны отличаться друг от друга во времени. Пространственное предсказание в настоящем документе отличается от «локального пространственного предсказания», которое применяют, пример, в способе О-дельта-кодирования, описанном в приложении 1. Когда применяют «локальное пространственное предсказание», предсказание получают в ходе выполнения способа, а значения предиктора не доступны до тех пор, пока не будут обработаны предшествующие значения. Это отличается по сравнению с временным предсказанием и другими типами пространственного предсказания. Опционально, предсказание может быть получено в ходе выполнения процедуры кодирования, например, как это происходит в процедуре О-дельта-кодирования (см. приложение 1). В таком случае используют локальные пространственные предикторы, определенные в О-дельта-кодировании.

Опционально, в кодере упомянутый по меньшей мере один алгоритм дельта-кодирования реализуют как:

(а) использование схемы обработки данных для применения ко входным данным (DA1) одной из форм разностного и/или суммирующего кодирования, с целью получения одной или более соответствующих кодированных последовательностей; и

5 (b) использование схемы обработки данных для выполнения, над одной или более соответствующими кодированными последовательностями, операции циклического обращения вокруг максимального значения и/или циклического обращения вокруг минимального значения, с целью формирования кодированных выходных данных (DA2 или DA3) (=ODelta).

Опционально, упомянутый алгоритм дельта-кодирования может быть использован
10 описанным ниже образом. Если разность между заданным исходным значением и соответствующим предсказанным значением всегда равна нулю или положительна или если разность между заданным исходным значением и соответствующим предсказанным значением равна нулю или положительна с применением квантования или без него, то в этом случае могут формироваться и передаваться только знаки, например знаковые
15 биты, дельта-значений вместе с информацией об алгоритме кодирования. То есть в подобных случаях циклическое обращение, которое применяют в алгоритме О-дельта-кодирования, вообще не требуется, поскольку и без циклического обращения значения всегда будут попадать в заданный диапазон количества бит. Алгоритмы, которые применяют подобным образом, получили соответствующие наименования, например,
20 способ I-дельта-кодирования (Incremental Delta, инкрементное дельта-кодирование) и способ D-дельта-кодирования (Decremental Delta, декрементное дельта-кодирование). Таким образом, в способе I-дельта-кодирования получают только положительные дельта-значения, а в способе D-дельта-кодирования получают только отрицательные дельта-значения, однако зачастую, предпочтительно, знак, то есть знаковый бит,
25 меняют. Замену знака выполняют только для значений D-дельта-кодирования и только при их передаче. В качестве результата, в обоих из описанных выше способов, очевидно, могут быть получены наборы постоянных значений, включая нулевые символы/значения, причем постоянные значения могут быть следствием квантования, или же квантование может не применяться; причем, если возникает такая ситуация и
30 квантование применяется, это означает, что разность между дельта-значениями не превосходит шага квантования.

Также, опционально, дельта-значения могут использоваться совместно с
пьедестальным значением. Это означает, что если встречаются и отрицательные, и
35 положительные дельта-значения, однако их абсолютные значения малы по сравнению с динамическим диапазоном и битовой глубиной кодируемых данных, то иногда выгодней передавать пьедестальное значение изменения, а именно, наибольшее отрицательное изменение, квантованное или без квантования. После этого могут передаваться только значения положительных изменений, например, как в упомянутом
40 выше способе I-дельта-кодирования. Способ, в котором применяют пьедестальное значения, для удобства называют «способом Р-дельта-кодирования». Часто, когда применяют способ Р-дельта-кодирования, динамический диапазон или битовая глубина данных могут быть сокращены сильнее, чем в случае применения исходного алгоритма О-дельта-кодирования, который описан в приложении 1. То есть данные могут быть
45 представлены меньшим количеством бит, поскольку максимальное значение данных будет меньше, и следовательно, разность между наибольшим и наименьшим из возможных значений будет также меньше. Это преимущество достижимо, поскольку в исходном способе О-дельта-кодирования (см. приложение 1) всегда необходимо формирование и передача динамического диапазона встречающихся значений, а именно

диапазона передаваемых значений ODelta.

Описанные выше способы I-дельта-кодирования, D-дельта-кодирования и P-дельта-кодирования, дают преимущества при применении в тех вариантах осуществления настоящего изобретения, где используемое предсказание представляет собой, например, предшествующий блок, канал, изображение или некоторый другой набор значений, определенных и объявленных до применения этих способов. Соответственно, разность между кодируемым значением и предсказанным значением может быть легко определена и оставаться впоследствии неизменной, и, следовательно, предсказание этого значения или разности совсем не зависят от остальных кодируемых значений. Методы предсказаний, в которых используют так называемые «временные предикторы» или «пространственные предикторы», особенно хорошо подходят для применений с квантованием, поскольку квантование значений разности оказывает влияние только на отдельные заданные декодируемые значения данных, и соответственно, ошибка, возникающая вследствие квантования, не накапливается в других декодируемых значениях данных. Квантование может использоваться также и с локальными пространственными предикторами, однако в этом случае, чтобы исключить накопление ошибки, в алгоритме при предсказании следующего значения должна быть учтена ошибка квантования. Известные типы дельта-кодирования реализуют при помощи вычитания, из заданного текущего значения, значения предсказания, которое получают с использованием локального пространственного предиктора, при этом разность этих значений может в результате давать как положительные, так и отрицательные значения, и, следовательно, вместе со значением разности всегда необходимо передавать знак. В O-дельта-кодировании (см. приложение 1) может использоваться как разность, так и сумма текущего значения и предсказанного значения, и при этом передачу значений выполняют с циклическим обращением этих значений, то есть, в результате все значения всегда остаются положительными.

Как упоминалось выше, способы I-дельта-кодирования, D-дельта-кодирования и P-дельта-кодирования отличаются как от существующих способов дельта-кодирования, так и от описанного в приложении 1 способа O-дельта-кодирования, где предсказанное значение передают с циклическим обращением. Также в способе O-дельта-кодирования, который описан в приложении 1, предикторы ограничены локальным пространственным предиктором только с одним значением, который всегда используют без квантования. Однако такие ограничения не являются обязательными для вариантов осуществления настоящего изобретения, в которых могут также использоваться другие типы предикторов и/или квантования, и даже, опционально, в комбинации с упомянутым выше способом O-дельта-кодирования. Может использоваться также исходный способ дельта-кодирования, в котором применяют циклическое обращение из способов I-дельта-кодирования, D-дельта-кодирования или P-дельта-кодирования, поскольку в некоторых случаях такое решение эффективно, а в некоторых случаях такое решение даже более эффективно, чем непосредственно упомянутые способы I-дельта-кодирования, D-дельта-кодирования и P-дельта-кодирования. Причина заключается в том, что в способе O-дельта-кодирования с использованием циклического обращения, при помощи одного и того же циклически обрабатываемого значения/символа, могут быть представлены как положительные, так и отрицательные разности. В исходном способе O-дельта-кодирования, где применяют циклическое обращение, его выполняют таким образом, что положительные и отрицательные значения остаются впоследствии различимыми, на основе диапазона значений и предсказанного значения, тогда как в упомянутых способах I-дельта-кодирования, D-дельта-кодирования и P-дельта-

кодирования передают только значения нулевой, положительной или отрицательной разности, и, соответственно, они не требуют дополнительного различения. Опционально, часто, вместе с информацией о выборе способа передают также информацию об использованном предсказании, и возможно, об использованном квантовании.

5 Предпочтительно, информацию о квантовании часто передают для всей последовательности данных за один раз, с использованием одного значения квантования или одного параметра качества. Примерами таких способов являются, например, способы, получившие наименования IDeltaBlockFromChannelO, PDeltaChannelR_2, DDeltaFrame_4, ODeltaBlockMode и DDeltaPacketPrevQ70. В первом из этих способов, то
10 есть в способе IDeltaBlockFromChannelO, передают только положительные значения разности относительно значений канала 0 в заданном соответствующем блоке, то есть, каждое из значений данных в текущем кодируемом блоке данных больше или равно значению в соответствующем местоположении в канале 0 блока, который используют в качестве предсказания.

15 Во втором из этих способов, PDeltaChannelR_2, передают предсказательное значение, за которым следуют положительные значения разности, а именно, разности между текущим каналом и каналом R, которые при этом квантуют с коэффициентом 2. Способ, обозначенный как DDeltaFrame_4, хорошо подходит, например, для постепенно затеняющихся изображений. Он дает отрицательные значения изменений, относительно
20 предыдущего кадра, квантованные с коэффициентом 4. Способ, обозначенный как ODeltaBlockMode, дает для текущей области блока циклически обращенные, вокруг заданного значения модуля, значения разности. Опционально, значение модуля может передаваться/ предоставляться для каждого из блоков независимо, или в качестве предсказанного значения модуля может использоваться значение модуля для всего
25 канала данных или значение модуля всего кадра данных. В способе DDeltaPacketPrevQ70 передают отрицательные значения данных для текущего пакета относительно предыдущего пакета того же размера, при этом квантование дельта-значений определяют с использованием коэффициента качества, равного 70.

Представленные выше примеры не имеют целью ограничить объем правовой защиты
30 настоящего изобретения, который задан приложенной формулой изобретения, поскольку для описания различных вариантов способа, соответствующего настоящему изобретению, могут использоваться также множество других аналогичных способов. Опционально, при использовании любых из этих способов, могут дополнительно передаваться пределы значений данных, которые чаще всего являются максимальными значениями, что
35 позволяет обеспечить наиболее эффективное сжатие передаваемых данных.

В настоящем изобретении определен и описан альтернативный метод эффективного использования существующих способов кодирования данных по частям, который не
40 влияет на функционирование собственно алгоритмов кодирования. Таким образом, способы, предложенные в описанных вариантах осуществления настоящего изобретения, например, могут эффективно применяться с другими способами кодирования или могут использоваться в качестве замены существующих способов кодирования. В способах, предложенных в настоящем изобретении, используют различные предикторы, и
опционально, различные квантователи, что позволяет снизить энтропию для любых энтропийных кодеров после выполнения способов, предложенных в вариантах
45 осуществления настоящего изобретения.

Опционально, кодер выполнен с возможностью применения по меньшей мере одного алгоритма квантования при кодировании входных данных (D1) с целью формирования кодированных данных (E2), при этом упомянутый по меньшей мере один алгоритм

квантования обеспечивает, в кодере, кодирование входных данных (D1) с потерями. Опционально, кодер выполнен с возможностью применения различных алгоритмов кодирования данных или различных структур данных, содержащихся во входных данных (D1).

5 Опционально, кодер выполнен с возможностью применения оптимизации «скорость - искажения» при поблочном кодировании входных данных (D1). Также, опционально, в кодере, оптимизацию «скорость - искажения» вычисляют для минимизации в кодере значения V следующего уравнения:

$$V = D + \lambda * R ,$$

10 где искажения (D) представляют собой сумму квадратов расхождений (SE) между входными данными (D1) и представлением входных данных (D1), кодированных в виде кодированных данных (E2) и декодированных в виде декодированных данных (D3), а скорость (R) представляет собой количество кодированных данных, измеренное, например, в битах.

15 Опционально, в кодере, по меньшей мере один алгоритм дельта-кодирования (ODelta, DDelta, IDelta, PDelta) и/или по меньшей мере один алгоритм энтропийного кодирования выполнен с возможностью применения по меньшей мере одного из следующего: способов DC-кодирования, способов скользящего кодирования, способов
20 многоуровневого кодирования, способов DCT-преобразования, способов линейного кодирования, способов кодирования с масштабированием, способов кодирования с использованием баз данных, диапазонного кодирования, кодирования Хаффмана, RLE-кодирования и SRLE-кодирования.

Опционально, кодер выполнен с возможностью кодирования входных данных (D1), включающих структуры данных, соответствующие по меньшей мере одному из
25 следующего: каналы YUV, каналы BGR. Также опционально, кодер выполнен с возможностью кодирования данных в упомянутых каналах в следующем порядке: Y, U, V или в следующем порядке G, B, R. Альфа-канал, а именно, канал прозрачности, может при этом кодироваться отдельно или вместе с остальными каналами.

30 Соответственно, кодируемые данные могут быть аудиоданными, и в этом случае значения амплитуды звука, которые могут иметь битовую глубину 8, 16 или 24 бита, могут кодироваться по отдельности, канал за каналом, или по нескольку каналов за один раз. Таким образом, независимо от того, являются ли кодируемые данные аудиоданными, изображениями, видеоданными, геномными данными, результатами измерений, текстовыми данными, двоичными данными или любыми другими данными,
35 битовая глубина кодируемых данных может меняться от 1 до, например, 256 битов для каждого элемента данных.

Опционально, кодер выполнен с возможностью включать, в кодированные данные (E2), данные, которые указывают на один или более алгоритмов кодирования, использованных кодером для кодирования входных данных (D1) с целью формирования
40 кодированных данных (E2).

В соответствии со вторым аспектом настоящего изобретения предложен способ использования кодера для кодирования входных данных (D1) с целью формирования соответствующих кодированных данных (E2), включающий:

45 (i) использование кодера (100) для обработки входных данных (D1) и для кодирования исходных значений по меньшей мере их части с использованием по меньшей мере одного алгоритма дельта-кодирования (ODelta, DDelta, IDelta, PDelta) с получением значений разностей, которые выражены с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений исходных значений;

(и) использование кодера (100) для формирования одного или более предикторов для использования при кодировании одной или более последующих частей входных данных (D1); и

5 (iii) использование кодера (100) для кодирования данных, сформированных при помощи упомянутого по меньшей мере одного алгоритма дельта-кодирования, и упомянутых одного или более предикторов путем применения по меньшей мере одного алгоритма энтропийного кодирования для формирования кодированных данных (E2), при этом упомянутые один или более предикторов включают по меньшей мере одно из следующего:

10 (i) один или более временных предикторов;

(ii) один или более локальных пространственных предикторов, подвергаемых квантованию;

(iii) один или более локальных пространственных предикторов, использующих заранее вычисленные значения.

15 Опционально, в предложенном способе, упомянутый по меньшей мере один алгоритм дельта-кодирования реализован как:

(a) использование схемы обработки данных для применения ко входным данным (DA1) одной из форм разностного и/или суммирующего кодирования, с целью получения одной или более соответствующих кодированных последовательностей; и

20 (b) использование схемы обработки данных для выполнения, над одной или более соответствующими кодированными последовательностями, операции циклического обращения вокруг максимального значения и/или циклического обращения вокруг минимального значения, с целью формирования кодированных выходных данных (DA2 или DA3) (= ODelta).

25 Опционально, способ O-дельта-кодирования (см. приложение 1), может применяться описанным ниже образом. Если разность между заданным исходным значением и соответствующим предсказанным значением всегда равна нулю или положительна или если разность между заданным исходным значением и соответствующим предсказанным значением равна нулю или положительна с применением квантования или без него, то
30 в этом случае могут формироваться и передаваться только знаки, а именно знаковые биты, дельта-значений вместе с информацией о способе кодирования. То есть в подобных случаях, циклическое обнуление вообще не требуется, поскольку значения всегда будут попадать в заданный диапазон количества бит и без циклического обнуления.

Способы, которые применяют описанным образом, могут для удобства называться, например, способом I-дельта-кодирования («инкрементное дельта-кодирование») и
35 способом D-дельта-кодирования («декрементное дельта-кодирование»). Таким образом, в способе I-дельта-кодирования получают только положительные дельта-значения, а в способе D-дельта-кодирования получают только отрицательные дельта-значения, однако зачастую, предпочтительно, знак, то есть знаковый бит, меняют. Оба из
40 описанных выше способов, очевидно, также всегда могут давать в результате постоянные значения, включая нулевые символы/значения, причем постоянные значения могут быть следствием квантования, или же квантование может не применяться; а именно, если возникает такая ситуация и квантование применяется, это означает, что разность между дельта-значениями не превосходит шага квантования.

45 При этом, опционально, дельта-значения могут использоваться совместно с пьедестальным значением. Это означает, что если встречаются и отрицательные, и положительные дельта-значения, однако их абсолютные значения малы по сравнению с динамическим диапазоном и битовой глубиной кодируемых данных, то иногда

выгодней передавать пьедестальное значение изменения, а именно, наибольшее отрицательное изменение, квантованное или без квантования. После этого могут передаваться только значения положительных изменений, например, как в упомянутом выше способе I-дельта-кодирования. Такой способ, в котором используют пьедестальные значения, может называться, например, «способом Р-дельта-кодирования». Часто, когда применяют способ Р-дельта-кодирования, динамический диапазон или битовая глубина данных могут быть сокращены сильнее, чем в случае применения исходного алгоритма О-дельта-кодирования, который описан в приложении 1. То есть, данные могут быть представлены меньшим количеством бит, поскольку максимальное значение данных будет меньше, и следовательно, разность между наибольшим и наименьшим из возможных значений будет также меньше. Это преимущество достижимо, поскольку в способе О-дельта-кодирования всегда необходимо формирование и передача динамического диапазона встречающихся значений, а именно диапазона передаваемых значений ODelta.

Описанные выше способы I-дельта-кодирования, D-дельта-кодирования и Р-дельта-кодирования дают преимущества при применении в тех решениях, где используемое предсказание представляет собой, например, предшествующий блок, канал, изображение или некоторый другой набор значений, определенных и объявленных до применения этих способов. Соответственно, разность между кодируемым значением и предсказанным значением может быть легко определена и оставаться впоследствии неизменной, и следовательно, предсказание этого значения или разности совсем не зависят от остальных кодируемых значений. Такие методы предсказания особенно хорошо подходят для применений с квантованием, поскольку в таких случаях квантование значений разности оказывает влияние только одно отдельное декодируемое значения данных, и соответственно, ошибка, возникающая вследствие квантования, не накапливается в других декодируемых значениях данных.

Опционально, часто вместе с информацией о выборе способа передают также информацию об использованном предсказании, и возможно, об использованном квантовании. Предпочтительно, информацию о квантовании часто передают для всей последовательности данных за один раз, с использованием одного значения квантования или параметра качества. Примерами таких способов являются, например, способы, которые для удобства именуют следующим образом: IDeltaBlockFromChannelO, PDeltaChannelR_2, DDeltaFrame_4, ODeltaBlockMode и DDeltaPacketPrevQ70. В первом из этих способов, то есть в способе IDeltaBlockFromChannelO, передают только положительные значения разности относительно значений канала 0 в заданном соответствующем блоке. То есть каждое из значений данных в текущем кодируемом блоке данных больше или равно значению в соответствующем местоположении в канале 0 блока, который используют в качестве предсказания.

Во втором из этих способов, PDeltaChannelR_2, передают пьедестальное значение, за которым следуют положительные значения разности, а именно, разности между текущим каналом и каналом R, которые при этом квантуют с коэффициентом квантования, равным 2. Способ DDeltaFrame_4 хорошо подходит, например, для постепенно затеняющихся изображений. Он дает отрицательные значения изменений, относительно предыдущего кадра, квантованные с коэффициентом квантования, равным 4. Способ ODeltaBlockMode дает для текущей области блока циклически обращенные, вокруг заданного значения модуля, значения разности. Опционально, значение модуля может передаваться/предоставляться для каждого из блоков независимо, или в качестве предсказанного значения модуля может использоваться значение модуля для всего

канала данных или значение модуля всего кадра данных.

В способе DDeltaPacketPrev передают отрицательные значения данных для текущего пакета относительно предыдущего пакета того же размера.

