



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

H03M 7/40 (2018.08); H03M 7/4006 (2018.08); H03M 7/42 (2018.08); H03M 7/6058 (2018.08); H04N 19/436 (2018.08); H04N 19/44 (2018.08); H04N 19/625 (2018.08); H04N 19/91 (2018.08)

(21)(22) Заявка: 2016130092, 20.02.2015

(24) Дата начала отсчета срока действия патента:  
20.02.2015

Дата регистрации:  
14.03.2019

Приоритет(ы):

(30) Конвенционный приоритет:  
20.02.2014 GB 1403039.9

(43) Дата публикации заявки: 21.03.2018 Бюл. № 9

(45) Опубликовано: 14.03.2019 Бюл. № 8

(85) Дата начала рассмотрения заявки РСТ на национальной фазе: 20.09.2016

(86) Заявка РСТ:  
EP 2015/025008 (20.02.2015)

(87) Публикация заявки РСТ:  
WO 2015/124324 (27.08.2015)

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

(72) Автор(ы):

КАЛЕВО Осси (FI),  
КЯРККЯЙНЕН Туомас (FI),  
ХУХТАНИЕМИ Артур (FI)

(73) Патентообладатель(и):

Гурулоджик Микросистемс Ой (FI)

(56) Список документов, цитированных в отчете о поиске: US 5838266 A, 17.11.1998. US 2011/0249754 A1, 13.10.2011. US 2012/0177300 A1, 12.07.2012. RU 2417518 C2, 27.04.2011.

(54) Способ и устройство для кодирования и декодирования исходных данных с использованием сжатия символов

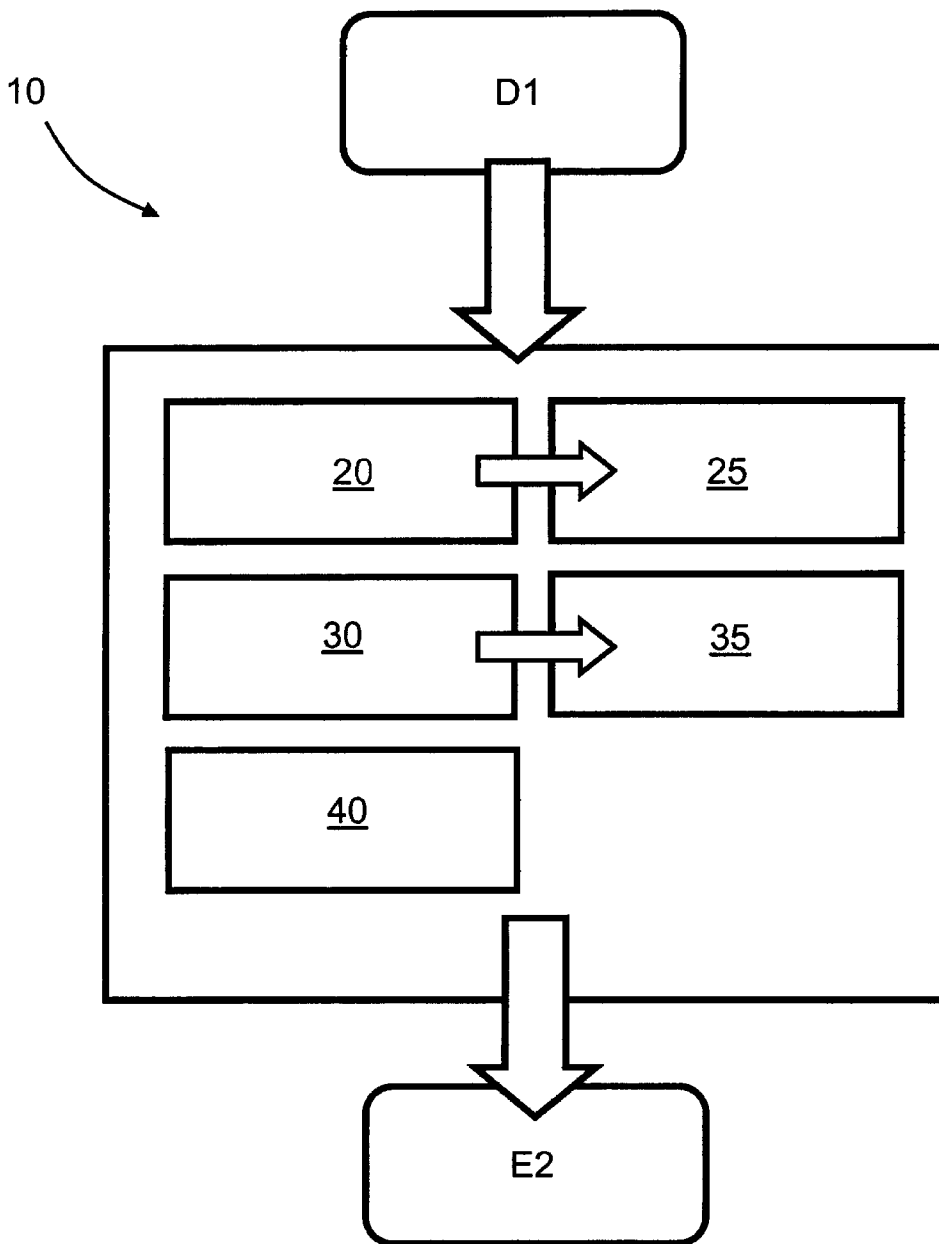
(57) Реферат:

Группа изобретений относится к области кодирования. Техническим результатом является повышение эффективности кодирования и декодирования данных. Способ содержит (а) анализ символов, представленных во входных данных (D1) и разделение и/или преобразование входных данных (D1) на один или более фрагментов данных; (b) генерацию, выполняемую как функция от появления символов по меньшей мере одного из следующего: одной или более

кодовых таблиц, одной или более частотных таблиц, одной или более таблиц длин кодовых слов, одной или более таблиц вероятностей для символов, представленных в одном или более фрагментах данных; (с) вычисление одного или более наборов индексов, связывающих символы в каждом фрагменте данных с элементами в по меньшей мере одном из следующего: одной или более кодовых таблицах, одной или более частотных таблицах, одной или более таблиц

длин кодовых слов, одной или более таблиц вероятностей; (d) сборку информации одного или более наборов индексов, связывающих символы в каждом фрагменте данных, совместно с по меньшей мере одним из следующего: одной или более частотными таблицами, одной или более кодовыми таблицами, одной или более таблицами

длин кодовых слов, одной или более таблиц вероятностей, информацией, указывающей на одну или более таких таблиц, в закодированные данные (E2); (e) сжатие символов в упомянутых одной или более таблиц в закодированные данные (E2). 6 н. и 20 з.п. ф-лы, 4 ил., 2 табл.



Фиг.2



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(51) Int. Cl.  
*H03M 7/30* (2006.01)  
*H03M 7/42* (2006.01)

(12) **ABSTRACT OF INVENTION**

(52) CPC

*H03M 7/40* (2018.08); *H03M 7/4006* (2018.08); *H03M 7/42* (2018.08); *H03M 7/6058* (2018.08); *H04N 19/436* (2018.08); *H04N 19/44* (2018.08); *H04N 19/625* (2018.08); *H04N 19/91* (2018.08)

(21)(22) Application: **2016130092, 20.02.2015**

(24) Effective date for property rights:  
**20.02.2015**

Registration date:  
**14.03.2019**

Priority:

(30) Convention priority:  
**20.02.2014 GB 1403039.9**

(43) Application published: **21.03.2018** Bull. № 9

(45) Date of publication: **14.03.2019** Bull. № 8

(85) Commencement of national phase: **20.09.2016**

(86) PCT application:  
**EP 2015/025008 (20.02.2015)**

(87) PCT publication:  
**WO 2015/124324 (27.08.2015)**

Mail address:  
**191036, Sankt-Peterburg, a/ya 24, "NEVINPAT"**

(72) Inventor(s):

**KALEVO Ossi (FI),  
KARKKAINEN Tuomas (FI),  
HUHTANIEMI Artur (FI)**

(73) Proprietor(s):

**Gurulogic Microsystems Oy (FI)**

(54) **METHOD AND DEVICE FOR CODING AND DECODING OF BASIC DATA USING COMPRESSION OF SYMBOLS**

(57) Abstract:

FIELD: coding.

SUBSTANCE: group of inventions refers to the field of coding. Method comprises (a) analyzing the symbols represented in the input data (D1) and splitting and/or converting the input data (D1) into one or more data fragments; (b) generating, performed as a function of the appearance of symbols of at least one of the following: one or more code tables, one or more frequency tables, one or more code word length tables, one or more probability tables for symbols represented in one or more pieces of data; (c) calculating one or more sets of indices linking the characters in each piece of data with the elements in at least one of the