Опционально, способ включает выполнение кодера с возможностью применения по меньшей мере одного алгоритма квантования при кодировании входных данных (D1) с целью формирования кодированных данных (E2), при этом упомянутый по меньшей мере один алгоритм квантования обеспечивает, в кодере, кодирование входных данных (D1) с потерями. Опционально, способ включает выполнение кодера с возможностью применения различных алгоритмов кодирования данных или различных структур данных, содержащихся во входных данных (D1).

Опционально, способ включает выполнение кодера (100) с возможностью применения оптимизации «скорость - искажения» при поблочном, по пакетному, поканальном, поракурсном или покадровом кодировании входных данных (D1). Также, опционально, в предложенном способе, оптимизацию «скорость - искажения» вычисляют для минимизации в кодере (100) значения V следующего уравнения:

$$V = D + \lambda * R ,$$

где искажения (D) представляют собой, сумму квадратов расхождений (SE) между входными данными (D1) и представлением входных данных (D1), кодированных в виде кодированных данных (E2) и декодированных в виде декодированных данных (D3), а скорость (R) представляет собой количество кодированных данных, измеренное, например, в битах.

Опционально, в предложенном способе по меньшей мере один алгоритм дельта-кодирования (ODelta, DDelta, IDelta, PDelta) и/или по меньшей мере один алгоритм энтропийного кодирования выполнен с возможностью применения по меньшей мере одного из следующего: способов DC-кодирования, способов скользящего кодирования, способов многоуровневого кодирования, способов DCT-преобразования, способов линейного кодирования, способов кодирования с масштабированием, способов кодирования с использованием баз данных, диапазонного кодирования, кодирования Хаффмана, RLE-кодирования и SRLE-кодирования.

Опционально, способ включает выполнение кодера (100) с возможностью кодирования входных данных (D1), включающих структуры данных, соответствующие по меньшей мере одному из следующего: каналы YUV, каналы BGR. Также, опционально, предложенный способ включает выполнение кодера (100) с возможностью кодирования данных каналов в следующем порядке: Y, U, V или в следующем порядке G, B, R. Альфа-канал, а именно, канал прозрачности, может при этом кодироваться отдельно или вместе с остальными каналами. Соответственно, кодируемые данные могут быть аудиоданными, и в этом случае значения амплитуды звука, которые могут иметь битовую глубину 8, 16 или 24 бита, могут кодироваться по отдельности, канал за каналом, или по нескольким каналам за один раз. Таким образом, независимо от того, являются ли кодируемые данные аудиоданными, изображениями, видеоданными, геномными данными, результатами измерений, текстовыми данными, двоичными данными или любыми другими данными, битовая глубина кодируемых данных может меняться от 1 до, например, 256 бит для каждого элемента данных.

Опционально, предложенный способ включает выполнение кодера (100) с возможностью включать, в кодированные данные (E2), такие данные, которые указывают на один или более алгоритмов кодирования, используемых кодером для кодирования входных данных (D1) с целью формирования кодированных данных (E2).

В соответствии с третьим аспектом настоящего изобретения предложен декодер для

декодирования кодированных данных (E2) с целью формирования соответствующих декодированных данных (D3), выполненный с возможностью исполнения алгоритмов, обратных алгоритмам кодирования, реализованным в кодере по первому аспекту настоящего изобретения.

5 То есть, предложен декодер для декодирования кодированных данных (E2) для формирования соответствующих декодированных данных (D3), выполненный с возможностью обработки кодированных данных (E2) с применением к ним по меньшей мере одного алгоритма энтропийного декодирования для формирования обработанных данных, а также с возможностью использования одного или более предикторов в
10 сочетании с по меньшей мере одним алгоритмом дельта-декодирования (обратный ODelta, обратный DDelta, обратный IDelta, обратный PDelta) для декодирования обработанных данных и формирования декодированных данных (D3),

при этом обработанные данные включают значения разностей, которые выражены с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений исходных данных, из которых были сформированы кодированные данные (E2),
15

при этом упомянутые один или более предикторов включают по меньшей мере одно из следующего:

- (i) один или более временных предикторов;
- 20 (ii) один или более локальных пространственных предикторов, подвергаемых квантованию;
- (iii) один или более локальных пространственных предикторов, использующих заранее вычисленные значения.

Упомянутые заранее вычисленные значения могут представлять собой любые значения, которые могут быть определены до выполнения обработки данных на основе пространственного местоположения; то есть, они не являются локальными пространственными предикторами, используемыми в О-дельта-кодировании, которое описано в приложении 1. Примеры подобных заранее вычисленных значений включают предыдущий блок слева или сверху относительно текущего кодируемого блока данных, сорасположенный блок в предыдущем канале, ракурсе или кадре, или внутреннее предсказание движения в заданном кадре данных.
25

Опционально, предложенный декодер выполнен с возможностью приема информации, указывающей на способ кодирования, и затем, выполнения предсказания согласно этому способу и вычисления декодируемых значений в соответствии с различными способами; другими словами, в декодере принимают информацию об использованном способе, и на основе этой информации может быть сделан вывод о том, следует ли использовать способ с пьедестальным значением (Р-дельта-кодирование), или о том, что будут встречаться только положительные значения разности (I-дельта-кодирование или D-дельта-кодирование), или о том, что будут встречаться отрицательные значения разности (D-дельта-кодирование), или о том, что используется регулярное циклическое обращение и предельные значения для различения между упомянутыми положительным и отрицательными значениями (О-дельта-кодирование).
35

Опционально, в работе, декодер принимает информацию о способе, затем выполняет предсказание согласно этому способу и вычисляет декодируемые значения в соответствии с различными способами; другими словами, принимают информацию об использованном способе, и на основе этой информации может быть сделан вывод о том, следует ли использовать способ с пьедестальным значением (Р-дельта-кодирование), или о том, что будут встречаться только положительные значения разности (I-дельта-
40

кодирование или D-дельта-кодирование), или о том, что будут встречаться отрицательные значения разности (D-дельта-кодирование), или о том, что используется регулярное циклическое обращение и предельные значения для различения между упомянутыми положительными и отрицательными значениями (O-дельта-кодирование).

5 Предсказание может быть определено один раз и оставаться неизменным, или оно может определяться в ходе выполнения процедуры, например, как это происходит в процедуре O-дельта-кодирования (см. приложение 1). Исходная процедура типа O-дельта-кодирования, в которой предсказание выполняют в ходе ее выполнения, называется локальным пространственным предсказанием ли локальным предсказанием, а новое предложенное решение, в котором предсказание блока определяют заранее и окончательно, называют «временным предсказанием» или «пространственным предсказанием». Помимо упомянутых выше блоков, каналов, ракурсов и кадров предсказание движения также удобно рассматривать в качестве «временного предсказания».

15 В соответствии с четвертым аспектом настоящего изобретения предложен способ декодирования кодированных данных (E2) в декодере с целью формирования соответствующих данных (D3), включающий выполнение, в декодере, способа, обратного способу по второму аспекту настоящего изобретения.

Способ декодирования кодированных данных (E2) в декодере (120) для формирования соответствующих данных (D3), включающий обработку кодированных данных (E2) с применением к ним по меньшей мере одного алгоритма энтропийного декодирования для формирования обработанных данных, а также использование одного или более предикторов в сочетании с по меньшей мере одним алгоритмом дельта-декодирования (обратный ODelta, обратный DDelta, обратный IDelta, обратный PDelta) для декодирования обработанных данных и формирования декодированных данных (D3), при этом обработанные данные включают значения разностей, которые выражены с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений исходных данных, из которых были сформированы кодированные данные (E2),

30 при этом упомянутые один или более предикторов включают по меньшей мере одно из следующего:

(i) один или более временных предикторов;

(ii) один или более локальных пространственных предикторов, подвергаемых квантованию;

35 (iii) один или более локальных пространственных предикторов, использующих заранее вычисленные значения.

Заранее вычисленные значения, как и локальные предикторы, описаны выше.

В соответствии с пятым аспектом настоящего изобретения предложен компьютерный программный продукт, включающий машиночитаемый носитель, на котором хранят машиночитаемые инструкции, при этом машиночитаемые инструкции могут быть исполнены на компьютеризованном устройстве, включающем процессорную аппаратуру, для исполнения способа по второму аспекту или по четвертому аспекту настоящего изобретения.

Нужно понимать, что отличительные признаки настоящего изобретения в пределах объема настоящего изобретения, заданного приложенной формулой изобретения, могут комбинироваться произвольным образом.

Описание чертежей

Далее, исключительно в качестве примера и со ссылками на приложенные чертежи,

будут описаны примеры осуществления настоящего изобретения, где:

Фиг. 1 представляет собой эскизную иллюстрацию примера верхнеуровневой структуры одного из вариантов осуществления настоящего изобретения.

5 Фиг. 2 представляет собой эскизную иллюстрацию примера канала из шести блоков в одном вариантах осуществления настоящего изобретения.

Фиг. 3 представляет эскизную иллюстрацию примера блока и его состава.

Фиг. 4 представляет собой эскизную иллюстрацию смежных значений данных, используемых для предсказания, в соответствии с одним из вариантов осуществления настоящего изобретения.

10 Фиг.5 представляет собой эскизную иллюстрацию кодера, декодера и кодека в соответствии с настоящим изобретением.

Фиг. 6-8 являются сопроводительными иллюстрациями для описания способов О-дельта-кодирования, приведенного в приложении 1.

15 На приложенных чертежах числа, выделенные подчеркиванием, используются для обозначения элементов, над которыми находится подчеркнутое число, или рядом с которыми оно расположено. Неподчеркнутые числовые обозначения относятся к объектам, указанным линией, которая соединяет подчеркнутое число и объект. Если число не выделено подчеркиванием и сопровождается связанной с ним стрелкой, это подчеркнутое число используется для обозначения общего элемента, на который

20 указывает стрелка.

Описание вариантов осуществления изобретения

В приведенном ниже описании вариантов осуществления настоящего изобретения использованы сокращения, проиллюстрированные в таблице 1:

25

30

35

40

45

Таблица 1: расшифровка сокращений, использованных при описании вариантов осуществления настоящего изобретения

Сокращение	Расшифровка
1D	Одномерный, например, в отношении сигнала или пакета данных
2D	Двумерный, например, в отношении сигнала или пакета данных
3D	Трёхмерный, например, в отношении сигнала или пакета данных
Блок	Множество элементов цифровых данных, то есть, фрагмент цифровых данных
CRC	Циклическая проверка по избыточности (Cyclic redundancy check)
Кодек	Кодер и декодер цифровых данных
БД	База данных в памяти с произвольным доступом или памяти в режиме «только для чтения»
DC	Постоянная составляющая изображения, то есть, среднее значение изображения, соответствующее средней яркости и представляющее собой наименьшую пространственную частоту, присутствующую в изображении
Дельта-кодирование	Дельта-кодирование – это способ хранения или передачи данных в форме разностей между следующими друг за другом значениям данных, вместо файлов с полными значениями данных
LAN	Локальная вычислительная сеть (Local Area Network)

Пакет	Элемент данных, содержащий, например, множество блоков данных
RAM	Память с произвольным доступом (Random Access Memory)
RD	«Скорость – искажения» (rate-distortion)
RLE	Кодирование длин серий (Run-Length Encoding)
ROI	Область интереса (Region of Interest)
ROM	Память «только для чтения» (Read Only Memory)
SRLE	Разделенное кодирование длин серий (Split Run-Length Encoding)
VLC	Кодирование переменной длины (Variable-Length Code)
XOR	Операция «исключающее или» (логическая функция)

Итак, варианты осуществления настоящего изобретения относятся к усовершенствованной структуре кодера и декодера и к соответствующим им усовершенствованным способам кодирования и декодирования данных. Варианты осуществления настоящего изобретения основаны на способах дельта-кодирования, например, на способе О-дельта-кодирования, который будет более подробно ниже и который может быть дополнительно усовершенствован в вариантах осуществления

настоящего изобретения. Способы дельта-кодирования сконфигурированы для задач кодирования аудио пакетов, блоков изображений, информационных пакетов Интернета, каналов, видеокадров или других данных с использованием множества различных

5 Способы кодирования в настоящем изобретении могут применяться как для кодирования без потерь, так и с потерями, при этом они включают три основных функциональных элемента:

(i) предсказание;

(ii) оператор О-дельта-кодирования, или аналогичный оператор, например, Р-дельта-кодирования, I-дельта-кодирования или D-дельта-кодирования, опционально, с

10 квантователем; и
(iii) энтропийное кодирование.

Способы кодирования в настоящем изобретении и соответствующие способы декодирования допускают применение в кодерах и декодерах соответственно, в соответствии с дальнейшим более подробным описанием на примере фиг. 5.

В приложении 1 можно найти описание оператора О-дельта-кодирования для применений, аналогичных DPCM-модуляции. В настоящем документе, в тексте, предшествующем приложению 1, такие операторы О-дельта-кодирования

20 модифицированы таким образом, чтобы допускать применение различных способов (локального) пространственного, временного или комбинированного предсказания. Способы, предложенные в настоящем изобретении, сконфигурированы таким образом, что они могут применяться для последовательности данных в целом, для отдельных кадров данных, для отдельных пакетов данных, для отдельных блоков данных или для отдельных пакетов данных и т.п. При этом также способы, предложенные в настоящем

25 изобретении, обеспечивают несколько отличающихся друг от друга способов кодирования, основанных на выбранных способах предсказания, выбранных операторах О-дельта-кодирования и выбранных способах кодирования и сжатия остатка. Опционально, в сочетании со способами, предложенными в настоящем изобретении, могут применяться и множество других способов кодирования, причем,

30 предпочтительно, эти способы применяют вместе с блочным кодером, описанном в приложении 2 и соответствующим патентному документу GB 2503295, который включен в настоящий документ путем ссылки, и блочным декодером, описанном в приложении 3 и соответствующим патентному документу GB 2505169, который также включен в настоящий документ путем ссылки. Например, при поблочном кодировании некоторого

35 канала данных, выбор наилучшего способа кодирования для применения в целях кодирования заданного блока данных, например, может выполняться с использованием оптимизации «скорость - искажения». Оптимизация «скорость - искажения» позволяет оптимизировать значение V в следующем уравнении:

$$40 \quad V = D + \lambda * R \quad (\text{Ур. 1})$$

где искажения (D), как правило, представляют собой сумму квадратов расхождений (SE) между исходными и декодированными значениями, а скорость (R) кодированных значений данных, как правило, выражена в битах. Вместе со способами, предложенных в настоящем изобретении, опционально, могут применяться множество других способов кодирования, например, способы DC-кодирования, способы скользящего кодирования,

45 способы многоуровневого кодирования, способы DCT-преобразования, способы линейного кодирования, способы кодирования с масштабированием, способов кодирования с использованием баз данных и др.

Способы, предложенные в настоящем изобретении, предпочтительно, применяют для отличающихся друг от друга структур данных, содержащихся в кодируемых входных данных (D1). К примеру, при помощи способов, предложенных в настоящем изобретении, опционально, кодируют каналы данных целиком, к примеру, канал яркости плоского изображения. Способы, предложенные в настоящем изобретении, являются простыми и могут быть без труда реализованы в различных устройствах и системах, например с использованием процессоров с ограниченным набором команд (reduced instruction set, RISC), которые на современном уровне техники применяют в низкоэнергетических портативных электронных устройствах, таких как мобильные телефоны, фото- или видеокамеры и т.п. Соответственно, результаты, которые позволяют получить способы, предложенные в настоящем изобретении, для канала в целом, легко сравнить с другими способами кодирования каналов, например, с каналом, все значения которого соответствуют черному цвету или заданному режиму, с каналом неподвижного изображения, с энтропийно кодированным исходным каналом, и кодированием канала при помощи блочного кодера. Выбор наилучшего способа кодирования канала, предпочтительно, выполняют при помощи оптимизации «скорость - искажения». Кодирование канала при помощи способа блочного кодирования означает, что канал данных кодирует поблочно с использованием различных способов кодирования, и в этом случае один и тот же способ кодирования не используют для всего канала данных в целом.

Способы, предложенные в настоящем изобретении, опционально, допускают применение при кодировании с потерями, а именно, при кодировании значения остатка с квантованием или без кодирования остатка, а также при кодировании без потерь, а именно, при нулевом остатке или при кодировании остатка без квантования. Все ранее кодированные и декодированные значения в кодере или в декодере, соответствующих настоящему изобретению, могут использоваться для предсказания текущих или будущих значений данных. Когда данные кодируют без потерь, применяя к ним способы кодирования, предложенные в настоящем изобретении, то при этом уже обработанные исходные значения, которые совпадают с декодированными значениями в случае кодирования без потерь, могут использоваться для предсказания текущих или будущих значений данных в кодере. Основными параметрами оператора О-дельта-кодирования являются highValue, lowValue и wrapValue (максимальное значение, минимальное значение и значение циклического обращения), при этом wrapValue равно по меньшей мере highValue - lowValue + 1. Более подробное описание этих параметров приведено в приложении 1. Значения highValue и lowValue могут быть определены для применения квантования. Например, исходные данные могут содержать значения от 0 до 255, однако может быть необходимо, чтобы конечный результат был квантован, к примеру, до значений от 0 до 78 (относительное квантование 78/255), с использованием выбранного коэффициента качества (например, равного 30, в случае, когда показатель качества измеряют в диапазоне от «одного» до «ста», где показатель качества «сто» соответствует сжатию без потерь). Могут также применяться смещения данных, до (предварительное смещение) или после (пост-смещение) применения оператора О-дельта-кодирования. При этом энтропийное кодирование, предпочтительно, выполняют после применения оператора О-дельта-кодирования, поскольку в противном случае снижение энтропии не сможет быть максимально эффективно использовано при кодировании данных.

Как уже упоминалось в отношении способов I-дельта-кодирования, D-дельта-кодирования и P-дельта-кодирования, они не требуют определения значения wrapValue, в отличие от способа О-дельта-кодирования (см. приложение 1). 0». Поэтому, в

описанных выше способах I-дельта-кодирования, D-дельта-кодирования и D-дельта-кодирования значения highValue' и lowValue', вместе с информацией о выбранном способе кодирования и опциональной информацией о квантовании, играют еще более важную роль.

5 В этих случаях значения highValue' и lowValue' относятся уже не к диапазону окончательных реальных значений данных, но к диапазону передаваемых значений разности. Значение lowValue' может также использоваться для определения пьедестального значения в способе P-дельта-кодирования, и в этом случае
10 максимальным передаваемым значением будет результат операции highValue' - lowValue', как таковой, или, возможно, сокращенный при помощи квантования. Квантование может быть определено при помощи делителя или параметра качества, или же как относительное изменение по сравнению с исходным динамическим диапазоном. Один из примеров подобной реализации приведен в предшествующем описании, в нем исходное значение highValue или наибольшая разность highValue' равны 255, при этом
15 диапазон значений ограничен, или сокращен, при помощи значения относительного квантования, равного 78/255. Параметр относительного квантования в данном примере передают, например, как значение 78, или иначе, как значение 0,3059 (то есть меньшее 78/255).

В качестве способа энтропийного кодирования, предпочтительно, выбирают
20 диапазонное кодирование или диапазонное SRLE-кодирование, однако могут использоваться и другие способы энтропийного кодирования, например, кодирование Хаффмана, RLE-кодирование или SRLE-кодирование. Когда применяют способ O-дельта-кодирования, описанный в приложении 1, значение предсказания, предпочтительно, всегда является предыдущим значением данных, а первое значение
25 данных должно быть инициализировано с помощью выбранного способа инициализации.