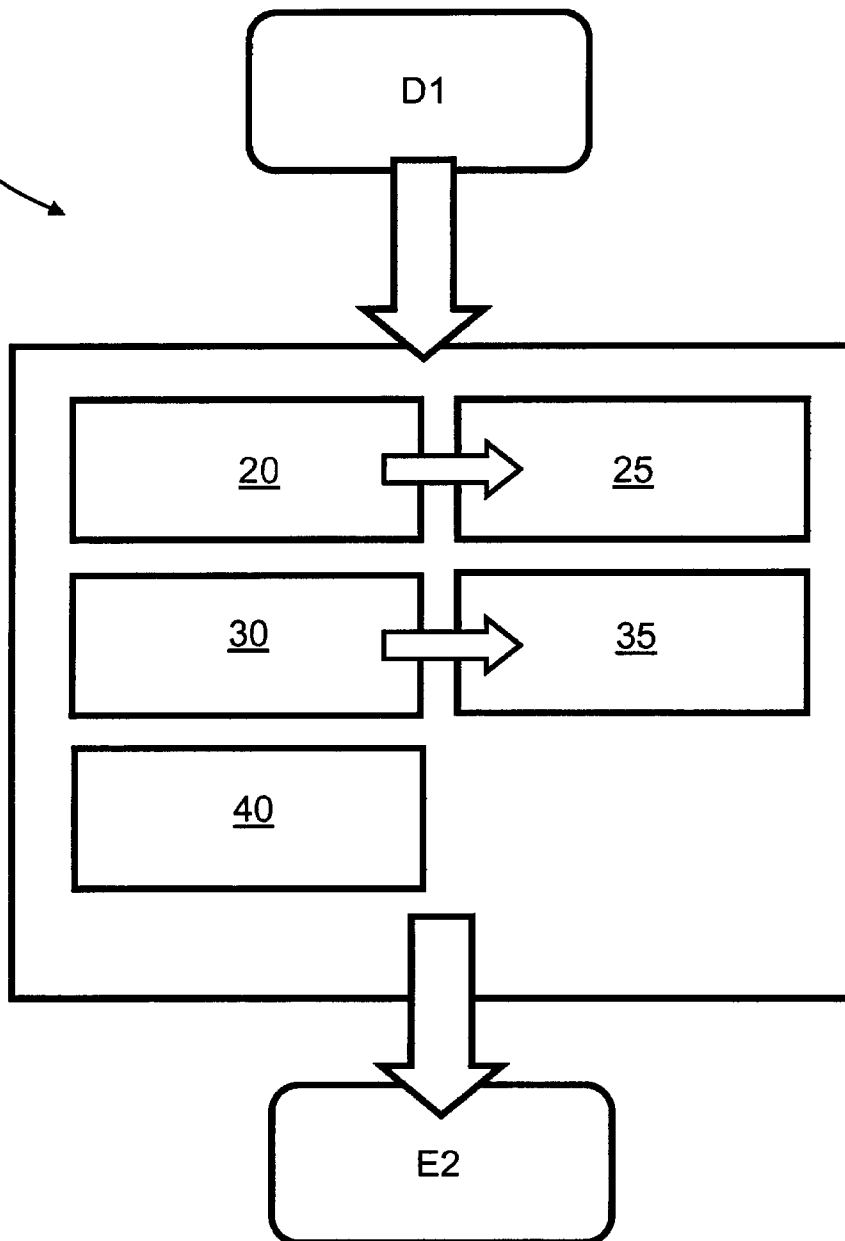
following: one or more code tables, one or more frequency tables, one or more codeword length tables, one or more probability tables; (d) compiling information of one or more sets of indices linking the symbols in each piece of data, together with at least one of the following: one or more frequency tables, one or more code tables, one or more code word length tables, one or more probability tables, information indicating one or more such tables, in coded data (E2); (e) compressing the characters in said one or more tables into encoded data (E2).

EFFECT: technical result is an increase in the efficiency of data coding and decoding.

RU 2 682 009 C2

RU 2 682 009 C2

10



Фиг.2

RU 2682009 C2

RU 2682009 C2

## ОБЛАСТЬ ТЕХНИКИ

Раскрытие настоящего изобретения относится к способам кодирования входных данных для генерации соответствующих кодированных данных. Кроме того, раскрытие настоящего изобретения также относится к способам декодирования указанных выше кодированных данных для генерации соответствующих декодированных выходных данных. Помимо этого, раскрытие настоящего изобретения также относится к кодерам и декодерам, способным реализовать указанные выше способы. Дополнительно раскрытие настоящего изобретения относится к компьютерным программным изделиям, содержащим машиночитаемый носитель информации, на котором хранятся машиночитаемые инструкции, исполняемые вычислительным устройством, в состав которого входят аппаратные средства обработки, выполняющие указанные выше способы.

## УРОВЕНЬ ТЕХНИКИ

На обзорной схеме, изображенной на фиг. 1, показано, что в известных способах кодирования входных данных D1 для генерации соответствующих кодированных выходных данных E2 ко входным данным D1 применяются один или более процессов T преобразования с целью генерации соответствующих преобразованных кодированных выходных данных E2, связанных с информацией C о данных кодовых таблиц, указывающей на одну или более кодовых таблиц, определяющих один или более используемых процессов T преобразования. Кодированные преобразованные данные E2 и информация C о данных кодовых таблиц, вместе образующие кодированные выходные данные E2, часто с помощью носителя данных и/или через сеть передачи данных передаются в один или более декодеров, которые способны выполнять один или более процессов  $T^{-1}$  преобразования для декодирования кодированных выходных данных E2 с целью генерации соответствующих декодированных данных D3. Часто желательно сжимать кодированные выходные данные E2 относительно входных данных D1, например, для уменьшения полной нагрузки сети связи при передаче кодированных выходных данных E2. Кроме того, желательно сжимать кодированные выходные данные E2 по существу без потерь, так чтобы декодированные данные D3 в точности воспроизводили информацию, включенную во входные данные D1. Достижимая степень сжатия кодированных выходных данных E2 относительно входных данных D1 потенциально недостаточна, если информация C о данных кодовых таблиц занимает значительный объем по отношению к кодированным преобразованным данным E2, то есть в этом случае информация C о данных кодовых таблиц соответствует значительному объему служебных данных в преобразованных кодированных данных E2.