В способах, предложенных в настоящем изобретении, значения предсказания могут выбираться различным образом, то есть не обязательно используется только
30 предыдущее значение данных. В соответствии со способами, предложенными в настоящем изобретении, допускается использование предсказанного значения, которое является одиночным выбранным значением данных, или значения, которое вычислено на основе множества значений данных. К примеру, такое значение может вычисляться на основе двух или более предшествующих значений данных (1D), на основе двух или более предшествующих значений данных в окрестности этого значения (2D, 3D, ...), на основе значения или значений в предыдущем блоке данных или пакете данных, на
35 основе значения или значений в предыдущем канале или каналах данных, на основе значения или значения в предыдущем кадре или кадрах данных, на основе любых из упомянутых выше значений данных и т.п.

Когда значение предсказания для текущего значения данных вычислено, вычисляют разность или сумму исходного и предсказанного значения данных, и используют ее в
40 качестве значения OValue. Значение OValue может быть квантовано, или скопировано без изменения, в значение QOValue, и затем оно может быть передано в оператор O-дельта-кодирования, выполняющий циклическое обращение (а именно, при помощи добавления или вычитания значения wrapValue), если значение QOValue меньше чем lowValue или больше, чем highValue.

45 Нужно понимать, в случае, когда применяют квантование, уровни квантования должны быть выбраны таким образом, чтобы циклическое обращение и обратное циклическое обращение, выполненные друг за другом, давали в точности одинаковый результат, то есть, чтобы операция циклического обнуления не меняла результат способа,

независимо от того, применяют ли положительное суммирование (или вычитание) или отрицательное суммирование (или вычитание), или наоборот. Циклическое обращение при этом также не должно давать в результате заведомо меньшего абсолютного значения или заведомо большего абсолютного значения. Это означает, что если для абсолютных значений данных используют два различных коэффициента квантования, то малые и большие значения данных должны квантоваться с меньшим коэффициентом квантования, чем средние значения данных, чтобы исключить неверную интерпретацию данных при выполнении деквантования и операции, обратной циклическому обращению. Обратимся к фиг. 1, где эскизно проиллюстрировано, каким образом трехмерная видеоинформация может быть разбита на различные структуры данных, такие как кадры, ракурсы, каналы, блоки данных, пакеты данных и отдельные значения данных. Опционально, могут также использоваться дополнительные структуры, например, группы кадров, группы блоков данных, инициализационные блоки данных и слайсы данных. Все эти различные структуры, в соответствии с настоящим изобретением, не требуют отделения друг от друга при обработке и кодировании входных данных (D1) с целью формирования кодированных данных (E2). Порядок следования данных может быть различным, однако в примере фиг. 1 блоки в заданном канале обрабатывают слева направо и сверху вниз. Все значения, которые уже были обработаны, то есть закодированы, и опционально, декодированы, если реализуют кодирование с потерями, могут быть также использованы для предсказания текущих или будущих значений, поскольку в таком случае как кодер, так и соответствующий декодер, предложенные в настоящем изобретении, могут иметь информацию о значениях, декодированных из кодированных значений.

В соответствии с настоящим изобретением, в большинстве случаев, предпочтительно, каналы YUV кодируют в следующем порядке: Y, U, V, а каналы BGR - в следующем порядке: G, R, B или G, B, R, поскольку это является наилучшим вариантом для временного предсказания каналов в способах, предложенных в настоящем изобретении. Временное предсказание каналов, как правило, является очень эффективным способом, когда заданное изображение кодируют в цветовом пространстве RGB. Временное предсказание каналов позволяет значительно снизить корреляцию между каналами. Если используют цветовое пространство YUV, то корреляция между каналами уже сама по себе невелика непосредственно из-за свойств этого цветового пространства. При этом, когда применяют цветовое пространство YUV, наибольшее количество информации содержит канал Y, и это означает, что каналы U и V могут кодироваться более эффективно.

Способы, предложенные в настоящем изобретении, опционально, могут иметь отдельные подспособы, а именно алгоритмы, выполненные с возможностью обработки отличающихся друг от друга структур данных. К примеру, канал одного из цветов может быть закодирован с использованием пространственного предсказания или временного предсказания. При этом могут также применяться подспособы, обеспечивающие различные типы временного предсказания. В одном из подспособов, например, для заданного канала используют значение в позиции, совпадающей с позицией в предыдущем кадре этого же канала, тогда как в другом подспособе может использоваться значение того же канала в позиции, совпадающей с позицией в предыдущем ракурсе, а в еще одном подспособе, к примеру, для предсказания значения канала 2 может использоваться значение в идентичной позиции канала 0.

Когда применяют временное предсказание, определять исходное значение предсказания не нужно, поскольку каждое значение в текущем канале имеет значение

предсказания, доступное из других кадров, ракурсов или каналов. «Предсказание временного типа» может также применяться для сходных блоков данных, пространственно следующих друг за другом в заданном канале. Как правило, когда применяют такой способ предсказания, присутствуют лишь несколько доступных вариантов блоков данных, поэтому количество различных подспособов увеличивается незначительно. Это также означает, что передача блочного дескриптора, типа «вектора движения», не является необходимой. Блочный дескриптор такого рода, опционально, может применяться, однако, как правило, дескриптор задают, например, с точностью до блока, а не отдельных значений данных, поэтому количество различных комбинаций значительно меньше, по сравнению с применением внутреннего предсказания вектора движения.

Даже в случае применения некоторой комбинации из альтернативных существующих алгоритмов с аналогичной точностью, в которых результат получают при помощи традиционных известных способов внутреннего или внешнего предсказания движения, способы, предложенные в настоящем изобретении, все равно обеспечивают значительное преимущество, поскольку решения, соответствующие настоящему изобретению, включают также эффективное кодирование остатка при помощи О-дельта-кодирования. Таким образом, в способах, предложенных в настоящем изобретении, при предсказании нет необходимости использовать отдельное внутреннее или внешнее предсказание движения, как нет и необходимости применять после этого отдельный способ кодирования остатка, например, DCT-преобразование.

В вариантах осуществления настоящего изобретения для пространственного предсказания, опционально, используют отличающиеся друг от друга значения предсказания. Известен метод предсказания на основе предыдущего значения (А для Х), которые применяют в существующих технологиях О-дельта-кодирования (см. приложение 1). Предпочтительно, одним из способов, которые применяют в вариантах осуществления настоящего изобретения, является использование, в качестве значения предсказания для Х, значения $P = A + B - C$. Опционально, могут также применяться множество других значений предсказания, например, $2A - D$, или предсказание РАЕТН, описанное в документации стандарта PNG. Значение предсказания, предпочтительно, сокращают или округляют до заданного диапазона возможных значений. К примеру, если заданное значение в канале 0, потенциально представимо значениями от 0 (lowValue) до 63 (highValue), и при этом $A = 60$, $B = 61$, $C = 52$, то $P = A + B - C = 69$, которое сокращают, округляют или принимают равным значению «насыщения» 63 (highValue).

Если $X = 62$, то OValue, при использовании модифицированного способа №1 О-дельта-кодирования, будет равно $62 - 63 = -1$. Это значение, при кодировании без потерь, не подвергают квантованию, и следовательно, значение QOValue также равно -1. Далее, нужно понимать, что -1 меньше, чем lowValue (0), и, следовательно, к значению QOValue должно быть добавлено значение wrapValue (64), в результате чего получают 63, которое используют в качестве значения ODelta. При кодировании значения Х с помощью данного способа, ChannelSpatialODeltaCoded, это значение помещают в буфер, для дальнейшего энтропийного кодирования. Аналогичную обработку выполняют также для остальных значений данных, и после того, как все значения канала обработаны, содержимое буфера значений ODelta сжимают, например, с использованием диапазонного кодирования, или последовательного кодирования длин серий (SRLE), получая значения выходных кодированных данных для канала 0.

Нужно понимать, что после обработки первой строки, для предсказания доступно только значение А, поэтому если в качестве значения предсказания используют $A+B -$

С, то в качестве значения предсказания будет использовано непосредственно значение А. Аналогично, для первого столбца доступно только значение В, которое будет использовано в качестве значения предсказания. Для самого первого значения, например верхнего левого значения в канале 0, не будет доступно никаких значений, которые
 5 могли бы быть использованы для предсказания. Для самого первого значения предсказания могут использоваться значения временного предсказания, то есть, значения, доступные, например, из предыдущего ракурса или предыдущего кадра. Если не имеется подходящего значения временного предсказания, то в качестве первого значения предсказания для канала 0 может применяться, например, нулевое значение
 10 или значение середины диапазона $(63 - 0 + 1) \div 2 = 32$, или отдельно передаваемое значение режима для канала/ракурса/кадра. В некоторых из способов допускается также использование различных предикторов, а также передача информации о том, какой из предикторов должен использоваться для каждого конкретного кадра, канала, блока или даже отдельного значения данных в кодированных данных (E2).

15 Аналогичные способы могут также применяться, например, для блоков данных. В приведенном ниже примере проиллюстрировано, каким образом блок 2 в канале 2 кодируют с помощью канала BlockChannelOODeltaCoded, который допускает квантование с коэффициентом, равным 4, для соответствующих кодированных значений, который, очевидно, предъявляют меньшие требования к качеству. Квантование с
 20 коэффициентом 4, означает, что значение lowValue равно 0, значение highValue равно 15 и значение wrapValue равно 16. В данном случае, например, блок 2 в текущем канале 2 содержит следующие значения:

45, 48, 50, 52

25 46, 48, 50, 51

46, 49, 49, 50

Блок 2 в канале 0 содержит следующие значения предсказания:

36, 39, 40, 42

30 36, 37, 39, 41

36, 39, 39, 41

В случае применения временного предсказания квантование кодированных значений, меняющее декодированные значения для канала 2, не влияет на значения предсказания
 35 в канале 0, поэтому обработка данных может быть упрощена, т.к. при определении значений OValue квантование можно не учитывать. В этом случае значения OValues будут следующими:

9, 9, 10, 10, 10, 11, 11, 10, 10, 10, 10, 9

40 После квантования значений, делением их на 4, будут сформированы следующие значения QOValues:

2 2 2 2 2 2 2 2 2 2 2 2

Если все значения находятся в заданном диапазоне, а именно от 0 до 15, циклическое обращение не требуется ни для одного из значений QOvalues, поэтому значения ODelta
 45 будут совпадать со значениями QOValues. При этом нужно также понимать, что все значения ODelta одинаковы, и поэтому можно сменить используемый способ кодирования с BlockChannelOODeltaCoded на способ BlockChannelOODeltaSame или на способ BlockChannelOIDeltaSame. При использовании способа BlockChannelOODeltaSame необходима передача только одного значения (2), и при этом значения блока могут

быть корректно декодированы в декодере, то есть, аналогичным образом, как это выполняется в кодере.

Для этого блока данных в декодер передают указание на то, что был применен способ кодирования BlockChannelOODeltaSame, и значение 2. После этого декодер
5 создает буфер, в котором содержатся 12 следующих значений (с разбиением на блоки размером 4×3):

2 2 2 2 2 2 2 2 2 2 2 2

В данном примере при деквантовании эти значения умножают на 4, и опционально, добавляют 1 для более точной оценки погрешности в диапазоне квантования. После
10 деквантования этих значений получают следующие сформированные значения:

8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8

После сложения этих деквантованных значений со значениями предсказания, которые совпадают со значениями в кодере из канала 0, значения блока будут иметь следующий
15 вид:

45, 48, 49, 51

45, 46, 48, 50

45, 48, 48, 50

Искажения, а именно, расхождения между заданными исходными данными и соответствующими декодированными данными после процедуры кодирования-
20 декодирования будут следующими:

0, 0, 1, 1

25 1, 2, 2, 1

1, 1, 1, 0

Эти искажения обусловлены квантованием и не являются значительными.

Предложенный способ очень эффективен, поскольку для всего блока данных размером
30 4×3 необходима передача лишь информации о выбранном способе кодирования и одного значения. Если в канале необходимо кодировать большее количество блоков при помощи такого же или по существу аналогичного способа кодирования, то все эти значения могут быть, например, диапазонно кодированы в заданном кодере и диапазонно декодированы в соответствующем заданном декодере, что позволяет
35 получить еще большую степень сжатия, а именно, равную: объем исходных или декодированных данных (D1) / объем кодированных данных (E2).

При использовании описанного выше ODelta-кодирования, опционально, канал 2 кодируют перед каналом 0, и канал 2, предпочтительно, используют для предсказания
40 канала 0. В этом случае, предпочтительно, все значения QOValues равны -2. Эти значения меньше lowValue, и, следовательно, необходимо циклическое обращение (то есть, сложение с wrapValue), после которого значения ODelta равны 14 (-2+16). Таким образом, описанное выше ODelta-кодирование, которое применяют при реализации вариантов осуществления настоящего изобретения, не расширяет диапазон значений и не требует присутствия знакового бита, и поэтому более эффективно, чем аналогичные известные
45 способы дельта-кодирования. О-дельта-кодирование также более эффективно, чем применение способа DC-кодирования канала разности для текущего блока данных.

Ниже будет рассмотрен пример, в котором выполняют пространственное кодирование блока размером 4×3 в канале 1 с использованием способа

BlockSpatialODeltaCoded. В данном примере способа, в качестве первого значения предсказания, используют значение 0. Исходные значения находятся в диапазоне от 0 до 255, а качество определено следующим образом: в кодере значение lowValue равно 0, значение highValue равно 35 и значение wrapValue равно 36. Очевидно, на

5 соответствующем этапе декодирования эти значения, предпочтительно, должны оставаться в исходном диапазоне, от 0 до 255, и, следовательно, на этапе декодирования lowValue равно 0, highValue равно 255, а wrapValue равно 256. Исходный блок, таким образом, имеет следующий вид:

141, 151, 148, 137

10 159, 150, 152, 147

159, 154, 153, 150

Коэффициент квантования для первых 16 уровней, то есть для уровней 0-15, и последних 16 уровней, то есть уровней 20-35 равен 7; для средних 4 уровней, а именно, 15 уровней 16-19, коэффициент квантования равен 8. Это означает, что все абсолютные значения ниже 112, а именно 0-111, квантуют с помощью значения 7, а именно делят на 7. Из следующих значений, вплоть до 144, а именно 112-143, вычитают значение 112, квантуют их с коэффициентом 8, и прибавляют значение 16. Из остальных значений, а именно, 144-255, вычитают значение 144, квантуют их с коэффициентом 7 и прибавляют

20 значение 20. Аналогично, деквантование первых 16 значений, а именно значений 0-15 выполняют при помощи умножения каждого из этих значений на 7. Средние значения, 16-19, декодируют при помощи вычитания из них значения 16, после чего умножают каждое из значений на 8 и добавляют значение 112. Последние значения, а именно, значения 20-25, декодируют, вычитая из каждого значения 20 и умножая их на 7, после

25 чего прибавляют значение 144.

Первое значение равно 141. Значение предсказания было инициализировано как нулевое, поэтому OValue равно 141, и соответственно, QOValue равно 19. Нет необходимости в циклическом обнулении, и поэтому значение ODelta также равно 19. Это значение декодируется в значение 136 ($16*7+3*8$) без циклического обращения и

30 при этом данное значение, опционально и предпочтительно, используют для предсказания будущих значений.

Второе значение равно 151. Опционально, используют значение предсказания, основанное на A и равное 136. OValue равно 15, а QOValue равно 2. Снова, нет необходимости в циклическом обращении, и значение ODelta равно 2. Это значение

35 декодируется обратно в значение 150 ($136+2*7$) без циклического обращения, при этом данное значение, опционально и предпочтительно, также используют для предсказания будущих значений.

Третье значение равно 142, а значение предсказания, основанное на A, теперь равно 150. OValue равно -2, а QOValue равно 0. Снова, нет необходимости в циклическом

40 обращении, т.к. хотя значение OValue и отрицательно, однако после квантование оно равно 0 и находится в пределах диапазона, и, следовательно, значение ODelta также равно 0. Это значение декодируется обратно в значение 150 ($150 + 0$) без применения циклического обращения, при этом данное значение, опционально и предпочтительно, также используют для предсказания будущих значений. Четвертое значение равно 137,

45 а значение предсказания, основанное на A, снова равно 150. Значение OValue равно -13. Значение QOValue равно -1. Теперь необходимо циклическое обращение, и значение ODelta равно $-1 + 36 = 35$. Когда это значение декодируют, значение $150 + 249$ превысит значение highValue, а именно, 255, поэтому необходимо циклическое обращение, и

результат составит $150 + 249 - 256 = 143$, который, опционально, используют для предсказания будущих значений.

Пятое значение находится в новой строке и равно 159. Предсказание теперь может выполняться с использованием значения *B*, которое равно 136. *OValue* равно 23, а *QOValue* равно 3. Циклическое обращение не требуется, и соответствующее значение *ODelta* равно 3. Это значение декодируется обратно в значение $136 + 21 = 157$ и, опционально и предпочтительно, его также используют для предсказания будущих значений. Шестое значение равно 150, и для него в качестве значения предсказания может быть использовано значение $A + B - C$. В данном случае значение предсказания будет равно $157 + 150 - 136 = 171$. Значение *OValue* равно -21, Значение *QOValue* равно -3. Необходимо циклическое обращение, и следовательно, значение *ODelta* равно 33. Декодированное значение должно быть циклически обращено, и результат составит 150 ($171 + 235 - 256$), при этом его, опционально и предпочтительно, также используют для предсказания будущих значений.

Обработку данных, предпочтительно, продолжают аналогично до конца заданного блока данных, при этом полный результат кодирования способом *ODelta*-кодирования имеет следующий вид:

```
19, 2, 0, 35, 3, 33, 0, 0, 0, 0, 1
```

Эти значения, предпочтительно, помещают в буфер и энтропийно кодируют, например, при помощи диапазонного *SRLE*-кодирования, вместе с остальными закодированными тем же образами значениями *ODelta*. В декодер, в составе кодированных данных (*E2*), передают информацию о применении способа *BlockSpatialODeltaCoded* и кодированные значения *ODelta*, после чего декодер способен восстановить значения в следующем виде:

```
136, 150, 150, 143
```

```
157, 150, 150, 143
```

```
157, 150, 150, 150
```

Искажения, а именно, ошибка между исходными данными (*D1*) и декодированными данными (*D3*) после обратных друг другу операций кодирования и декодирования следующие:

```
5, 1, -2, -8
```

```
2, 0, 2, 4
```

```
2, 4, 3, 0
```

Нужно понимать, что первое значение может быть практически любым, при этом хорошая оценка при инициализации, предпочтительно, позволяет легко улучшить результаты. Подобные первые значения могут также передаваться отдельно, что позволяет повысить эффективность кодирования остальных значений. Остальные значения, как правило, содержат много нулей, а также несколько значений, близких к 1, что соответствует +1, и близких к 35, что соответствует -1; в данном примере *highValue* было равным 35, а *wrapValue* было равным 36.

В некоторых случаях все кодированные значения являются нулевыми, и в этом случае, предпочтительно, применяют способ *BlockSpatialODeltaNotCoded*, в котором выполняют только пространственное предсказание и не передают никаких значений для *O*-дельта-кодирования. Как правило, для пространственного предсказания нужны кодированные значения, однако временное предсказание часто работает также с постоянными значениями, на что указывает слово "Same" («неизменное значение») в

названии соответствующего способа, или без кодированных значений, на что указывает "NotCoded" (без кодирования) в названии, вместо всех кодированных значений, что отражено при помощи слова "Coded" (кодировано) в названии способа. При применении способов, предложенных в настоящем изобретении, их, как правило, используют в качестве независимых друг от друга способов. Допускается также использование одного базового способа и затем применение подспособа, в котором определено, например, использование заданного источника для временного предсказания, метода кодирования и т.п. В следующем примере, в таблице 2, приведен список отличающихся друг от друга подспособов кодирования каналов или блоков данных. Каждый из подспособов обозначается, то есть, задается при помощи соответствующих 3 бит.