Существуют несколько известных способов кодирования входных данных D1 для генерации кодированных выходных данных E2. Например, известное кодирование по Хаффману или другие способы кодирования VLC часто применяют для сжатия данных различных типов. Кроме того, возрастает популярность арифметического кодирования или интервального кодирования для сжатия входных данных, однако эти способы недостаточно эффективны в следующих ситуациях:

(i) частотная таблица для входных данных D1 еще не известна кодеру, предназначенному для кодирования входных данных D1 с целью генерации соответствующих кодированных выходных данных E2, и декодеру, предназначенному для декодирования кодированных выходных данных E2; и

(ii) объем входных данных относительно невелик, например, в ситуации, когда входные данные D1 передаются небольшими сегментами или фрагментами данных,

каждый из которых сопровождается соответствующей частотной таблицей.

Как упоминалось выше, такая неэффективность вызвана доставкой одной или более частотных таблиц, расходующих значительное пространство данных, если они не могут быть выбраны с использованием относительно небольшого количества  
5 идентификационных параметров из списка возможных частотных таблиц, которые, например, хранятся локально в декодере. Кроме того, отыскать в таком списке подходящие частотные таблицы представляется менее вероятным, чем подходящие кодовые таблицы. Часто входные данные D1, подлежащие кодированию, также могут  
10 изменяться локально, например, они преобразуются во время передачи через сеть связи для согласования с территориальными местными стандартами передачи данных по сети связи.

Существуют известные способы, позволяющие доставлять кодовые или частотные таблицы вместе с передачей кодированных информационных данных, полученных на основе символов. В наиболее известных способах применяется непосредственная  
15 доставка дерева Хаффмана или частот символов. Такие известные способы не столь удовлетворительны, поскольку они требуют доставки большого объема информации из кодера в соответствующий декодер. Кроме того, существуют также известные способы доставки длин символов кодовой таблицы, например, используемые в известной библиотеке IPP компании Intel, которая в настоящее время уже устарела; в ней  
20 используется способ сжатия кодовой таблицы с помощью утилиты "HuffLenCodeTablePack" и декодирования ее с помощью утилиты "HuffLenCodeTableUnpack"; однако этот способ не приносит удовлетворительных результатов и иногда даже увеличивает размер данных во время процессов кодирования. Кроме того, для выполнения способа также требуется наличие 256 символов, каждый  
25 из которых - от 0 до 255 - имеет ненулевую длину для своих кодовых слов. Способы доставки кодовых таблиц по-прежнему очевидно являются наиболее эффективными механизмами доставки, доступными в настоящее время для префиксных кодов, генерируемых, например, с помощью технологии кодирования по Хаффману. При доставке дерева Хаффмана из кодера в соответствующий декодер сгенерированные  
30 кодовые символы, поступающие из кодера, всегда одинаковы в кодере и декодере. Если доставляется только частотная таблица, для генерации фактического дерева Хаффмана на основе частотной таблицы должны использоваться одинаковые алгоритмы в кодере и декодере, если дерево Хаффмана требуется для обеспечения корректного декодирования символов в декодере. Если доставляются длины символов кодовой  
35 таблицы, то одинаковые способы преобразования длин символов в частотную таблицу также требуется использовать в кодере и декодере для обеспечения корректного декодирования символов. Передача длин символов из кодера в декодер не представляет собой практичный способ доставки частот для арифметического и интервального кодирования, поскольку они разработаны для поддержки более точных частотных  
40 таблиц, чем те, что формируются просто путем сообщения длин кодовых символов. Длины кодовых символов также могут использоваться при арифметическом и интервальном кодировании. Однако эти способы не предоставляют преимуществ по сравнению, например, с кодированием по Хаффману, если позднее не выполняется адаптивное обновление таблиц для последующих данных. Доставка информации,  
45 указывающей вероятности, обычно позволяет получать более оптимальные результаты кодирования при использовании интервального или арифметического кодирования в отличие от кодирования по Хаффману. Вероятности символов могут вычисляться путем деления частот появления символов на сумму частот появления символов, то есть на

количество символов. Доставка таких вероятностей предпочтительно выполняется с использованием масштабированных значений вероятностей. Масштабированные значения вероятностей могут вычисляться путем умножения исходных значений вероятностей символов на целое число, предпочтительно являющееся степенью двух,