Таблица 2 Способы кодирования каналов и блоков данных

Способ	Базовый способ	Подспособ
Channel_Channel0ODeltaCoded	ChannelODelta	000
Channel_Channel1ODeltaCoded	ChannelODelta	001
Channel_ViewPreviousChannel0ODeltaCoded	ChannelODelta	010

Channel_ViewPreviousChannel1ODeltaCoded	ChannelODelta	011
Channel_ViewPreviousChannel2ODeltaCoded	ChannelODelta	100
Channel_FramePreviousODeltaCoded	ChannelODelta	101
Channel_ChannelPreviousODeltaNotCoded	ChannelODelta	110
Channel_SpatialODeltaCoded	ChannelODelta	111
Block_Channel0ODeltaCoded	BlockODelta	000
Block_Channel1ODeltaCoded	BlockODelta	001
Block_Channel0ODeltaSame	BlockODelta	010
Block_Channel0ODeltaNotCoded	BlockODelta	011
Block_FramePreviousODeltaCoded	BlockODelta	100
Block_BlockLeftODeltaCoded	BlockODelta	101
Block_BlockUpODeltaCoded	BlockODelta	110
Block_SpatialODeltaCoded	BlockODelta	111

Различные каналы, опционально, кодируют при помощи отличающихся друг от друга способов кодирования. Например, заданный первый канал (канал 0) кодируют с помощью блочного кодера, в котором, опционально, используют также способы BlockODelta для кодирования блоков данных, заданный второй канал (канал 1) кодируют с помощью способа О-дельта-кодирования на основе способа Channel_Channel0ODeltaCoded с временным предсказанием, а заданный третий канал (канал 2) кодируют с помощью способа О-дельта-кодирования на основе способа Channel_SpatialODeltaCoded с пространственным предсказанием. Если все каналы кодируют с помощью одного и того же способа, например, с помощью способа О-дельта-кодирования на основе пространственного предсказания, то, например, для соответствующего кадра передают информацию о выбранном способе, Frame_SpatialODeltaCoded, и для этого кадра не будет требоваться передача информации о других способах кодирования, то есть не будет требоваться применение других способов кодирования канала или способов кодирования блоков данных.

Далее на примере фиг. 1-4 будут описаны структуры данных, используемые в вариантах осуществления настоящего изобретения. Проиллюстрированы структуры данных и значения данных, которые могут применяться для пространственного предсказания в способах, предложенных в настоящем изобретении. На фиг. 1 показан пример структуры данных верхнего уровня, данные в которой состоят, например, из трех кадров, при этом каждый кадр имеет, например, два ракурса, а каждый ракурс имеет, например, три канала. Опционально, данные верхнеуровневой структуры могут содержать слайсы, группы блоков данных или иные структуры данных. Опционально, некоторые структуры могут отсутствовать, например, пакеты и/или ракурсы.

Предсказание значений данных, теоретически, может значительно отличаться в зависимости от различных имеющихся структур или в зависимости от выбранного порядка обработки значений или применяемых структур.

На фиг. 2 показан пример канала, имеющего шесть блоков данных. Блоки при кодировании обрабатывают в порядке слева направо и сверху вниз. На фиг. 2 каждый блок включает, например, три пакета, при этом каждый пакет имеет, например, 4 значения. При этом на фиг. 3 показан пример блока данных и составляющие его элементы.

Соответственно, в этом примере общий объем данных составляет: $-3*2*3*6*3*4=1296$ значений. В следующем примере показаны расположенные поблизости друг к другу значения данных, используемые для предсказания. На фиг. 4 значения A-N являются предшествующими значениями данных для позиции X. Значения от 'O' до 'T' представляют собой, например, предыдущие значения данных, если они находятся в одном или более других блоков, обработанных ранее, чем заданный текущий блок. Предшествующие пакеты данных, блоки данных, каналы, ракурсы и кадры могут также содержать значения данных, которые могут быть использованы в целях предсказания. Таким образом, фиг. 4 представляет собой схематическую иллюстрацию соседних значений данных, которые потенциально могут использоваться для предсказания в соответствии с настоящим изобретением.

Как уже упоминалось, в вариантах осуществления настоящего изобретения, описанных выше, предложены способы, позволяющие повысить эффективность сжатия данных за счет применения модифицированного варианта оператора O-дельта-кодирования. Способы, предложенные в вариантах осуществления настоящего изобретения, допускают применение множества альтернативных, отличающихся друг от друга способов предсказания, а также опционально квантования, корректно функционирующего совместно с циклическим обращением значений ODelta. В способах, предложенных в настоящем изобретении, применяют также энтропийное кодирование, что позволяет максимально эффективно использовать все преимущества от снижения энтропии, обеспечиваемого оператором O-дельта-кодирования. Предложенные способы применимы для различных типов структур данных, например, кадров, каналов, блоков данных и пакетов данных. Для каждой структуры применяют строго заданные способы кодирования, формирующие малые объемы соответствующих данных, кодированных методом O-дельта-кодирования. Экономия пропускной способности, необходимой для хранения и передачи кодированных данных, достижимая по сравнению с пропускной способностью, необходимой для соответствующих некодированных данных, потенциально является очень существенной и позволяет получить значительные преимущества.

Описанные выше способы и варианты осуществления настоящего изобретения, предпочтительно, могут применяться в связи с кодерами и декодерами данных. В

соответствии с иллюстрацией фиг. 5, варианты осуществления настоящего изобретения относятся:

(i) к кодеру 100 для кодирования входных данных D1 с целью формирования соответствующих кодированных данных E2 и к соответствующим, описанным выше, способам кодирования входных данных D1 с целью формирования соответствующих кодированных данных E2;

(ii) к декодеру 120 для декодирования кодированных данных E2 с целью формирования соответствующих декодированных данных D3. Опционально, декодированные данные D3 в точности идентичны входным данным D1 в случае кодирования без потерь, или декодированные данные D3 приблизительно совпадают с входными данными D1 в случае кодирования с потерями, или данные D3 отличаются от входных данных D1, например, вследствие преобразования, однако содержат по существу ту же информацию, которая присутствовала во входных данных D1, а также информацию о соответствующих способах для декодирования кодированных данных E2 с целью формирования декодированных данных D3;

(iii) к кодеку 130, включающему комбинацию из по меньшей мере одного кодера 100 и по меньшей мере одного декодера 120, при этом кодек 130, опционально, реализуют в одном устройстве или фактически реализуют с распределением по нескольким различным устройствам; например, кодер 130, может быть реализован в виде системы широковещательной передачи информации, в которой кодер 100 находится в первом пространственном местоположении, а множеством декодеров 120 во множестве отличающихся пространственных местоположений.

В пределах сущности и объема, заданных приложенной формулой изобретения, возможны модификации вариантов осуществления настоящего изобретения, приведенных в предшествующем описании. Такие выражения, как «включающий», «содержащий», «охватывающий», «состоящий из», «имеющий», «представляющий собой», которые использованы в описании и в формуле настоящего изобретения, должны пониматься как неисключающие, то есть допускающие также присутствие объектов, компонентов или элементов, явно не упомянутых. Использование единственного числа может также пониматься как относящееся к множественному числу. Числовые обозначения на приложенных чертежах, приведенные в скобках, имеют целью упрощение понимания пунктов формулы изобретения и не должны пониматься как ограничивающие сущность изобретения, заданного этими пунктами формулы изобретения.

ПРИЛОЖЕНИЕ 1: Обзор О-дельта-кодирования

Ниже приведен обзор О-дельта-кодирования. Представлено О-дельта-кодирование и связанные с ним технологии, включая кодер 1010, способ его использования, декодер 1020 и способ его использования.

Кодер (1010) для кодирования входных данных (DA1), включающих последовательность числовых значений, с получением соответствующих кодированных выходных данных (DA2 или DA3), включает схему обработки данных для применения к входным данным (DA1) одной из форм разностного и/или суммирующего кодирования для формирования одной или более соответствующих кодированных последовательностей, при этом одна или более соответствующих кодированных последовательностей подвергается циклическому переходу относительно максимального значения и/или циклическому переходу относительно минимального значений для формирования кодированных выходных данных (DA2 или DA3).

Способ использования кодера (1010) для кодирования входных данных (DA1),

включающих последовательность числовых значений, с получением соответствующих кодированных выходных данных (DA2 или DA3), включает:

- 5 (а) использование схемы обработки данных кодера (1010) для применения к входным данным (DA1) одной из форм разностного и/или суммирующего кодирования для получения одной или более соответствующих кодированных последовательностей; и
- (б) использование схемы обработки данных для выполнения, над одной или более соответствующими кодированными последовательностями, операции циклического перехода относительно максимального значения и/или циклического перехода относительно минимального значения для формирования кодированных выходных
- 10 данных (DA2 или DA3).

Декодер (1020) для декодирования кодированных данных (DA2, DA3 или DA4) с получением соответствующих декодированных выходных данных (DA5), включает схему обработки данных для обработки одной или более частей кодированных данных (DA2, DA3 или DA4), при этом схема обработки данных выполнена с возможностью

15 применения одного из видов разностного и/или суммирующего декодирования к одной или более соответствующих кодированных последовательностей упомянутых одной или более частей, при этом одна или более кодированные последовательности подвергаются операции циклического перехода относительно максимального значения и/или циклического перехода относительно минимального значения, для формирования

20 декодированных выходных данных (DA5).

Способ использования декодера (1020) для декодирования кодированных данных (DA2, DA3 или DA4) с получением соответствующих декодированных выходных данных (DA5), включает:

использование схемы обработки данных для обработки одной или более частей

25 кодированных данных (DA2, DA3 или DA4), при этом схема обработки данных выполнена с возможностью применения одной из форм разностного и/или суммирующего декодирования к одной или более соответствующим кодированным последовательностям упомянутых одной или более частей, при этом одну или более кодированных последовательностей подвергают операции циклического перехода

30 относительно максимального значения и/или циклического перехода относительно минимального значения, для формирования декодированных выходных данных (DA5).

Далее, исключительно в качестве примера и со ссылками на приложенные чертежи, будут описаны примеры осуществления для приложения 1,

где: Фиг. 6 представляет собой иллюстрацию кодека, включающего кодер и декодер,

35 реализованных для функционирования согласно настоящему изобретению;

Фиг. 7 представляет собой иллюстрацию шагов способа кодирования данных, выполняемого в кодере, показанном на фиг. 6; и

Фиг. 8 представляет собой иллюстрацию шагов способа декодирования данных, выполняемого в декодере, показанном на фиг. 6.

40 При описании вариантов осуществления настоящего изобретения будут использованы следующие сокращения и определения, приведенные в таблице 3:

Таблица 3: Сокращения и определения

Сокращение	Описание
ADC	Аналогово-цифровой преобразователь (analog-to-digital converter)
Кодек	Кодер и соответствующий декодер для цифровых данных
DAC	Цифро-аналоговый преобразователь (digital-to-analog converter)
DB	База данных (database) в памяти с произвольным доступом (Random Access

	Memory, RAM) или в памяти в режиме «только для чтения» (Read Only Memory, ROM)
DC	Постоянная составляющая (DC) исходного изображения, то есть, среднее значение изображения, а именно, значение, соответствующее средней яркости и представляющее собой низшую составляющую пространственной частоты изображения
RLE	Кодирование длин серий
ROI	Анализируемая область (Region of interest)
ROM	Память «только для чтения»
VLC	Кодирование переменной длины

В целом, на примере фиг. 6, настоящее изобретение относится к кодеру 1010 и связанному с ним способу функционирования; предпочтительно, кодер 1010 реализован в виде прямого О-дельта-кодера. При этом настоящее изобретение относится также к соответствующему декодеру 101020; предпочтительно, декодер 1020 реализован в виде обратного О-дельта-декодера. В вариантах осуществления настоящего изобретения, предпочтительно, применяется оператор прямого О-дельта-кодирования, который является вариантом упомянутого выше существующего способа дельта-кодирования, оптимизированным для побитного кодирования, а также вариантом, оптимизированным по диапазону в случае других данных. О-дельта-кодирование используют в вычислительном оборудовании или специальном цифровом оборудовании, где применяются информационные слова переменной длины, например, 8/16/32/64 бит, и/или где применяется кодирование переменной длины для 8/16/32/64-битных элементов данных, исходное значение которых выражено в диапазоне от 1 до 64 бит, и соответствующее кодированное значение формируют с использованием от 1 до 64 бит. Очевидно, кодер 1010 и декодер 1020 являются, в любом случае, осведомленными о типе числовых значений, присутствующих в данных DA1, например, исходных данных, и соответственно их определение или передача не будут рассматриваться здесь более подробно. Просто предполагается, что известен диапазон чисел (от MIN до MAX), и что данные DA1 подходят для применения.

Существующие методы дельта-кодирования расширяют диапазон значений от исходного (MIN, MAX) до результирующего (MIN-MAX, MAX-MIN). Это означает, что они также создают отрицательные значения, даже если исходные данные содержат только положительные значения. Оператор О-дельта-кодирования, соответствующий настоящему изобретению, никогда не создает значений, не лежащих в диапазоне соответствующих исходных значений, и не расширяет используемого диапазона данных, и, следовательно, эффективно применяется, например, при понижении энтропии и

соответствующем сжатии данных. Рассмотрим пример, в котором при помощи существующих способов дельта-кодирования обрабатывают потоки 5-битных данных, а именно, данных, лежащих в диапазоне значений от 0 до 31: значения данных, формируемых такими способами дельта-кодирования будут лежать в диапазоне от -31 до +31, а именно, будут присутствовать 63 значения, которые, соответственно, могут быть представлены с использованием 6 бит (т.е. знакового бита + 5 бит); в отличие от них, оператор прямого О-дельта-кодирования при работе с упомянутыми выше потоками 5-битных данных будет формировать значения, также лежащие в диапазоне от 0 до 31. При этом существующие способы дельта-кодирования не допускают рекурсивной реализации, тогда как оператор прямого или обратного О-дельта-кодирования в соответствии с настоящим изобретением может быть реализован рекурсивно и при этом также сохраняет используемый диапазон значений. Диапазон значений не обязательно должен иметь точность до одного бита, например, если значения от 0 до 31 представляемы 5 битами, оператор О-дельта-кодирования может использовать любой диапазон значений, к примеру, диапазон значений от 0 до 25, и при этом также функционировать корректным образом.

В принципе, способы О-дельта-кодирования, описанные в настоящем изобретении, всегда способны функционировать на базе имеющегося диапазона данных, примеры чего будут рассмотрены ниже. Способы О-дельта-кодирования могут быть усовершенствованы за счет предоставления информации, указывающей наименьшее встречающееся в данных значение ("lowValue") и наибольшее встречающееся в данных значение ("highValue"). Нужно отметить, что $lowValue \geq MIN$, а $highValue \leq MAX$, и что эти значения являются опциональными.

Ниже описаны два примера операторов прямого и обратного О-дельта-кодирования в соответствии с настоящим изобретением. Первый пример операторов прямого и обратного О-дельта-кодирования является эффективным и относительно простым в реализации, например, в электронном и/или вычислительном оборудовании, выполненном с возможностью исполнения одного или более программных продуктов, записанных на машиночитаемом носителе для хранения данных.

При реализации операторов прямого или обратного О-дельта-кодирования в соответствии с настоящим изобретением, предпочтительно, все значения данных в исходной последовательности являются положительными, и наименьшее значение равно 0. Опционально, может использоваться некоторое значение смещения, например, значение предварительного смещения или значение пост-смещения, для смещения значения данных таким образом, чтобы все они были положительными и наименьшим значением был «0». Оператор О-дельта-кодирования в соответствии с настоящим изобретением допускает непосредственное применение с любыми типами данных; как правило, он позволяет получить сжатие данных, а именно снизить требуемую скорость передачи данных, поскольку при добавлении значения смещения ко всем значениям или при вычитании значения смещения из всех значений диапазон значений данных может быть задан с использованием меньшего количества битов. Например, если исходные значения данных, перед применением оператора прямого или обратного О-дельта-кодирования лежат в диапазоне от -11 до +18, этот диапазон может быть преобразован в диапазон от 0 до 29 с помощью значения смещения, равного 11, при этом преобразованный диапазон, соответственно, может быть представлен 5 битами. Если подобное значение предварительного смещения или значение пост-смещения не применяют, исходные данные требуют для своего описания по меньшей мере 6 битов, а на практике, для удобства, часто применяют полный 8-битный байт со знаком.

Аналогичная оптимизация диапазона данных также возможна при использовании обобщенного оператора прямого или обратного О-дельта-кодирования. То есть, если с помощью оператора прямого или обратного О-дельта-кодирования, или какого-либо иного способа, формируют значения данных, которые могут быть представлены, с использованием значения смещения, меньшим количеством битов, чем полный диапазон значений, то такая оптимизация диапазона может выполняться на любом из этапов способа О-дельта-кодирования. Когда используют значение смещения, будь оно отрицательным или положительным, его необходимо передавать из кодера 1010 в декодер 1020, как это показано ниже на примере фиг. 6, фиг. 7 и фиг. 8.

Оператор прямого О-дельта-кодирования может иметь 1-битную реализацию, например, для побитного кодирования исходных данных DA1; в случае 1-битной реализации способ 1 и способ 3, в соответствии с последующим более подробным описанием, формируют значение «0», когда в исходных данных, проиллюстрированных на фиг. 6, отсутствуют изменения битового значения, и значение «1», когда происходит изменение битового значения. Предсказание для первого бита в исходных данных, опционально, имеет значение «0», и соответственно, значение первого бита исходных данных DA1 остается неизменным. Альтернативно и опционально возможно применение значения предсказания для первого бита в исходных данных, имеющего значение «1», но такое решение не дает никаких преимуществ при кодировании; по этой причине, если, опционально, предсказание для 1-битных данных всегда подразумевается равным «0», отсутствуют необходимость сообщения о выборе, то есть заранее заданное значение «0» применяют для кодера 1010 и декодера 1020, что снимает необходимость передачи информации об этом значении предсказания, и следовательно, повышает эффективность сжатия данных.

Ниже описан один из примеров прямого О-дельта-кодирования в соответствии с настоящим изобретением. В уравнении 1 представлен пример исходной последовательности битов, а именно, двадцать семь битов, включающих 17 единиц («1») и двадцать нулей («0»):

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

Ур. 1

ее энтропию E вычисляют на основе уравнения 2 (ур. 2):

$$E = 17 * \log_{10} \left(\frac{37}{17} \right) + 20 * \left(\frac{37}{20} \right) = 11.08523$$

Ур. 2

Количество битов, Min_bit, необходимое для кодирования энтропии E в уравнении 2 (ур. 2), может быть вычислено на основе теоремы Шеннона о кодировании источника, описанной в упомянутых выше документах DA7 и DA8, как это показано в уравнении 3 (ур. 3):

$$\text{Min_bits} = \frac{E}{\log_{10}(2)} = 36.82 \text{ бит}$$

Ур. 3

При выполнении обработки исходной битовой последовательности оператором прямого О-дельта-кодирования в соответствии с предшествующим описанием, а именно способом 1 и способом 3, получают следующую последовательность битов,

включающую тридцать семь битов, в которой имеются тринадцать единиц («1») и двадцать четыре нуля («0»):

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

5

Ур. 4

ее энтропию E вычисляют на основе уравнения 5 (ур. 5):

$$E = 13 * \log_{10}\left(\frac{37}{13}\right) + 20 * \left(\frac{37}{24}\right) = 10.41713$$

10

Ур. 5

Что может быть выражено минимальным количеством битов, Min_bits , полученным согласно уравнению 6 (ур. 6):

15

$$Min_bits = \frac{E}{\log_{10}(2)} = 34.60 \text{ бит}$$

Ур. 6

Битовую последовательность из уравнения 4 (ур. 4), предпочтительно подвергают
20 дополнительному кодированию с целью сжатия данных, например, с использованием по меньшей мере одного из следующего: кодирование длин серий (RLE), кодирование Хаффмана, арифметическое кодирование, интервальное кодирование, модификатор энтропии или SRLE-кодирование.

Оператор O -дельта-кодирования позволяет уменьшить количество битов,
25 необходимых для представления исходных данных $DA1$ при применении соответствующего способа энтропийного кодирования, например, RLE- или SRLE-кодирование выполняют над данными после применения оператора, к примеру, над данными из уравнения 4 (ур. 4), вместо исходных данных из уравнения 1 (ур. 1); 1-битный вариант оператора прямого O -дельта-кодирования, а именно способ 1 и способ 3,
30 формирует единицы («1»), когда в исходной битовой последовательности из уравнения 1 (ур. 1) присутствует много изменений, и формирует нули («0»), когда в исходной последовательности по уравнению 1 (ур. 1) имеется длинный поток одинаковых битов.

Обратный вариант оператора O -дельта-кодирования, является буквально обратным
35 способу 1 и способу 3, т.е. в нем битовые значения «0» заменяют на значения «1», а значения «1» заменяют на значения «0», когда в кодированном потоке данных, т.е. в данных $DA2$, присутствует значение «1», и не изменяют битовое значение, когда в кодированном потоке данных $DA2$ присутствует значение «0». Когда такой оператор O -дельта-кодирования применяют к битовому потоку, полученному в результате применения оператора прямого O -дельта-кодирования, то в качестве декодированных
40 данных $DA5$ получают исходных поток данных $DA1$; однако, как упоминалось ранее, предпочтительно, применяют дополнительное кодирование, такое как VLC-кодирование или кодирование Хаффмана, что тоже необходимо учитывать; это означает, что данные $DA3$ формируют на основе данных $DA2$ с использованием прямой операции энтропийного кодирования, а данные $DA4$ формируют на основе данных $DA3$ с
45 использованием обратной операции энтропийного кодирования.

Предпочтительно, исходный поток данных $DA1$, перед применением к нему кодирования, разбивают на два или более фрагмента. Такое разбиение дает возможность получить большую оптимизацию при кодировании исходного потока данных $DA1$.

Например, такое разбиение является предпочтительным, потому что изменяющиеся последовательности в данных DA1 дают больше единиц («1») при прямом О-дельта-кодировании, то есть, при применении способа 1 и способа 3, тогда как ровные, неизменяющиеся последовательности, то есть «ровные» последовательности, дают больше нулей «0», что, например, является более выгодным для последующего VLR-кодирования или кодирования Хаффмана, поэтому энтропия E может быть снижена для всего битового потока, образующего данные DA1, за счет разбиения его на множество фрагментов, которые могут кодироваться отдельно в соответствии с предыдущим описанием.

Далее будет описан один из примеров прямого О-дельта-кодирования согласно настоящему изобретению, в котором применяют множество отдельно кодируемых фрагментов. Первый фрагмент, включающий последовательность исходных одиночных битов, включает в общей сложности шестнадцать битов, а именно семь единиц («1») и девять нулей («0»), в соответствии с уравнением 7 (ур. 7):

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

Ур. 7

где $H(X) = 4,7621$, а $B = 15,82$, H - энтропия, а B - это Max_bit. Когда последовательность исходных битов из уравнения 7 (ур. 7) подвергают обработке прямым оператором О-дельта-кодирования, получают последовательность соответствующих преобразованных бит, в соответствии с уравнением 8 (ур. 8):

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

Ур. 8

где $H(X) = 4,3158$, а $B = 14,34$.

Второй фрагмент, включающий последовательность исходных одиночных битов, соответствует уравнению 9 (ур. 9):

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

Ур. 9

где $H(X) = 6,3113$, а $B = 20,97$. Когда последовательность исходных битов из уравнения 9 (ур. 9) подвергают обработке прямым оператором О-дельта-кодирования, получают последовательность соответствующих преобразованных битов, в соответствии с уравнением 10 (ур. 10):

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

Ур. 10

где $H(X) = 1,7460$, а $B = 5,80$. В данных примерах, как упоминалось ранее, $H(X)$ представляет собой энтропию E , а B представляет собой минимальное количество битов, необходимых для кодирования.

Наилучшее сжатие в данном примере на основе уравнения 7 (ур. 7) и уравнения 10 (ур. 10) получают, когда оба фрагмента отдельно подвергают обработке прямым оператором О-дельта-кодирования (а именно, кодирование в 14,34 битах + 5,80 битах = 20,14 бит в общей сложности); это меньше, чем 36,82 бита, требуемых изначально, а именно, меньше 34,60 бит, требуемых для битов после применения оператора прямого О-дельта-кодирования, или исходного количества битов, требуемого после разбиения (= 15,82 бит + 20,97 бит = 36,79 бит). Предпочтительно, разбиение исходного битового потока в данных DA1 на фрагменты выполняют автоматически при помощи анализа энтропии E исходных данных DA1 и соответствующей энтропии H модифицированных

данных, то есть данных, входящих в состав данных DA2, по частям.

Сжатие данных, опционально, реализуют «грубо», просто разбивая фрагменты данных DA1 на новые кодируемые фрагменты, если в данных DA1, имеются множество фрагментов с длинными сериями, при условии, что на протяжении последовательности
5 есть достаточно большая область данных, в которой значения битов быстро изменяются. Опционально, некоторые из фрагментов данных DA1 кодируют без применения оператора О-дельта-кодирования, например, если имеется длинная серия одинаковых битов с относительно малым количеством отдельных отличающихся битов между ними; в таком случае оператор прямого О-дельта-кодирования не обладает значительными
10 преимуществами для сжатия данных.

Разбиение данных DA1 на менее крупные фрагменты имеет недостаток, заключающийся в необходимости формирования дополнительной служебной информации, которая увеличивает объем закодированных данных DA2. Такая служебная информация включает, например, информацию, указывающую на количество битов
15 или байтов данных, связанных с каждым новым фрагментом. Однако в любом случае передача по меньшей мере некоторого объема служебных данных является необходимой, и следовательно, при разбиении исходных данных на два фрагмента будет присутствовать лишь небольшой дополнительный объем служебных данных.

Для получения закодированного битового потока, который позднее может быть
20 декодирован, предпочтительно, после применения оператора прямого О-дельта-кодирования реализуют энтропийное кодирование, например, VLC-кодирование, кодирование Хаффмана, арифметическое кодирование, интервальное кодирование, RLE-кодирование, SRLE-кодирование, модификатор энтропии и т.п. По сравнению с фактическим кодированием данных, выполнение оптимизационных вычислений на
25 основе вычисленной энтропии E и значений оценки минимальных битов является более простым и вычислительно эффективным. Такой порядок выполнения позволяет получить значительное повышение скорости, и часто также дает оптимальный результат в отношении сжатия закодированных данных DA2. Альтернативно оптимизация энтропии может выполняться таким образом, что сначала кодируют данные исходного бита,
30 алфавитного символа, числа, байта или слова из данных DA1 с помощью некоторого другого способа, в результате чего получают битовый поток, оптимизированный по энтропии, и после этого применяют оператор прямого О-дельта-кодирования для модификации энтропийно-оптимизированного битового потока, в результате чего получают соответствующие закодированные данные, а именно, данные DA2. При этом
35 данные, полученные применением оператора О-дельта-кодирования, могут также кодироваться с использованием других способов кодирования, с получением данных DA3 на основе данных DA2.

В обобщенном операторе прямого О-дельта-кодирования применяют параметр, который описывает диапазон значений, используемых в данных DA1, то есть значение
40 или число битов, которые необходимы для представления этих значений. При этом оператор О-дельта-кодирования применяют в способе, который позволяет использовать положительные и отрицательные значения смещения, или, другими словами, положительные и отрицательные значения «пьедестала». Например, если данные DA1 представлены семью битами, то есть имеют возможные значения от 0 до 127, но содержат
45 только значения в диапазоне от 60 до 115, то, когда к данным DA1 прикладывают значение смещения, равное -60, в результате получают смещенные данные, имеющие значения в диапазоне от 0 до 55, которые при этом могут быть представлены как значения, содержащие только шесть битов, то есть таким образом может быть

достигнута определенная степень сжатия данных. Таким образом, обобщенный оператор прямого О-дельта-кодирования дает улучшенные результаты, когда в данных DA1 присутствует полный диапазон значений данных, то есть представляемых семью битами и для удобства представленных 8-битными байтами.

5 В соответствии с настоящим изобретением значения прямого О-дельта-кодирования, а именно, способа 1, могут быть вычислены с помощью процедуры, описанной примером фрагмента программного кода, приведенного ниже, для данных, которые имеют только положительные значения ($lowValue = MIN=0$ и $highValue = MAX = 127$, $wrapValue = 127-0 + 1 = 128$):

10

```
wrapValue = power(2, bits) = power(2, 7) = 128
```

```
preDAictionValue = (lowValue + highValue + 1) DAiv 2 = (wrapValue + 1) DAiv 2 + lowValue = 64
```

```
for all pixels
```

15

```
begin
```

```
  if(originalValue >= preDAictionValue) then
```

```
    ODAeltaValue = originalValue - preDAictionValue
```

```
  else
```

20

```
    ODAeltaValue = wrapValue + originalValue - preDAictionValue
```

```
  preDAictionValue = originalValue
```

```
EndDA
```

25 Для более детального описания упомянутого выше оператора О-дельта-кодирования ниже будет приведен еще один пример. Исходная последовательность значения соответствует уравнению 11 (ур. 11):

65, 80, 126, 1, 62, 45, 89, 54, 66

Ур. 11

30 Соответствующие значения дельта-кодирования приведены в уравнении 12 (ур. 12):

65, 15, 46, -125, 61, -17, 44, -35, 12

Ур. 12

Соответствующие значения О-дельта-кодирования приведены в уравнении 13 (ур.

35 13): 1, 15, 46, 3, 61, 111, 44, 93, 12

Ур. 13

где был применен циклический переход в пределах параметра wrapValue.

40 Оператор обратного О-дельта-кодирования, а именно, способ 1, может применяться для формирования значений обратного О-дельта-кодирования, например, в соответствии с реализацией при помощи следующего примера программного кода:

45

```

wrapValue = power(2, bits) = power(2, 7) = 128
preDAactionValue = (wrapValue + 1) DAiv 2 + lowValue = 64
5   for all pixels
      begin
          ODAelta Value = originalValue + preDAactionValue
          if (ODAeltaValue >= wrapValue) then
10          ODAelta Value = O-дельта-кодированияValue - wrapValue
          preDAactionValue = ODAeltaValue
      endDA

```

15 При исполнении этого программного кода и применении его к уравнению 13 (ур. 13), он формирует значения, приведенные в уравнении 14 (ур. 14):

65, 80, 126, 1, 62, 45, 89, 54, 66

Ур. 14

20 В данном примере в качестве значения power-of-two (степень двойки) используется wrapValue. Однако это не является обязательным, и wrapValue может иметь любое значение, большее, чем наибольшее значение данных, или значение, больше, чем используемый диапазон, если имеются также отрицательные значения, или диапазон в исходной последовательности данных модифицируют с помощью предварительного смещения. Это будет более подробно проиллюстрировано еще на одном примере, приведенном ниже.

25 Итак, подытоживая предшествующее описание на примере фиг. 6, настоящее изобретение относится к кодеру 1010 и декодеру 1020. Опционально, кодер 1010 и декодер 1020 применяют совместно в виде кодека, обозначенного в целом позицией 1030. Кодер 1010 выполнен с возможностью приема исходных входных данных DA1, которые кодируют, например, с использованием способа прямого О-дельта-кодирования, и получают соответствующие закодированные данные DA2 или DA3. 30
 Кодированные данные DA2 или DA3, опционально, передают по сети 1040 связи, или сохраняют на носителе 1050 для хранения данных, например, таком носителе данных, как оптический диск, память «только для чтения» (ROM) или аналогичном носителе. Декодер 1020 выполнен с возможностью приема закодированных данных DA2 или DA3, 35
 например передаваемых в виде потока по сети 1040 связи или предоставленных на носителе 1050 для хранения данных, и с возможностью применения обратного способа, например способа обратного О-дельта-кодирования, в результате чего получают соответствующие декодированные данные DA5, например по существу аналогичные исходным данным DA1. Кодер 1010 и декодер 1020 реализуют, предпочтительно, с 40
 использованием цифровой аппаратуры, например, вычислительной аппаратуры, которая выполнена с возможностью исполнения одного или более программных продуктов, например, кодов, приведенных в качестве примеров осуществления настоящего изобретения в настоящем описании. Альтернативно, кодер 1010 и/или декодер 1020 реализуют с использованием специальной цифровой аппаратуры.

45 В способе О-дельта-кодирования, исполняемом в кодере 1010, применяют шаги в соответствии с иллюстрацией фиг. 7. На первом шаге 1100, являющемся опциональным, обрабатывают входные данные DA1 для определения диапазона значений элементов этих данных. На втором шаге 1110, являющемся опциональным, на основе диапазона

значений вычисляют смещение, необходимое для смещения элементов данных в положительные значения, в результате чего получают соответствующий набор смещенных элементов. На третьем шаге 1120 элементы, опционально смещенные на втором шаге 1110, подвергают прямому О-дельта-кодированию, в результате чего получают соответствующие О-дельта-кодированные значения. На четвертом шаге 1130 О-дельта-кодированные значения и опциональное значение смещения, минимальное значение (lowValue) и/или максимальное значение (highValue) отдельно кодируют, например, с использованием кодирования длин серий (RLE), интервального кодирования или кодирования Хаффмана, в результате чего на основе данных DA2 формируют данные DA3. Значение смещения, минимальное значение (lowValue) и/или максимальное значение (highValue) не всегда могут быть сжаты, что требуется для их передачи, с использованием подходящего количества битов, из кодера 1010 в декодер 1020. При этом значение смещения, минимальное значение (lowValue) и/или максимальное значение (highValue) по отношению к прямому оператору О-дельта-кодирования являются опциональными элементами; к примеру, в некоторых ситуациях значение смещения имеет значение «0», lowValue имеет значение MIN, а highValue имеет значение MAX, то есть смещение не применяется, и используется полный диапазон. В частности, когда оператор прямого О-дельта-кодирования реализуют для 1-битных данных, то есть для побитного кодирования, он совсем не требует значения смещения, и тогда шаги 1100 и 1110 во всех случаях опускают. Если на шаге 1110 используют также значение смещения, вместе с ним должно быть обновлено значение диапазона, отражающее наибольшее и наименьшее значения. Количество различных значений, то есть wrapValue, также должно быть известно декодеру 1020, или, если это не так, кодер 1010 должен передавать его в декодер 1020 в составе сжатых данных. Опционально в кодере и декодере используют заданное по умолчанию значение wrapValue (= highValue - lowValue + 1). Опционально, кодер 1010 и/или декодер 1020 функционируют рекурсивно, например, для поиска оптимального варианта разбиения входных данных DA1, для их кодирования, на фрагменты, что позволяет получить оптимальное сжатие данных DA1, в результате чего формируют кодированные данные DA2.

В способе обратного О-дельта-кодирования, исполняемом в декодере 1020, применяют шаги в соответствии с иллюстрацией фиг. 3. На первом шаге 1200 данные DA2/DA3 или DA4 подвергают кодированию, которое обратно применяемому на описанном выше шаге 1130, и получают О-дельта-декодированные данные, при этом О-дельта-декодированные данные содержат О-дельта-кодированные значения и, опционально, имеют отдельное значение смещения. На втором шаге 1210 О-дельта-кодированные значения декодируют и формируют последовательность элементов данных. На третьем шаге 1220 последовательность элементов данных смещают с использованием опционального значения предварительного смещения и формируют декодированные данные DA5; в определенных ситуациях такое смещение задано равным «0», то есть по существу смещение не применяется. Также способ может быть выполнен без необходимости применения значения смещения, например, при осуществлении 1-битного кодирования, то есть побитного кодирования, в этом случае шаг 1220 может быть опущен. При этом декодеру 1020, чтобы он был способен корректно декодировать принятые в нем элементы данных, должно также быть известно значение wrapValue.

За счет применения смещения и получения только положительных значений, может быть достигнуто более эффективное сжатие в данных DA2 или DA3. Если все значения данных уже являются положительными, то добавлять какое-либо значение смещения не является необходимым. Очевидно, отрицательные значения смещения, опционально,

могут быть использованы для сужения используемого диапазона, как это показано в следующем примере, однако не являются обязательными.

Способы, показанные на фиг. 7 и фиг. 8, могут быть, опционально, дополнительно оптимизированы за счет использования только реально имеющихся значений, к которым
5 применяют О-дельта-кодирования. Такая оптимизация требует, чтобы используемые значения были известны. Например, в одном из приведенных выше примеров, в исходном наборе DA1 данных присутствуют только значения от 1 (= исходный минимум) до 126
(исходный максимум). В этом случае значение смещения равно 1 (-> lowValue = исходный минимум - смещение = 1 - 1 = 0, и highValue = исходный максимум - смещение = 126 - 1
10 = 125) После вычитания из исходных данных DA1 значения предварительного смещения в результате получают следующие значения, приведенные в уравнении 15 (ур. 15):

64, 79, 125, 0, 61, 44, 88, 53, 65

Ур. 15

Из уравнения 15 (ур. 15) определяют, что максимальное значение равно 125 (highValue
15 = исходный максимум - смещение = 126 - 1 = 125), так что «количество» (= максимальное значение разности = highValue - lowValue) может быть в этом случае равно 125, или наименьшее значение wrapValue может быть равно 126 (= «количество» + 1 = highValue - lowValue + 1). Далее необходимо сохранить и/или передать эти значения, и тогда
20 предыдущий пример может быть модифицирован, при помощи изменения значений процедуры, следующим образом:

wrapValue = 126 ("0" to "125" => 126 DAifferent values)

preDAiction Value = (lowValue + highValue + 1) DAiv 2 = (wrapValue + 1) DAiv 2 + lowValue = 63

25 Соответствующие значения после применения оператора О-дельта-кодирования приведены в уравнении 16 (ур. 16):

1, 15, 46, 1, 61, 109, 44, 91, 12

Ур. 16

30 Нужно понимать, что «все отрицательные значения разности» уменьшены в два раза (то есть, изменение диапазона = 128 - 126). Аналогично, в декодере 1020 значения процедуры должны быть изменены следующим образом:

wrapValue = 126

35 **preDAictionValue = (wrapValue + 1) DAiv 2 + lowValue = 63**

Соответствующие значения обратного О-дельта-кодирования приведены в уравнении 17 (ур. 17):

64, 79, 125, 0, 61, 44, 88, 53, 65

Ур. 17

40 Когда к уравнению 17 (ур. 17) добавляют предварительное смещение, получают приведенный в уравнении 18 (ур. 18) результат, соответствующий исходным данным из уравнения 15 (ур. 15), а именно:

45 **65, 80, 126, 1, 62, 45, 89, 54, 66**

Ур. 18

В данном примере диапазон значений является почти полным, поэтому за счет применения оператора прямого О-дельта-кодирования со значением смещения и максимальным значением (highValue) достигается лишь относительно небольшое

преимущество. Однако при этом все же может быть достигнуто снижение энтропии E , что дает меньшее количество значений в частотной таблице или в таблице кодирования, когда они переданы корректно. Более значительная выгода может быть достигнута при меньшей степени использования диапазона.