5 то есть - на  $2^n$ , где  $n$  - целое число, с последующим округлением до ближайшего целого значения. Сумма этих масштабированных вероятностей в виде целых чисел выравнивается для того, чтобы она стала аналогичной значению множителя. Символ управляющего кода также предпочтительно создается для символов, для которых иным  
10 способом не выделено их собственное ненулевое масштабированное значение вероятности. Это означает, что вероятность тех символов, для которых требуется управляющий код, меньше той, что может быть представлена с помощью выбранного значения множителя. Кроме того, возможно создавать масштабированные вероятности без применения управляющего кода с помощью двух различных механизмов. Значение  
15 множителя может увеличиваться, а затем могут рассчитываться новые значения вероятностей. Кроме того, возможно обновлять до единицы те масштабированные значения вероятностей доступных символов, которые равны нулю. При таком обновлении значений вероятностей требуется, чтобы увеличение значений вероятностей компенсировалось уменьшением значений вероятностей других символов. Это  
20 выполняется для того, чтобы получить сумму вероятностей в точности такой же, как значение множителя. Согласно этой процедуре значения вероятностей не так хорошо оптимизированы, как это возможно было бы сделать, но управляющие символы не требуются, и в некоторых случаях такой подход можно рассматривать как оптимальное решение по кодированию. Длины символов или значения вероятностей определяют  
25 приблизительную оценку частотной таблицы, которая может использоваться для способов, в которых применяются символы с кодированием переменной длины, например для кодирования по Хаффману, интервального кодирования, арифметического кодирования и любых других способов кодирования переменной длины. Следует принимать во внимание, что таблица масштабированных вероятностей может  
30 непосредственно использоваться в качестве приблизительной оценки частот символов, если они требуются, и в этом случае длины символов необходимо вначале, перед их использованием, преобразовать в приблизительную оценку частот символов. Преобразование длин символов в частотную таблицу иллюстрируется ниже в процессе описания кодирования данных и доставки таблиц.

В множестве способов кодирования данных, известных на практике,  
35 оптимизированные кодовые таблицы вообще не используются, то есть в этих способах используются фиксированные кодовые таблицы для кодирования данных с целью генерации соответствующих кодированных данных и фиксированные кодовые таблицы для последующего декодирования кодированных данных. Иногда таблицы обновляются с использованием адаптивных способов, основанных на доставляемых символах.  
40 Согласно некоторым известным способам иногда используется пара различных кодовых таблиц (в альтернативном варианте используется пара частотных таблиц) для кодирования данных в кодере и, соответственно, для декодирования кодированных данных в декодере, при этом индекс, определяющий выбранную кодовую таблицу вероятностей или частотную таблицу, доставляется в виде информации, передаваемой  
45 из кодера в декодер. В определенных способах применяются отдельные таблицы для яркостного и цветового каналов, для межблочных и внутриблочных данных или для различных видов данных; однако отдельные таблицы передаются неэффективным способом; например, этот вопрос обсуждается на следующем веб-сайте (Wikipedia):

[http://en.wikipedia.org/wiki/Huffman\\_coding](http://en.wikipedia.org/wiki/Huffman_coding). В процессе распаковки с использованием способов Хаффмана должно реконструироваться дерево Хаффмана. В простейшем случае, когда частоты символов относительно предсказуемы, дерево поддается реконструкции и даже статистической настройке в каждом цикле сжатия, и, таким образом, каждый раз повторно используется за счет по меньшей мере некоторого показателя эффективности сжатия; в альтернативном варианте информация о дереве Хаффмана может передаваться а priori, то есть заранее.

Простой подход, заключающийся в наращивании частотных счетчиков, связанных с символами, которые кодируются в выходной поток сжатых данных, обладает серьезным недостатком, заключающимся в увеличении на практике объема сжатых данных по меньшей мере на несколько килобайтов (кБ), поэтому такой простой подход мало используется на практике. При сжатии данных с использованием канонического кодирования модель сжатия может в точности реконструироваться с помощью всего  $V2^B$  бит информации, где  $V$  - количество битов на символ, например, если символ содержит 8 бит, то потребуется 2 кБ.

Другой способ заключается в простом побитовом наращивании дерева Хаффмана в сжатом выходном потоке. Например, в предположении, что значение 0 представляет родительский узел, а 1 - лист, всякий раз, когда встречается лист, утилита построения дерева просто считывает следующие 8 бит для определения значения символа этого конкретного листа. Такой процесс продолжается рекурсивно до тех пор, пока не достигается последний лист, после чего дерево Хаффмана точно реконструировано, например, в декодере. При использовании такого способа объем служебных данных возрастает примерно от 2 до 320 байт при условии, что применяется 8-битовый алфавит.

Для разъяснения других известных способов кодирования данных и соответствующих способов декодирования кодированных данных далее приводится обзор декодирования по Хаффману. Следует принимать во внимание, что вместо декодирования по Хаффману также могут использоваться любые другие способы, например интервальное или арифметическое декодирование. Перед началом сжатия файла данных компрессор в кодере должен определить коды, применяемые при выполнении сжатия.

Если используется декодирование по Хаффману, то перед запуском сжатия заданного файла данных, содержащего символы, для генерации соответствующих кодированных выходных данных кодер должен определить коды, подлежащие использованию для представления заданных данных. Удобно использовать коды, основанные на вероятностях, то есть на частотах появления символов в заданном файле данных.

Однако частоты, вероятности или длины символов должны регистрироваться, например, как побочная, то есть дополнительная, информация в кодированных выходных данных, для того чтобы любой декодер Хаффмана мог декодировать кодированные выходные данные с целью генерации соответствующих декодированных данных. Удобно использовать частоты появления или длины символов в виде целых чисел либо вероятностей, которые могут быть выражены масштабированными целыми числами; причем такие целые числа, включаемые в дополнительные данные, очень часто увеличивают объем кодированных выходных данных на несколько сотен байтов.

Опционально возможно также записывать непосредственно коды с переменной длиной в кодированные выходные данные, но это при некоторых условиях может быть затруднительным, поскольку коды могут иметь взаимно различные размеры. В альтернативном варианте возможно в кодированные выходные данные записывать дерево Хаффмана, однако это требует больший объем данных для передачи по сравнению с простой передачей частот появления символов в заданных данных.



В процессе функционирования декодера должна предоставляться информация, касающаяся того, что находится в начале кодированного сжатого файла, принятого декодером для декодирования. На основе данных, выделенных из кодированного сжатого файла, например, находящихся в начале этого файла, декодер способен сконструировать алфавит дерева Хаффмана. После формирования дерева Хаффмана в декодере этот декодер затем способен декодировать остаток файла с использованием дерева Хаффмана в качестве средства декодирования. Декодер применяет относительно простой алгоритм декодирования, который включает следующие шаги:

- (a) переход к корню дерева Хаффмана, затем чтение первого бита кодированных выходных данных, подлежащих декодированию с использованием дерева Хаффмана;
- (b) если первым битом является "1", то осуществляется переход по верхней ветви дерева Хаффмана; если первым битом является "0", то осуществляется переход по нижней ветви дерева Хаффмана;
- (c) чтение второго бита кодированных выходных данных и использование второго бита таким же образом, как и на шаге (b), в направлении "листьев" дерева Хаффмана и т.д., до тех пор пока не будет достигнут "лист" дерева Хаффмана, в котором определяется исходный несжатый символ, часто связанный ASCII код; этот код затем выводится из декодера; и
- (d) шаги (b) и (c) повторяются, пока кодированные выходные данные не будут декодированы.

Известная в настоящий момент процедура кодирования по Хаффману преимущественно предназначена для использования в том случае, если длина кодируемая строка достаточно большая относительно кодовой таблицы, применяемой для генерации строки. Кроме того, такой современный способ кодирования по Хаффману преимущественно предназначен для применения в том случае, если кодовая таблица определена заранее как для кодера, так и для соответствующего декодера. В связи с этим требуются альтернативные способы кодирования, с помощью которых можно преодолеть упомянутые выше ограничения, связанные с известными подходами к кодированию и декодированию данных, например к упомянутым способам кодирования и декодирования по Хаффману.

### СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Настоящее изобретение направлено на поиск усовершенствованного способа кодирования данных (D1) для генерации соответствующих кодированных данных (E2).

Настоящее изобретение также направлено на поиск способа реализации усовершенствованного кодера, который способен выполнять упомянутый улучшенный способ кодирования данных.

Настоящее изобретение направлено на поиск улучшенного способа декодирования данных (E2) для генерации соответствующих декодированных данных (D3).

Настоящее изобретение направлено на поиск улучшенного декодера для декодирования вышеупомянутых кодированных данных (E2) для генерации соответствующих декодированных данных (D3).

В соответствии с первым аспектом предлагается способ кодирования входных данных (D1) в кодере для генерации соответствующих кодированных данных (E2), включающий:

(a) анализ символов, представленных во входных данных (D1), и разделение и/или преобразование входных данных (D1) с получением одного или более фрагментов данных;

(b) генерацию, в зависимости от появления символов, по меньшей мере одного из следующего: одной или более кодовых таблиц, одной или более частотных таблиц,

одной или более таблиц длин кодовых слов, одной или более таблиц вероятностей, для символов, представленных в одном или более фрагментах данных;

(с) вычисление одного или более наборов индексов, связывающих символы в каждом фрагменте данных с элементами по меньшей мере в одном из следующего: в одной или более кодовых таблицах, в одной или более частотных таблицах, в одной или более таблицах длин кодовых слов, в одной или более таблицах вероятностей;

(d) объединение информации одного или более наборов индексов, относящихся к символам в каждом фрагменте данных, с по меньшей мере одним из следующего: одной или более частотными таблицами, одной или более кодовыми таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более таких таблиц, с получением кодированных данных (E2);

(е) сжатие символов в одной или более таблиц с получением кодированных данных.

Преимущество настоящего изобретения заключается в том, что способ включает разбиение, то есть разделение, входных данных (D1) на один или более фрагментов данных и/или сжатие символов во входных данных (D1) таким образом, чтобы входные данные (D1) могли быть эффективно кодированы, например, с помощью способа, наилучшим образом подходящего для каждого фрагмента данных или сжатого символа, с использованием индексов ("indexes") и соответствующих одной или более таблиц, на которые указывают индексы.

Опционально, в этом способе по меньшей мере одна из одной или более таблиц заранее задана.