5 Далее будет рассмотрен один из примеров осуществления применимого на практике 1-битного способа прямого и обратного О-дельта-кодирования, а именно, способа 1 и способа 3 кодирования данных. Пример будет продемонстрирован при помощи исполняемого компьютерного программного кода. В способе, то есть в способе 1 и
10 способе 3, применяют описанные выше операторы прямого и обратного О-дельта-кодирования. Программный код выполнен с возможностью, при его исполнении в вычислительной аппаратуре, обработки данных из одного байтового буфера и помещения их в другой байтовый буфер. В программном коде функции GetBit (получить бит), SetBit (установить бит) и ClearBit (очистить бит) всегда обновляют значение HeaDAerBits (биты заголовка). Если следующий бит будет находиться в следующем
15 байте, обновляют также значение HeaDAerlnDAex (указатель на заголовок). Опционально, программный код может быть оптимизирован, в результате чего для источника и для назначения будет использоваться только один набор значений HeaDAerlnDAex и HeaDAerBits, так что значения будут обновляться только при записи данного бита в буфер назначения.

20

25

30

35

40

45

```
procedure EncodeODelta1u(APtrSrc : PByte; ASrcDstBitLen : PCardinal; APtrDst : PByte)
var
  iSrcHeaderIndex, iSrcHeaderBits, iIndex,
5   iDstHeaderIndex, iDstHeaderBits : Cardinal;
  bBit, bLastBit : Boolean;
begin
  // Reset offsets
  iSrcHeaderIndex := 0;
10  iSrcHeaderBits := 0;
  iDstHeaderIndex := 0;
  iDstHeaderBits := 0;

  // Initialise delta value
  bLastBit := False;

  // Go through all bits
  for iIndex := 0 to ASrcDstBitLen-1 do
20  begin
    // Read bit
    bBit := GetBit(APtrSrc, @iSrcHeaderIndex, @iSrcHeaderBits);

    // Set destination bit if current source bit is different than previous source bit
25  if (bBit <> bLastBit) then
    begin
      SetBit(APtrDst, @iDstHeaderIndex, @iDstHeaderBits);
      bLastBit := bBit;
30  end
    else ClearBit(APtrDst, @iDstHeaderIndex, @iDstHeaderBits);
    end;
  end;
35  end;

function DecodeODelta1u(APtrSrc : PByte; ASrcDstBitLen : PCardinal; APtrDst : PByte) :
Boolean;
var
```

40

45

```

iSrcHeaderIndex, iSrcHeaderBits, iIndex,
iDstHeaderIndex, iDstHeaderBits : Cardinal;
bBit, bLastBit : Boolean;
5
begin
  // Reset offsets
  iSrcHeaderIndex := 0;
  10
  iSrcHeaderBits := 0;
  iDstHeaderIndex := 0;
  iDstHeaderBits := 0;

  15
  // Initialise delta value
  bLastBit := False;

  // Go through all bits
  20
  for iIndex := 0 to ASrcDstBitLen^1 do
  begin
    // Read bit
    25
    bBit := GetBit(APtrSrc, @iSrcHeaderIndex, @iSrcHeaderBits);

    // Change bit value if source bit is true
    if (bBit = True) then
    30
    begin
      if (bLastBit = True) then
        bLastBit := False
      else bLastBit := True;
    35
    end;

    // Set destination bit based on bit value (True or False)
    40
    if (bLastBit) then
      SetBit(APtrDst, @iDstHeaderIndex, @iDstHeaderBits)
    else ClearBit(APtrDst, @iDstHeaderIndex, @iDstHeaderBits);
    end;
  45
  end;

```

Упомянутые выше прямые и обратные операторы О-дельта-кодирования, а именно, способ 1 или способ 3, предпочтительно применяют для сжатия любых типов данных,

имеющих цифровой формат, например, видеоданных, данных изображений, аудиоданных, графических данных, сейсмологических данных, медицинских данных, значений измерений, позиционных обозначений и масок. При этом с помощью оператора прямого О-дельта-кодирования могут быть сжаты также один или более аналоговых сигналов, если они сначала, перед сжатием, преобразованы в соответствующие цифровые данные, например, с использованием аналогово-цифровых преобразователей (ADC). Если используют оператор обратного О-дельта-кодирования, то после этой операции может применяться цифро-аналоговый преобразователь (DAC), если данные должны быть преобразованы в один или более аналоговых сигналов. Однако нужно понимать, что оператор прямого О-дельта-кодирования сам по себе, как правило, не эффективен для сжатия данных, но позволяет обеспечить эффективное сжатие данных при применении в комбинации с другими способами кодирования, например, кодирования переменной длины (VLC), арифметического кодирования, интервального кодирования, кодирования длин серий, SRLE-кодирования, модификатора энтропии и т.п. Такие способы кодирования применяют для данных DA2 после применения оператора прямого О-дельта-кодирования в кодере 1010. Кодированные данные DA2 должны быть соответствующим образом декодированы в исходный вид до того, как результирующие данные передают в оператор обратного О-дельта-кодирования, реализованный в декодере 1020. Оператор О-дельта-кодирования может также быть реализован в комбинации с другими типами модификаторов энтропии. В определенных ситуациях применение оператора прямого О-дельта-кодирования может давать повышение энтропии E , и при этом, предпочтительно, оператор прямого О-дельта-кодирования применяется алгоритмами сжатия данных избирательно, т.е. используется при кодировании данных только тогда, когда он положительно сказывается на степени сжатия данных, например, его применяют избирательно, исходя из характера сжимаемых данных, к примеру, избирательно применяют к избранным фрагментам входных данных DA1, как это было описано выше.

Оператор прямого О-дельта-кодирования был создан, например, для применения в комбинации с блочным кодером, описанном в заявке на патент США 13/584005, которая полностью включена в настоящий документ путем ссылки, а оператор обратного О-дельта-кодирования был создан для применения в комбинации с блочным декодером, описанном в заявке на патент США 13/584047, которая полностью включена в настоящий документ путем ссылки. Опционально оператор прямого О-дельта-кодирования и оператор обратного О-дельта-кодирования, предпочтительно, применяют в комбинации со способом многоуровневого кодирования, описанным в заявке на патент США 13/657382, которая полностью включена в настоящий документ путем ссылки. Предпочтительно, все типы 1-битных данных, к примеру, присутствующих в данных DA1, которые включают двоичные состояния, подвергают обработке 1-битным вариантом оператора прямого О-дельта-кодирования, в результате чего получают соответствующие преобразованные данные, которые затем подвергают фактическому энтропийному кодированию с формированием кодированных данных DA2 или DA3. Опционально, в соответствии с предшествующим описанием, оператор прямого О-дельта-кодирования применяют избирательно в зависимости от характера исходных данных DA1.

Опционально, возможно применение других способов изменения энтропии данных, до или после применения оператора прямого О-дельта-кодирования. К примеру, оператор прямого О-дельта-кодирования может также применяться непосредственно для многобитных данных, с использованием обобщенного варианта оператора прямого

О-дельта-кодирования. При этом описанный выше 1-битный вариант оператора прямого О-дельта-кодирования, предпочтительно, применяют для многобитных данных после того, как все используемые биты сначала помещают в специальную последовательность битов.

5 Когда в кодере 1010, в комбинации с прямым оператором О-дельта-кодирования, применяют несколько способов сжатия данных, в декодере 1020 соответствующие обратные операции выполняют в обратном порядке, например:

В кодере 1010 применяют следующую последовательность способов:

10 [данные DA1] => оператор прямого О-дельта-кодирования (способ 2)

=> VLC

=> EM => Арифметическое кодирование

[данные DA3]

15

Ур. 19

В декодере 1020 применяют следующую обратную последовательность способов:

[данные DA3] => обратное арифметическое кодирование

=> обратный EM

20

=> обратное VLC

=> оператор обратного О-дельта-кодирования (способ 2)

=> [данные DA5]

25

Ур. 20

где "VLC" - кодирование переменной длины, а "EM" - модификатор энтропии.

Оператор О-дельта-кодирования, описанный выше, обеспечивает обратимое кодирование без потерь. При этом оператор О-дельта-кодирования, опционально, может быть реализован специально для 1-битовых потоков данных, например, при выполнении побитного кодирования, а также и для других данных. Предпочтительно, любые данных могут быть обработаны с использованием обобщенного варианта оператора прямого О-дельта-кодирования. Предпочтительно, оператор прямого О-дельта-кодирования применяют в случае необходимости сжатия данных, а соответствующий оператор обратного О-дельта-кодирования применяют в случае необходимости декомпрессии сжатых данных. Опционально, если для сжатия применяют оператор обратного О-дельта-кодирования, оператор прямого О-дельта-кодирования и соответствующую ему обратную операцию применяют в обратном порядке; другими словами, сначала выполняют оператор обратного О-дельта-кодирования над исходным битовым потоком, и затем применяют оператор прямого О-дельта-кодирования для восстановления исходного битового потока. Один оператор О-дельта-кодирования повышает энтропию, а второй оператор О-дельта-кодирования уменьшает энтропию. Лишь в очень редких случаях оператор прямого О-дельта-кодирования совсем не меняет энтропию, и в этих случаях оператор обратного О-дельта-кодирования также не изменяет энтропию. Нужно отметить, что когда применяют операторы прямого и обратного О-дельта-кодирования, например, для способа 1, обратный порядок применения этих операторов соответствует нормальному порядку в способе 4. Подобное изменение порядка может выполняться также в способе 2 и способе 3.

В 1-битном, то есть предназначенном для побитного кодирования данных, варианте

оператора прямого О-дельта-кодирования его применение начинается, предпочтительно, без предсказания, а именно, по умолчанию в нем предполагается предсказание исходного значения «0». В обобщенном варианте оператор прямого О-дельта-кодирования начинают применять с предсказания, которое представляет собой середину
 5 используемого диапазона данных; например, если для значений входных данных DA1 используются 5 битов, то есть имеются тридцать два различных значения в диапазоне от 0 до 31, значение предсказания будет равно $32/2 = 16$. Предпочтительно, в оператор О-дельта-кодирования должна быть предоставлена информация об
 10 используемом диапазоне данных для элементов данных, обрабатываемых с помощью оператора.

Варианты осуществления, описанные выше, обеспечивают возможность снижения энтропии E, присутствующей в данных DA1, которые представлены в виде битов или числовых значений. Оператор прямого О-дельта-кодирования практически во всех
 15 случаях обеспечивает лучшее снижение энтропии, чем дельта-кодирование. Одинаковый результат достигается только в том случае, когда дельта-кодирование применяют в комбинации с байтовым циклическим переходом, и в разностном О-дельта-операторе с исходным предсказанием (способ 1) используют значения wrapValue = 256, lowValue = MIN = 0 и highValue = MAX = 255. Если используют другой способ прямого О-дельта-
 20 кодирования, или если во входных данных присутствует не полный диапазон, то оператор О-дельта-кодирования дает лучшие результаты путем передачи выбранного способа или lowValue, и/или highValue, то есть, того, что автоматически изменяет wrapValue. Меньшая энтропия позволяет обеспечить более высокую степень сжатия данных. Более высокие степени сжатия данных позволяют применять меньшие объемы
 25 памяти для хранения данных, а также использовать меньшие полосы пропускания при передаче сжатых данных, что дает соответствующее снижение энергопотребления.

В предшествующем описании предполагалось, что в кодере 1010 выполняется некоторая форма вычислений разности и суммы, а в декодере 1020 выполняются соответствующие обратные вычисления. В кодере 1010 допускается также применение
 30 других способов предсказания, и тогда соответствующее обратное предсказание будет применяться в декодере 1020. Это означает, что фактически существуют по меньшей мере четыре различных способа прямого О-дельта-кодирования, а также по меньшей мере четыре соответствующих способа обратного О-дельта-кодирования. Ниже приведено подробное и полное описание этих способов. Опционально, вычисления
 35 могут выполняться рекурсивно, в результате чего в кодированных данных DA2 (или DA3) получают большую степень сжатия. При осуществлении подобных рекурсивных вычислений применяют изменение диапазона чисел как функцию выполненного количества рекурсивных вычислений. Например, в кодере 1010, над данными DA1 выполняют следующую последовательность вычислений, в результате чего формируют
 40 кодированные данные DA2 (или DA3):

[Данные DA1] => (кодер) оператор прямого О-дельта-кодирования (способ 3)

=> (кодер) оператор прямого О-дельта-кодирования (способ 3)

(кодер) EM =>

=> (кодер) оператор прямого О-дельта-кодирования (способ 1)

(кодер) VLC

[Данные DA3]

Ур.

21

а в декодере 1020 выполняют соответствующие обратные операции:

[Данные DA3] (декодер) VLC =>

=> (декодер) оператор обратного О-дельта-кодирования (способ 1)

(декодер) EM =>

=> (декодер) оператор обратного О-дельта-кодирования (способ 3)

=> (декодер) оператор обратного О-дельта-кодирования (способ 3)

[Данные DA5]

Ур. 22

Всякий раз в этих четырех способах при применении к данным операторов кодирования, как это показано уравнением 21 (ур. 21, соответствующее способу 1), уравнением 22 (ур. 22, соответствующим способу 2), уравнением 23 (ур. 23, соответствующим способу 3) и уравнением 24 (ур. 24, соответствующим способу 4), опционально, возможны попытки применения всех этих способов, поскольку один из них может дать большее снижение энтропии обработанных данных, чем остальные. После оптимизации применения способов в кодере 1010 и/или декодере 1020, предпочтительно, эти или другие способы применяют до тех пор, пока выбранный способ или способы снижают энтропию по сравнению с требуемым объемом информации в данных. Так, способы 1-4 могут применяться для кодирования числовых значений, при этом определение «числовых значений» включает 1-битные данные, а также недвоичные числа, присутствующие в побитно кодируемых битовых подтоках или в многобитных значениях.

Операция вычисления разности дает разность последовательных числовых значений; и, соответственно, операция суммирования дает сумму последовательных числовых значений. Эти операции, выполняемые в кодере 1010, имеют соответствующие им обратные операции в декодере 1020. Разности или суммы могут вычисляться на основе текущего входного значения и предыдущего входного или результирующего значения, используемого в качестве значения предсказания. Могут также применяться и другие значения предсказания, при этом для их формирования могут применяться более ранние входные и выходные значения в кодере, при условии, что в декодере при этом может быть выполнена обратная операция.

Ни один из этих способов не обеспечивает значительного сжатия данных в кодере 1010 и декодере 1020, однако все способы могут эффективно применяться для снижения энтропии, в результате чего другие способы сжатия оказываются способными впоследствии сжимать данные, имеющие меньшую энтропию, более эффективно. Эти другие способы сжатия, опционально, представляют собой по меньшей мере одно из следующего: кодирование Хаффмана, арифметическое кодирование, интервальное кодирование, RLE-кодирование, SRLE-кодирование, кодирование модификатором энтропии. Однако для всех способов необходима передача нескольких числовых значений, чтобы с их помощью операция кодирования и соответствующая ей обратная операция всегда могли быть выполнены однозначным образом, например, чтобы обеспечивалось сжатие данных без потерь и последующая декомпрессия данных без потерь. Очевидно, кодер 1010 и декодер 1020 имеют информацию о типе числовых значений, содержащихся во входных данных DA1. Предпочтительно, предполагается,

что известен диапазон чисел, то есть диапазон, заданный значениями MIN и MAX. В принципе, способы могут во всех случаях работать на основе имеющегося диапазона данных. Числовые значения, необходимые для операций кодирования, включают наименьшее встречающееся числовое значение (lowValue) и наивысшее встречающееся числовое значение (highValue); где lowValue больше или равно MIN, а highValue меньше или равно MAX.

На основе этих значений могут быть получены также и другие необходимые числа. Предпочтительно, эти значения передают в различной форме, а недостающие значения, предпочтительно, вычисляют. Например, если известны два значения из набора ["lowValue", "highValue", «число»], где «число» равно [highValue - lowValue], то третье значение может быть вычислено на их основе. Исключение некоторых значений из данных DA2 и затем вычисление их в декодере 1020 позволяет получить большую степень сжатия в данных DA2.

Помимо этих значений необходимо число P, которое может применяться в качестве предыдущего значения при вычислении первого значения, то есть, «предсказание». В качестве числа P, то есть «предсказания», всегда может быть выбрано значение, находящееся между 0 и значением «числа». При этом в рассмотренных выше операциях кодирования должно быть обеспечено значение wrapValue, чтобы в декодере 1020, при декодировании данных DA2/DA3 или DA4, могли быть восстановлены исходные данные, а именно, это значение используют для максимально возможного сужения диапазона значений, формируемых операциями кодирования. Однако, это значение "wrapValue" должно быть большим, чем «число», и предпочтительно, оно будет иметь значение «число»+1. Опционально, в зависимости от характера данных DA1, как упоминалось выше, первое значение предсказания может быть выбрано равным «0», например, если предполагается, что данные DA1 содержат в основном малые значения, а не большие значения; альтернативно, значение первого «предсказания» может быть выбрано равным «числу», если предполагается, что данные DA1 содержат в основном большие значения, а не малые значения. В случае если относительно величины значений не было сделано никаких предположений, то, предпочтительно, в качестве значения «предсказания» используют значение "(wrapValue + 1) DAiv 2 + lowValue".

Далее будут описаны примеры операций, выполняемых в вычислительной аппаратуре при реализации вариантов осуществления настоящего изобретения.

В кодере 1010 первую прямую операцию вычисления разности, а именно, способ 1, предпочтительно, реализуют следующим образом; для всех значений данных, в программном цикле, вычисляют выходное значение, а именно, «результат», которое соответствует входному, то есть «исходному», значению:

результат = исходное - предсказание

если результат < lowValue, то результат = результат + wrapValue

Наконец, значение предсказания для следующего входного значения назначают равным текущему входному значению, а именно:

предсказание = исходное

В декодере 1020 первую обратную операцию вычисления разности, а именно, способ 1, предпочтительно, реализуют следующим образом: для всех значений данных, в программном цикле, вычисляют выходное значение, а именно, «результат», которое соответствует входному, то есть «исходному», значению:

результат = исходное + предсказание

если результата > highValue, то результат = результат – wrapValue

5 Наконец, значение предсказания для следующего входного значения назначают равным текущему результату, а именно:

предсказание = результат

10 В кодере 1010 вторую прямую операцию вычисления разности, а именно, способ 2, предпочтительно, реализуют следующим образом; для всех значений данных, в программном цикле, вычисляют выходное значение, а именно, «результат», которое соответствует входному, то есть «исходному», значению:

результат = исходное - предсказание

15 если результат < lowValue, то результат = результат + wrapValue

Наконец, значение предсказания для следующего входного значения назначают равным текущему результату, а именно:

предсказание = результат

20 В декодере 1020 вторую обратную операцию вычисления разности, а именно, способ 2, предпочтительно, реализуют следующим образом: для всех значений данных, в программном цикле, вычисляют выходное значение, а именно, «результат», которое соответствует входному, то есть «исходному», значению:

25 результат = исходное + предсказание

если результата > highValue, то результат = результат – wrapValue

Наконец, значение предсказания для следующего входного значения назначают равным текущему входному значению, а именно:

30 предсказание = исходное

35 В кодере 1010 первую прямую операцию суммирования а именно, способ 3, предпочтительно, реализуют следующим образом; для всех значений данных, в программном цикле, вычисляют входное значение, а именно, «результат», которое соответствует входному, то есть «исходному», значению:

результат = исходное + предсказание

если результата > highValue, то результат = результат – wrapValue

40 Наконец, значение предсказания для следующего входного значения назначают равным текущему входному значению следующим образом:

предсказание = исходное

45 В декодере 1020 первую обратную операцию суммирования а именно, способ 3, предпочтительно, реализуют следующим образом; для всех значений данных, в программном цикле, вычисляют входное значение, а именно, «результат», которое соответствует входному, то есть «исходному», значению:

результат = исходное - предсказание

если результат < lowValue, то результат = результат + wrapValue

5 Наконец, значение предсказания для следующего входного значения назначают равным текущему результату, а именно:

предсказание = результат

10 В кодере 1010 вторую прямую операцию суммирования а именно, способ 4, предпочтительно, реализуют следующим образом; для всех значений данных в программном цикле вычисляют входное значение, а именно, «результат», которое соответствует входному, то есть «исходному», значению:

результат = исходное + предсказание

если результата > highValue, то результат = результат - wrapValue

15

Наконец, значение предсказания для следующего входного значения назначают равным текущему входному значению следующим образом:

предсказание = результат

20

В декодере 1020 вторую обратную операцию суммирования а именно, способ 4, предпочтительно, реализуют следующим образом; для всех значений данных, в программном цикле, вычисляют входное значение, а именно, «результат», которое соответствует входному, то есть «исходному», значению:

25

результат = исходное - предсказание

если результат < lowValue, то результат = результат + wrapValue

Наконец, значение предсказания для следующего входного значения назначают равным текущему входному значению, а именно:

30

предсказание = исходное

35 Такие операции вычисления разности и суммирования, т.е. все четыре способа, применимы также к 1-битным данным, то есть могут применяться побитно, а именно, при реализации вариантов кодера 1010 и декодера 1020 с применением О-дельта-кодирования. В случае 1-битных данных, следующие значения являются уже известными кодери 1010 и декодери 1020, а именно, MIN = 0, MAX = 1. При этом, предпочтительно, предполагают, что lowValue = MIN = 0, а highValue = MAX = 1. Также, в этом случае «число», соответственно, равно [highValue - lowValue = 1-0 = 1], а значение wrapValue, 40 предпочтительно, выбирают равным значению «0», поскольку считается, что исключительно 1-битные данные могут иметь только положительные значения, начинающиеся с lowValue = MIN = 0. В случае 1-битных данных способ 1 и способ 3 дают идентичные друг другу результаты кодирования. Аналогично, способ 2 и способ 4 дают идентичные друг другу результаты кодирования. Осведомленность об этом 45 значительно упрощает информацию, которую необходимо передавать в данных DA2, поскольку могут быть сделаны допущения по заданным по умолчанию значениям, а именно: необходимо передавать только информацию о том, сколько раз выполняются операции вычисления разности, т.е. способ 1 и способ 2, и выбранное предсказание (входное значение (способ 1) или результирующее значение 2), чтобы декодер 1020 мог

быть способен выполнить соответствующую обратную операцию вычисления разности необходимое количество раз при декодировании данных DA2, DA3 или DA4, с целью формирования декодированных данных DA5.

Первый пример, который был создан с помощью способа 1 или способа 3, дающего аналогичные выходные данные, допускает также обработку с использованием способа 2 или способа 4, которые также формируют аналогичные выходные данные. Результат, показанный ниже, может быть получен при помощи применения этих способов к данным ур. 1:

0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 .

На этот раз обработанные данные имеют двадцать четыре единицы («1») и тринадцать нулей («0»), то есть, энтропия такая же, как и в первом примере, однако количество единиц и нулей поменялось местами. Это случается не всегда, напротив, при использовании различных способов также часто меняется и энтропия. Например, после первых четырех элементов данных способ 1 и/или способ 3 формирует три единицы («1») и один ноль («0»), тогда как в исходных данных и данных, обработанных способом 2 и/или способом 4, присутствуют две единицы («1») и два нуля («0»). Соответственно, в данном случае способ 1 и/или способ 3 дают меньшую энтропию, чем способ 2 и/или способ 4, а также энтропию, меньшую исходной.

В многобитной реализации, если данные DA1 содержат значения в диапазоне от -64 до +63, то $MIN = -64$, а $MAX = 63$. Если принять $lowValue = MIN$ и $highValue = MAX$, то «число» = 127, а $wrapValue$, предпочтительно, выбирают равным 128. Однако если данные DA1 изменяются случайным образом, значение «предсказания», предпочтительно назначают равным значению $[(wrapValue + 1) DA_{i-2} + lowValue = 64 + (-64) = 0]$.

Следует отметить, что если первое значение равно, например, -1, то первое значение, закодированное с помощью способа 1 и/или способа 2 прямого О-дельта-кодирования, будет равно $-1 - 0 = -1$, а соответственно, закодированное с помощью 3 и/или способа 4 прямого О-дельта-кодирования: $-1 + 0 = -1$. Следующие значение могут затем меняться в зависимости от дальнейших изменений данных, например, если вторым значением будет 5, то способ 1 прямого О-дельта-кодирования даст $5 - (-1) = 6$, способ 2 прямого О-дельта-кодирования даст $5 - (-1) = 6$, способ 3 прямого О-дельта-кодирования даст $5 + (-1) = 4$, а способ 4 прямого О-дельта-кодирования даст $5 + (-1) = 4$. Декодер 1020 в этом случае будет способен формировать, в качестве первого значения при использовании способа 1 и/или способа 2 обратного О-дельта-кодирования, $-1 + 0 = -1$, и при использовании способа 3 и/или способа 4 обратного О-дельта-кодирования, $-1 - 0 = -1$. Соответственно, второе значение, полученное с помощью способа 1 обратного О-дельта-кодирования, будет равно $6 + (-1) = 5$, с помощью способа 2 обратного О-дельта-кодирования, оно будет равно $6 + (-1) = 5$, с использованием способа 3 обратного О-дельта-кодирования, оно будет равно $4 - (-1) = 5$, и с помощью способа 4 обратного О-дельта-кодирования, оно будет равно $4 - (-1) = 5$.

Данное решение может затем быть оптимизировано, если диапазон чисел фактически содержит только значения между -20 и +27. В данном примере возможна передача, например, $lowValue = -20$, и $highValue = 27$. Если передают оба значения, то можно вычислить, что «количество» = 47, и $wrapValue$ в этом случае, предпочтительно, выбирают равным 48. Теперь можно вычислить значение «предсказания», которое равно $48 DA_{i-2} + (-20) = 4$. Далее, предыдущий пример дал бы, например, для значения -1, при использовании способа 1 или способа 2 О-дельта-кодирования кодирования: $-1 - 4 = -5$, а при использовании способа 3 или способа 4 О-дельта-кодирования кодирования: $-1 + 4 = 3$. Аналогично, второе значение для способов О-дельта-кодирования

кодирования было бы равным $(5 - (-1))=6$, $(5 - (-5))=10$, $(5+(-1)) = 4$ $(5+3) = 8$.

Декодирование снова выполняется корректно и дает первое значение для способа 1 и/или способа 2, равное, $-5 + 4 = -1$, а для способа 3 и/или способа 4, равное $3 - 4 = -1$.

Соответственно, вторые значения для различных способов будут декодированы как
 5 $(6 + (-1)) = 5$, $(10 + (-5)) = 5$, $(4 - (-1)) = 5$ и $(8 - 3) = 5$.

Следует отметить, что все значения в рассмотренных выше примерах находятся внутри диапазона, а именно, от -64 до 63 или от -20 до +27, поэтому для данных примеров значений не требуется ввод поправки, однако если какая-либо из положительных или отрицательных разностей является достаточно большой, то необходимо вносить в
 10 данные поправку, при помощи приведенных уравнений 21-24 (ур. 21-24), чтобы результирующие значения находились внутри диапазона. Следует отметить, что под поправкой в настоящем документе понимается значение циклического перехода.

Если известно lowValue, закодированные значения, предпочтительно, конфигурируют так, чтобы они начинались с 0 и заканчивались значением «числа», что позволяет
 15 упростить таблицу кодирования, которую необходимо передавать из кодера 1010 в декодер 1020 вместе с энтропийно-кодированными данными DA3. Эту операцию называют пост-смещением, при этом значение пост-смещения должно быть удалено из значений закодированных данных после энтропийного декодирования до выполнения операции обратного О-дельта-кодирования над данными DA4.

Как отмечалось выше, возможна также реализация смещения с использованием функциональности предварительного смещения, при которой исходные входные данные (DA1) преобразуют в положительные элементы, содержащие значения от нуля до «числа», уже непосредственно перед выполнением способа О-дельта-кодирования. Также, в такой ситуации, передача информации, необходимой для этой операции,
 25 предпочтительно, должна выполняться таким образом, чтобы способы «предварительного смещения» и способ О-дельта-кодирования не передавали повторно одну и ту же информацию, или опускали информацию, уже известную в другом способе. Для получения корректных выходных данных DA5 результат предварительного смещения должен быть устранен из декодированных данных перед выполнением
 30 операции обратного О-дельта-кодирования.

Таблица 4: Известный уровень техники

Документ	Подробности
Р1	«Кодирование переменной длины» ("Variable-length code"), 35 Википедия (по состоянию на 28 ноября 2012 года) URL: http://en.wikipedia.org/wiki/Variable-length_code
Р2	«Кодирование длин серий», Википедия (по состоянию на 28 40 ноября 2012 года) URL: http://en.wikipedia.org/wiki/Run-length_encoding

P3	«Кодирование Хаффмана», Википедия (по состоянию на 28 ноября 2012 года) URL: http://en.wikipedia.org/wiki/Huffman_coding
P4	«Арифметическое кодирование», Википедия (по состоянию на 28 ноября 2012 года) URL: http://en.wikipedia.org/wiki/Arithmetic_coding
P5	«Математическая теория связи» ("A Mathematic Theory of Communication"), Клод Шеннон, 1948 год (по состоянию на 28 ноября 2012 года) URL: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf
P6	«Дельта-кодирование», Википедия (по состоянию на 28 ноября 2012 года) URL: http://en.wikipedia.org/wiki/Delta_coding
P7	Теорема Шеннона о кодировании источника; Википедия (по состоянию на 28 ноября 2012 года) URL: http://en.wikipedia.org/wiki/Source_coding_theorem
P8	«Энтропия» – Википедия (по состоянию на 28 ноября 2012 года) URL: http://en.wikipedia/wiki/Entropy

ПРИЛОЖЕНИЕ 2: Обзор блочного кодера

Блочный кодер описан в патенте GB 2503295, содержание которого включено сюда путем ссылки. Предложен способ кодирования входных данных и кодер для кодирования входных данных.

Способ кодирования входных данных с формированием соответствующих кодированных выходных данных характеризуется тем, что указанный способ включает следующие шаги:

а) разделение указанных входных данных на множество блоков или пакетов, размер которых зависит от характера их содержимого, при этом указанные блоки или пакеты имеют один или более размеров;

б) применение множества преобразований к содержимому указанных блоков или пакетов с формированием соответствующих преобразованных данных;

с) проверку качества представления указанных преобразованных данных блоков или пакетов по сравнению с содержимым указанных блоков или пакетов перед применением указанных преобразований для определения того, удовлетворяет ли качество представления указанных преобразованных данных одному или более критериям качества;

д) в случае если качество представления указанных преобразованных данных указанного одного или более блоков или пакетов не удовлетворяет указанному одному или более критериям качества, дополнительное разделение и/или объединение указанного одного или более блоков или пакетов и повторение шага (б); и

е) в случае если качество представления указанных преобразованных данных указанного одного или более блоков или пакетов удовлетворяет указанному одному

или более критериям качества, вывод указанных преобразованных данных для предоставления указанных кодированных выходных данных, представляющих указанные входные данные.

5 Кодер для кодирования входных данных с формированием соответствующих кодированных выходных данных характеризуется тем, что указанный кодер содержит аппаратные средства для обработки данных, которые осуществляют:

а) разделение указанных входных данных на множество блоков или пакетов, размер которых зависит от характера их содержимого, при этом указанные блоки или пакеты имеют один или более размеров;

10 б) применение по меньшей мере одного преобразования к содержимому указанных блоков или пакетов с формированием соответствующих преобразованных данных;

с) проверку качества представления указанных преобразованных данных блоков или пакетов по сравнению с содержимым указанных блоков или пакетов перед применением указанных преобразований для определения того, удовлетворяет ли
15 качество представления указанных преобразованных данных одному или более критериям качества;

д) в случае если качество представления указанных преобразованных данных указанного одного или более блоков или пакетов не удовлетворяет указанному одному или более критериям качества, дополнительное разделение и/или объединение

20 указанного одного или более блоков или пакетов и повторение шага (b); и

е) в случае если качество представления указанных преобразованных данных указанного одного или более блоков или пакетов удовлетворяет указанному одному или более критериям качества, вывод указанных преобразованных данных для предоставления указанных кодированных выходных данных, представляющих

25 указанные входные данные.

ПРИЛОЖЕНИЕ 3: Обзор блочного декодера

Блочный декодер описан в патенте GB 2505169, содержание которого включено сюда путем ссылки. Предложен способ декодирования кодированных входных данных и кодер для декодирования входных данных.

30 Способ декодирования кодированных входных данных с формированием соответствующих декодированных выходных данных, характеризуется тем, что указанный способ включает следующие шаги:

а) обработку указанных кодированных входных данных для извлечения из них информации о заголовке, указывающей на кодированные данные, относящиеся к блокам
35 и/или пакетам, включенным в указанные кодированные входные данные, при этом указанная информация о заголовке включает данные, указывающие на одно или более преобразований, примененных для кодирования и сжатия исходных данных блока и/или пакета для включения их в качестве кодированных данных, относящихся к указанным блокам и/или пакетам;

40 б) подготовку поля данных в устройстве хранения данных для приема декодированного содержимого блока и/или пакета;

с) извлечение информации, описывающей указанное одно или более преобразований, и последующее применение инверсии указанного одного или более преобразований для декодирования указанных кодированных и сжатых исходных данных блока и/или
45 пакета с формированием соответствующего декодированного содержимого блока и/или пакета для заполнения указанного поля данных;

д) разбиение и/или объединение блоков и/или пакетов в указанном поле данных в соответствии с информацией о разбиении и/или объединении, включенной в указанные

кодированные входные данные; и

е) вывод данных из указанного поля данных в виде декодированных выходных данных, когда кодированные входные данные по меньшей мере частично декодированы.

Декодер для декодирования входных данных с формированием соответствующих декодированных выходных данных, характеризуется тем, что указанный декодер

содержит аппаратные средства для обработки данных, которые осуществляют:

а) обработку указанных кодированных входных данных для извлечения из них информации о заголовке, указывающей на кодированные данные, относящиеся к блокам и/или пакетам, включенным в указанные кодированные входные данные, при этом

указанная информация о заголовке включает данные, указывающие на одно или более преобразований, примененных для кодирования и сжатия исходных данных блока и/или пакета для включения их в качестве кодированных данных, относящихся к указанным блокам и/или пакетам;

б) подготовку поля данных в устройстве хранения данных для приема

декодированного содержимого блока и/или пакета;

с) извлечение информации, описывающей указанное одно или более преобразований, и последующее применение инверсии указанного одного или более преобразований для декодирования указанных кодированных и сжатых исходных данных блока и/или пакета с формированием соответствующего декодированного содержимого блока и/или пакета для заполнения указанного поля данных;

д) разбиение и/или объединение блоков и/или пакетов в указанном поле данных в соответствии с информацией о разбиении и/или объединении, включенной в указанные кодированные входные данные; и

е) вывод данных из указанного поля данных в виде декодированных выходных данных, когда кодированные входные данные по меньшей мере частично декодированы.

(57) Формула изобретения

1. Кодер (100) для кодирования входных данных (D1) для формирования соответствующих кодированных данных (E2), выполненный с возможностью обработки входных данных (D1) и кодирования исходных значений по меньшей мере их части с использованием по меньшей мере одного алгоритма дельта-кодирования с получением значений разностей, которые выражены с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений исходных значений, а также с возможностью формирования одного или более предикторов для использования при кодировании одной или более последующих частей входных данных (D1), при этом кодер (100) также выполнен с возможностью кодирования данных, сформированных при помощи упомянутого по меньшей мере одного алгоритма дельта-кодирования, такого как ODelta, DDelta, IDelta, PDelta, и упомянутого одного или более предикторов с применением по меньшей мере одного алгоритма энтропийного кодирования для формирования кодированных данных (E2), при этом упомянутые один или более предикторов включают по меньшей мере одно из следующего:

(i) один или более временных предикторов;

(ii) один или более локальных пространственных предикторов, подвергаемых квантованию; и

(iii) один или более локальных пространственных предикторов, использующих заранее вычисленные значения.

2. Кодер (100) по п. 1, отличающийся тем, что упомянутый по меньшей мере один алгоритм дельта-кодирования реализован как:

(a) использование схемы обработки данных для применения к входным данным (DA1) одной из форм разностного и/или суммирующего кодирования для получения одной или более соответствующих кодированных последовательностей; и

(b) использование схемы обработки данных для выполнения, над одной или более соответствующими кодированными последовательностями, операции циклического обращения вокруг максимального значения и/или циклического обращения вокруг минимального значения, для формирования кодированных выходных данных (DA2 или DA3) (= ODelta).

3. Кодер (100) по п. 1 или 2, отличающийся тем, что упомянутый по меньшей мере один алгоритм дельта-кодирования реализован посредством по меньшей мере одного из следующего:

(i) выражение значений разностей без знака, когда значения разностей включают только отрицательные значения (= DDelta) или только положительные значения (= IDelta);

(ii) использование значения сдвига для выражения значений разностей в пределах диапазона значений исходных значений (= PDelta).

4. Кодер (100) по любому из пп. 1-3, в котором в состав кодированных данных (E2) включают информацию, указывающую выбор по меньшей мере одного алгоритма дельта-кодирования, такого как ODelta, DDelta, IDelta, PDelta.

5. Кодер (100) по любому из пп. 1-4, выполненный с возможностью применения по меньшей мере одного алгоритма квантования при кодировании входных данных (D1) для формирования кодированных данных (E2), при этом упомянутый по меньшей мере один алгоритм квантования обеспечивает в кодере кодирование входных данных (D1) с потерями.

6. Кодер (100) по любому из пп. 1-5, выполненный с возможностью применения отличающихся друг от друга алгоритмов для кодирования данных в отличающихся друг от друга структурах данных, присутствующих во входных данных (D1).

7. Кодер (100) по любому из пп. 1-6, выполненный с возможностью применения оптимизации «скорость – искажения» при поблочном кодировании входных данных (D1).

8. Кодер (100) по п. 7, отличающийся тем, что оптимизацию «скорость – искажения» рассчитывают в кодере (100) для минимизации значения V в уравнении

$$V = D + \lambda * R,$$

где D - искажения, которые представляют собой сумму квадратов расхождений (SE) между входными данными (D1) и представлением входных данных (D1), кодированных в виде кодированных данных (E2) и декодированных в виде декодированных данных (D3);

R - скорость, которая представляет количество кодированных данных, измеренное в битах.