Опционально, способ включает доставку кодированных данных (E2), содержащих информацию одного или более наборов индексов, относящихся к символам в каждом фрагменте данных, вместе с по меньшей мере одним из следующего: одной или более частотными таблицами, одной или более кодовыми таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более таких таблиц, а также сжатыми символами.

Опционально способ включает генерирование одной или более таблиц для их восстановления во время декодирования на основе их симметрии и соответствующей информации, указывающей на симметрию.

Опционально способ включает доставку по меньшей мере одной из одной или более таблиц таким образом, чтобы по меньшей мере одна из одной или более таблиц сохранялась для последующего повторного использования.

Опционально, в этом способе информация одного или более наборов индексов включает:

- i) индексы символов, которые появляются во фрагменте, как таковом, и значения вероятности которых должны быть вставлены в кодированные данные (E2), после общего подсчета индексов во фрагменте данных; или
- ii) биты, показывающие, какие символы из всех возможных значений символов появляются во фрагменте, причем для этих символов значения вероятности должны быть вставлены в кодированные данные (E2); или
- iii) информацию, устанавливающую, начиная с начала, все вероятности значений символов, которые должны быть вставлены в кодированные данные (E2), которая сама по себе также выражает подсчет индексов.

Указанное выше разбиение, то есть разделение, опционально включает дополнительное подразделение входных данных (D1).

Опционально на шаге (а) обычно выполняется разделение, однако иногда также

необходимо сжать доступный фрагмент данных с помощью оптимальной кодовой таблицы без разделения данных на новые фрагменты данных. Кроме того, иногда исходные данные не разделяют, а вместо этого создают новые данные, например, посредством одного или более преобразований в один или более фрагментов данных, которые требуется наиболее эффективно сжать. Кодеры, реализуемые в соответствии с раскрытием настоящего изобретения, могут использоваться для создания различных фрагментов данных. Таким образом, следует принимать во внимание, что этот способ особенно хорошо подходит для кодеков видеосигнала и звукового сигнала, которые способны кодировать данные, содержащие фрагменты из различных временных интервалов. Различные кадры или секции представляют собой различные фрагменты данных, и они опционально могут разделяться на один или более фрагментов данных с использованием кодеров, реализующих один или более способов, соответствующих раскрытию настоящего изобретения. Все эти фрагменты данных могут повторно использовать любую ранее доставленную таблицу, например, в том же или в предшествующем кадрах.

Варианты осуществления настоящего изобретения позволяют эффективно доставлять кодовые или частотные таблицы. Это при передаче и/или хранении данных уменьшает объем служебной информации, требуемой для доставки и/или сохранения таблиц. Это также позволяет кодировать меньшие фрагменты данных путем использования кодовых таблиц, лучше оптимизированных для каждого отдельного фрагмента данных. Таким образом, можно достичь значительного увеличения эффективности сжатия, в результате чего уменьшается объем, требуемый для хранения данных, полоса частот передачи и энергопотребление.

Частоты различных частей данных часто взаимно различаются, и часто их относительная энтропия данных также взаимно различается, и по этим причинам выгодно разделять данные на множество частей, то есть фрагментов данных. Предпочтительно для отдельных частей используются различные кодовые таблицы в зависимости от характера данных и/или типа данных, и/или содержания данных, входящих в части данных; при этом под "характером" понимается одна или более характеристик и/или параметров данных. В рамках настоящего изобретения предлагаются способы, которые позволяют более эффективно разделить заданный большой файл данных на меньшие части, то есть на фрагменты данных, при этом дополнительное преимущество состоит в том, что доставка кодовой таблицы или частотной таблицы может быть оптимизирована для таких фрагментов данных. Такое разделение большого файла данных позволяет достичь существенного преимущества в том, что касается модификации энтропии задействованных данных, и поэтому данный способ позволяет значительно снизить объем кодированных данных, подлежащих передаче. Значения данных в одном или более фрагментах данных также могут разделяться, как было указано выше. Такое разделение, то есть разбиение, значений данных может быть реализовано, например, путем взаимного отделения MSB (Most Significant Bits, старшие биты) и LSB (Least Significant Bits, младшие биты). Значения данных также опционально разделяются на более чем два отдельных фрагмента значений данных.

Опционально способ включает применение одного или более алгоритмов сжатия данных, выполняемых на шаге (d), для генерации кодированных данных (E2). Кроме того, опционально согласно способу к одному или более алгоритмам сжатия данных относится по меньшей мере один из следующих алгоритмов: кодирование по Хаффману, VLC, энтропийное кодирование, арифметическое кодирование, интервальное

кодирование, однако способы кодирования не ограничены приведенными алгоритмами.

Опционально способ включает разделение входных данных (D1) на множество фрагментов данных и применение параллельной архитектуры процессоров для по существу параллельной обработки множества фрагментов данных.

5 Опционально способ включает генерацию одного или более наборов индексов на основе множества значений данных, которые объединяются друг с другом. Кроме того, опционально согласно способу индексы выводятся из одного или более RGB-пикселей или YUV-пикселей, содержащих значения пикселей R, G и B или значения пикселей Y, U и V.