9. Кодер (100) по любому из пп. 1-8, отличающийся тем, что упомянутые по меньшей мере один алгоритм дельта-кодирования, такой как ODelta, DDelta, IDelta, PDelta, и/или по меньшей мере один алгоритм энтропийного кодирования выполнены с возможностью применения по меньшей мере одного из следующего: способов DC-кодирования, способов скользящего кодирования, способов многоуровневого кодирования, способов DST-преобразования, способов линейного кодирования, способов кодирования с масштабированием, способов кодирования с использованием баз данных, диапазонного кодирования, кодирования Хаффмана, RLE-кодирования и SRLE-кодирования.

10. Кодер (100) по любому из пп. 1-9, выполненный с возможностью кодирования

входных данных (D1), включающих структуры данных, которые соответствуют по меньшей мере одному из следующего: каналы YUV, каналы BGR.

11. Кодер (100) по п. 10, выполненный с возможностью кодирования данных в упомянутых каналах в следующем порядке: Y, U, V или в следующем порядке: G, B, R.

12. Кодер (100) по любому из пп. 1-11, выполненный с возможностью включать, в кодированные данные (E2), данные, которые указывают один или более алгоритмов кодирования, используемых кодером (100) для кодирования входных данных (D1) для формирования кодированных данных (E2).

13. Способ применения кодера (100) для кодирования входных данных (D1) для формирования соответствующих кодированных данных (E2), включающий:

(i) использование кодера (100) для обработки входных данных (D1) и для кодирования исходных значений по меньшей мере их части с использованием по меньшей мере одного алгоритма дельта-кодирования, такого как ODelta, DDelta, IDelta, PDelta, с получением значений разностей, которые выражены с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений исходных значений;

(ii) использование кодера (100) для формирования одного или более предикторов для использования при кодировании одной или более последующих частей входных данных (D1); и

(iii) использование кодера (100) для кодирования данных, сформированных при помощи упомянутого по меньшей мере одного алгоритма дельта-кодирования, и упомянутых одного или более предикторов путем применения по меньшей мере одного алгоритма энтропийного кодирования для формирования кодированных данных (E2), при этом упомянутые один или более предикторов включают по меньшей мере одно из следующего:

(i) один или более временных предикторов;

(ii) один или более локальных пространственных предикторов, подвергаемых квантованию;

(iii) один или более локальных пространственных предикторов, использующих заранее вычисленные значения.

14. Способ по п. 13, отличающийся тем, что упомянутый по меньшей мере один алгоритм дельта-кодирования реализован как:

(a) использование схемы обработки данных для применения к входным данным (DA1) одной из форм разностного и/или суммирующего кодирования для получения одной или более соответствующих кодированных последовательностей; и

(b) использование схемы обработки данных для выполнения, над одной или более соответствующими кодированными последовательностями, операции циклического обращения вокруг максимального значения и/или циклического обращения вокруг минимального значения для формирования кодированных выходных данных (DA2 или DA3) (= ODelta).

15. Способ по п. 13, отличающийся тем, что упомянутый по меньшей мере один алгоритм дельта-кодирования реализован посредством по меньшей мере одного из следующего:

(i) выражение значений разностей без знака, когда значения разностей включают только отрицательные значения (= DDelta) или только положительные значения (= IDelta);

(ii) использование значения сдвига для выражения значений разностей в пределах диапазона значений исходных значений (= PDelta).

16. Способ по любому из пп. 13-15, включающий выполнение кодера (100) с

возможностью применения по меньшей мере одного алгоритма квантования при кодировании входных данных (D1) для формирования кодированных данных (E2), при этом упомянутый по меньшей мере один алгоритм квантования обеспечивает в кодере кодирование входных данных (D1) с потерями.

5 17. Способ по любому из пп. 13-16, включающий выполнение кодера (100) с возможностью применения отличающихся друг от друга алгоритмов кодирования данных в отличающихся друг от друга структурах данных, присутствующих во входных данных (D1).

10 18. Способ по любому из пп. 13-17, включающий выполнение кодера (100) возможностью применения оптимизации «скорость – искажения» при поблочном кодировании входных данных (D1).

19. Способ по п. 18, отличающийся тем, что оптимизацию «скорость – искажения» рассчитывают в кодере (100) для минимизации значения V в уравнении

$$V = D + \lambda * R,$$

15 где D - искажения, которые представляют собой сумму квадратов расхождений (SE) между входными данными (D1) и представлением входных данных (D1), кодированных в виде кодированных данных (E2) и декодированных в виде декодированных данных (D3);

20 R - скорость, которая представляет количество кодированных данных, измеренное в битах.

20. Способ по любому из пп. 13-19, отличающийся тем, что упомянутые по меньшей мере один алгоритм кодирования, такой как ODelta, DDelta, IDelta, PDelta, и/или по меньшей мере алгоритм энтропийного кодирования выполнены с возможностью применения по меньшей мере одного из следующего: способов DC-кодирования, способов скользящего кодирования, способов многоуровневого кодирования, способов DCT-преобразования, способов линейного кодирования, способов кодирования с масштабированием, способов кодирования с использованием баз данных, диапазонного кодирования, кодирования Хаффмана, RLE-кодирования и SRLE-кодирования.

25 21. Способ по любому из пп. 13-20, включающий выполнение кодера (100) с возможностью кодирования входных данных (D1), включающих структуры данных, которые соответствуют по меньшей мере одному из следующего: каналы YUV, каналы BGR.

30 22. Способ по п. 21, включающий выполнение кодера (100) с возможностью кодирования данных в упомянутых каналах в следующем порядке: Y, U, V или в следующем порядке: G, B, R.

35 23. Способ по любому из пп. 13-22, включающий выполнение кодера (100) с возможностью включения, в кодированные данные (E2), данных, которые указывают один или более алгоритмов кодирования, используемых кодером (100) для кодирования входных данных (D1) для формирования кодированных данных (E2).

40 24. Декодер (120) для декодирования кодированных данных (E2) для формирования соответствующих декодированных данных (D3), выполненный с возможностью обработки кодированных данных (E2) с применением к ним по меньшей мере одного алгоритма энтропийного декодирования для формирования обработанных данных, а также с возможностью использования одного или более предикторов в сочетании с по

45 меньшей мере одним алгоритмом дельта-декодирования, таким как обратный ODelta, обратный DDelta, обратный IDelta, обратный PDelta, для декодирования обработанных данных и формирования декодированных данных (D3), при этом обработанные данные включают значения разностей, которые выражены

с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений исходных данных, из которых были сформированы кодированные данные (E2),

при этом упомянутые один или более предикторов включают по меньшей мере одно из следующего:

- (i) один или более временных предикторов;
- (ii) один или более локальных пространственных предикторов, подвергаемых квантованию;
- (iii) один или более локальных пространственных предикторов, использующих

заранее вычисленные значения.

25. Декодер (120) по п. 24, отличающийся тем, что упомянутый по меньшей мере один алгоритм дельта-декодирования реализован как:

(a) использование схемы обработки данных для применения к входным данным (E2) одной из форм разностного и/или суммирующего декодирования для получения одной или более соответствующих декодированных последовательностей; и

(b) использование схемы обработки данных для выполнения, над одной или более соответствующими декодированными последовательностями, операции циклического обращения вокруг максимального значения и/или циклического обращения вокруг минимального значения для формирования декодированных выходных данных (D3) (= обратный ODelta).

26. Декодер (120) по п. 24 или 25, отличающийся тем, что упомянутый по меньшей мере один алгоритм дельта-декодирования реализован посредством по меньшей мере одного из следующего:

(i) декодирование значений разностей, когда значения разностей включают только отрицательные значения (= обратный DDelta) или только положительные значения (= обратный IDelta);

(ii) использование значения сдвига для декодирования значений разностей, выраженных в пределах диапазона значений исходных значений (= обратный PDelta).

27. Декодер (120) по любому из пп. 24-26, выполненный с возможностью применения по меньшей мере одного алгоритма квантования при кодировании входных данных (D1) для формирования декодированных данных (D2), при этом упомянутый по меньшей мере один алгоритм квантования обеспечивает, в декодере (120), декодирование кодированных данных (E2) с потерями.

28. Декодер (120) по любому из пп. 24-27, выполненный с возможностью применения отличающихся друг от друга алгоритмов декодирования данных в отличающихся друг от друга структурах данных, присутствующих в кодированных данных (E2).

29. Декодер (120) по любому из пп. 24-28, отличающийся тем, что упомянутые по меньшей мере один алгоритм дельта-декодирования, такой как обратный ODelta, обратный DDelta, обратный IDelta, обратный PDelta, и/или по меньшей мере один алгоритм энтропийного декодирования выполнены с возможностью применения по меньшей мере одного из следующего: способов DC-кодирования, способов скользящего кодирования, способов многоуровневого кодирования, способов DCT-преобразования, способов линейного кодирования, способов кодирования с масштабированием, способов кодирования с использованием баз данных, диапазонного кодирования, кодирования Хаффмана, RLE-кодирования и SRLE-кодирования.

30. Декодер (120) по любому из пп. 24-29, выполненный с возможностью декодирования кодированных данных (E2), включающих структуры данных, которые соответствуют по меньшей мере одному из следующего: каналы YUV, каналы BGR.

31. Декодер (120) по п. 30, выполненный с возможностью декодирования данных в упомянутых каналах в следующем порядке: Y, U, V или в следующем порядке: G, B, R.

32. Декодер (120) по любому из пп. 24-31, выполненный с возможностью декодирования данных, включенных в состав кодированных данных (E2), которые указывают один или более алгоритмов дельта-декодирования, необходимых для применения в декодере (120) для декодирования кодированных данных (E2) для формирования декодированных данных (D3).

33. Способ декодирования кодированных данных (E2) в декодере (120) для формирования соответствующих данных (D3), включающий обработку кодированных данных (E2) с применением к ним по меньшей мере одного алгоритма энтропийного декодирования для формирования обработанных данных, а также использование одного или более предикторов в сочетании с по меньшей мере одним алгоритмом дельта-декодирования, таким как обратный ODelta, обратный DDelta, обратный IDelta, обратный PDelta, для декодирования обработанных данных и формирования декодированных данных (D3), при этом обработанные данные включают значения разностей, которые выражены с использованием диапазона значений, который не увеличен по сравнению с диапазоном значений исходных данных, из которых были сформированы кодированные данные (E2),

при этом упомянутые один или более предикторов включают по меньшей мере одно из следующего:

- (i) один или более временных предикторов;
- (ii) один или более локальных пространственных предикторов, подвергаемых квантованию;
- (iii) один или более локальных пространственных предикторов, использующих заранее вычисленные значения.

34. Машиночитаемый носитель, имеющий хранимые на нем машиночитаемые инструкции, исполняемые при помощи компьютеризованного устройства, включающего процессорную аппаратуру, для выполнения способа по п.13.

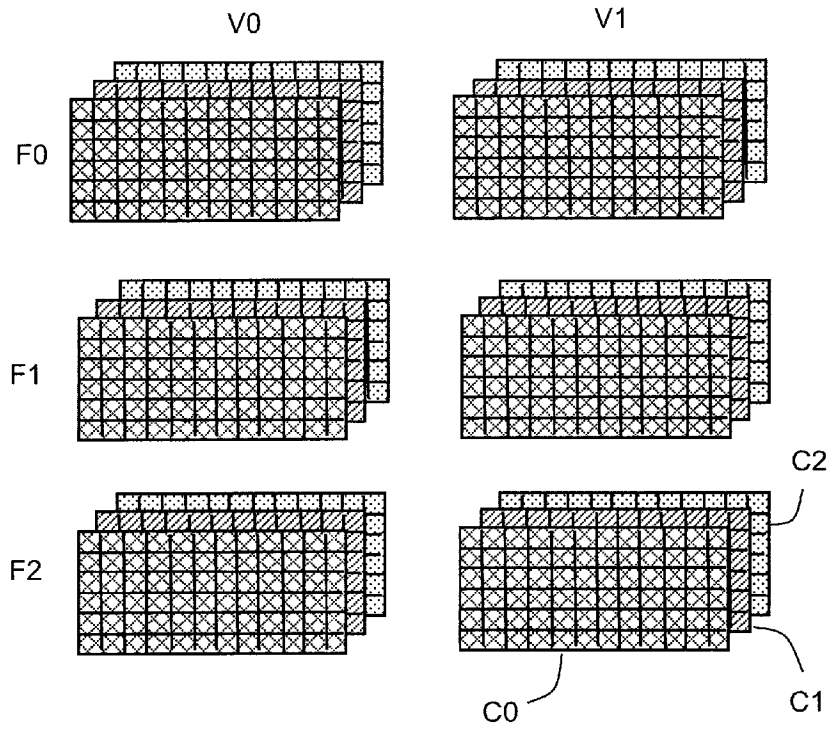
35. Машиночитаемый носитель, имеющий хранимые на нем машиночитаемые инструкции, исполняемые при помощи компьютеризованного устройства, включающего процессорную аппаратуру, для выполнения способа по п.33.

35

40

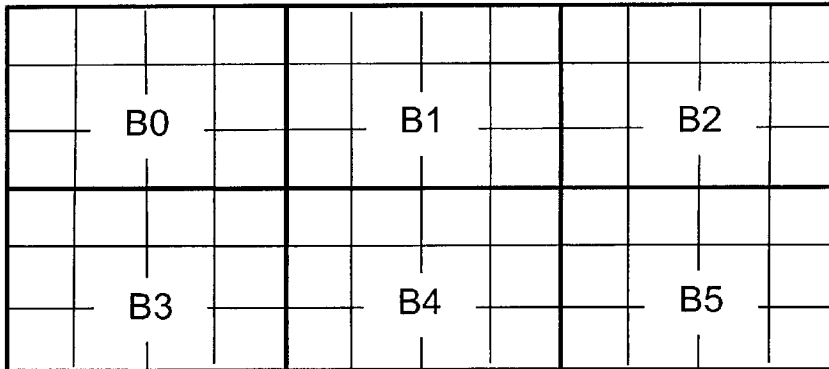
45

1/8



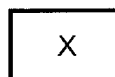
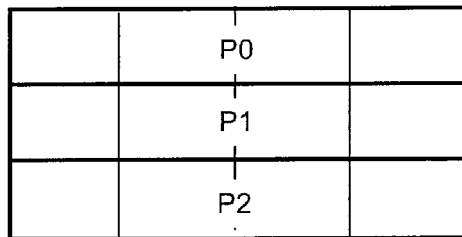
Фиг.1

2/8



Фиг.2

3/8



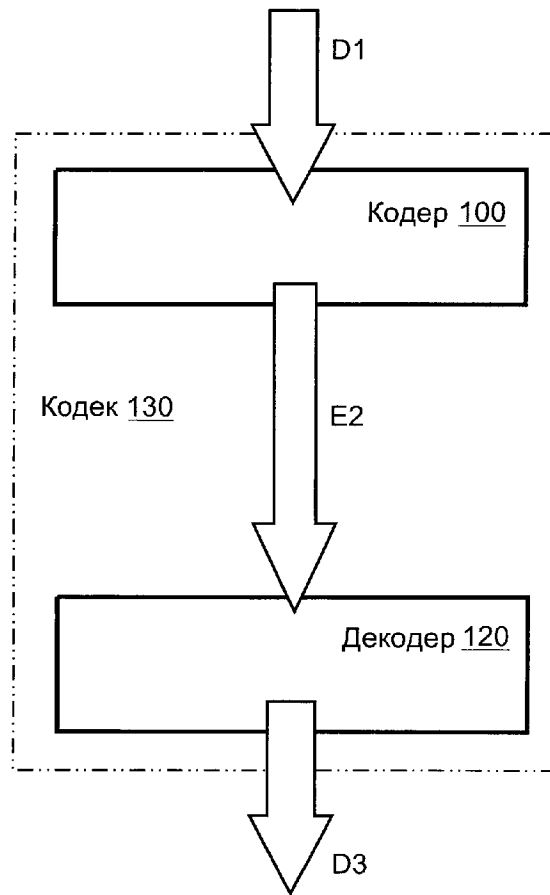
Фиг.3

4/8

F	H	E	L	M	N
G	C	B	I	J	K
D	A	X			
o	p				
q	r				
s	t				

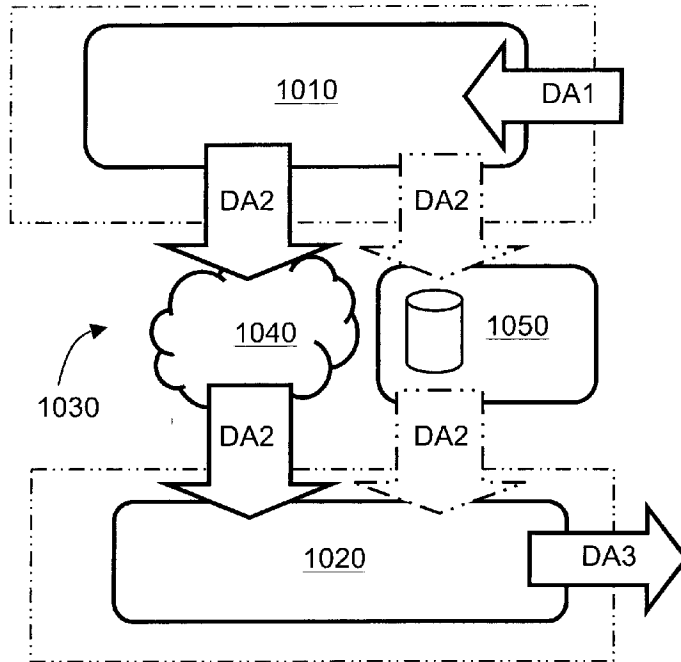
Фиг.4

5/8



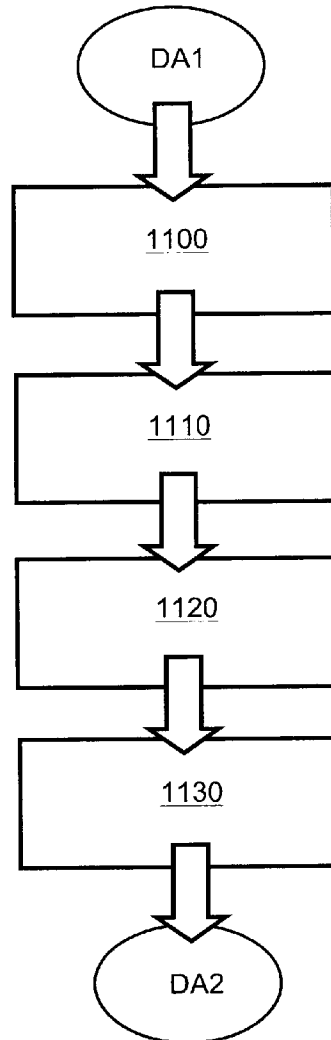
Фиг.5

6/8



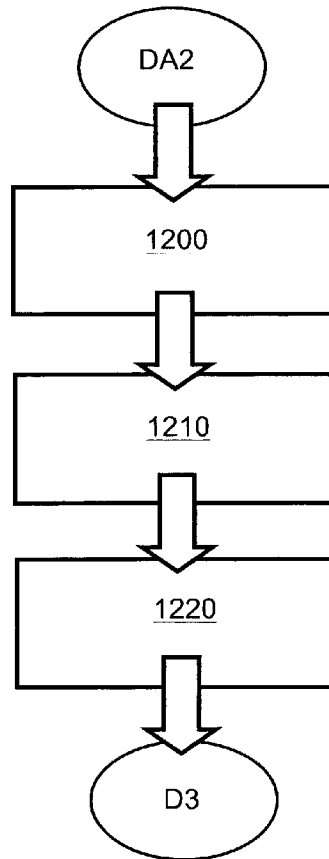
Фиг.6

7/8



Фиг.7

8/8



Фиг.8