10 Кроме того, опционально способ включает динамическое переключение между объединением фрагментов данных в некодированном виде или в виде кодированных данных (E2), в зависимости от достижимого коэффициента сжатия фрагментов данных при включении их в кодированные данные (E2).

Опционально способ включает ввод в кодированные данные (E2) по меньшей мере 15 одного завершающего бита для указания на то, что символ относится к "изменению кодовой таблицы" или к "концу данных".

Опционально способ включает генерацию для заданного фрагмента данных по существу только достаточного количества индексов, требуемых для ссылки на один или более символов, представленных в заданном фрагменте данных.

20 Опционально доставляемые кодовые таблицы также сжимаются, например, путем применения кодирования по Хаффману, и такой способ сжатия кодовых таблиц также опционально выполняется со связанными с ним одной или более таблицами. Также опционально, в этом способе сжатие одной или более кодовых таблиц использует одну или более вспомогательную кодовую таблицу.

25 Опционально способ включает доставку по меньшей мере одной из одной или более таблиц так, что по меньшей мере одна из одной или более таблиц может быть использована в декодере для декодирования переданных впоследствии данных.

Опционально способ включает шаг включения в кодированные данные (E2) одного или более идентификационных кодов, указывающих, откуда доступны одна или более 30 кодовых таблиц через одну или более баз данных и/или одну или более баз данных на прокси-сервере.

Опционально способ предназначен для кодирования одного или более следующих типов данных: захваченные звуковые сигналы, захваченные видеосигналы, захваченные изображения, текстовые данные, сейсмографические данные, сигналы датчиков, данные, 35 преобразованные из аналоговой в цифровую форму (ADC), данные биомедицинского сигнала, календарные данные, экономические данные, математические данные, двоичные данные.

Опционально способ включает сжатие одной или более кодовых таблиц для вставки в кодированные данные (E2). Опционально, одну или более кодовых таблиц сжимают 40 с помощью кодирования по Хаффману.

В соответствии со вторым аспектом предлагается кодер, предназначенный для кодирования входных данных (D1) с целью генерации соответствующих кодированных данных (E2), содержащий:

(a) анализатор для анализа символов, представленных во входных данных (D1), и 45 разделения и/или преобразования входных данных (D1) с получением одного или более фрагментов данных;

(b) генератор для генерации, в зависимости от появления символов, по меньшей мере одного из следующего: одной или более кодовых таблиц, одной или более частотных

таблиц, одной или более таблиц длин кодовых слов, одной или более таблиц вероятностей для символов, представленных в одном или более фрагментах данных;

(с) процессор для вычисления одного или более наборов индексов, связывающих символы в каждом фрагменте данных с элементами по меньшей мере в одном из следующего: в одной или более кодовых таблицах, в одной или более частотных таблицах, в одной или более таблицах длин кодовых слов, в одной или более таблицах вероятностей;

(d) сборщик данных для объединения информации одного или более наборов индексов, относящихся к символам в каждом фрагменте данных, с по меньшей мере одним из следующего: одной или более частотными таблицами, одной или более кодовыми таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более таких таблиц, с получением кодированных данных (E2); и

(е) компрессор для сжатия символов в одной или более таблиц с получением кодированных данных (E2).

Такое разделение, то есть разбиение, опционально включает подразделение входных данных (D1).

Опционально на шаге (а) обычно выполняется разделение, однако иногда также необходимо сжать доступный фрагмент данных с помощью оптимальной кодовой таблицы без разделения данных на новые фрагменты данных. Кроме того, иногда исходные данные не разделяются, а вместо этого создаются новые данные, например, посредством одного или более преобразований в один или более фрагментов данных, которые требуется наиболее эффективно сжать. Кодеры, реализуемые в соответствии с раскрытием настоящего изобретения, могут использоваться для создания различных фрагментов данных. Таким образом, следует принимать во внимание, что этот способ в особенности хорошо подходит для кодеков видеосигнала и звукового сигнала, которые кодируют данные, содержащие фрагменты из различных временных интервалов. Различные кадры или секции представляют собой различные фрагменты данных, и они опционально могут разделяться на один или более фрагментов данных с использованием кодеров, реализующих способ, соответствующий раскрытию настоящего изобретения. Все эти фрагменты данных могут повторно использовать любую таблицу, доставленную ранее, например, в том же или в предшествующем кадрах.

Опционально кодер способен осуществлять доставку кодированных данных (E2), содержащих информацию одного или более наборов индексов, относящихся к символам в каждом фрагменте данных, вместе с по меньшей мере одним из следующего: одной или более частотными таблицами, одной или более кодовыми таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более таких таблиц, а также сжатыми символами

Опционально кодер способен осуществлять доставку по меньшей мере одной из одной или более таблиц так, что по меньшей мере одна из одной или более таблиц может быть сохранена для последующего повторного использования.

Опционально кодер способен применять один или более алгоритмов сжатия данных в сборщике данных для генерации кодированных данных (E2). Кроме того, опционально в кодере к одному или более алгоритмам сжатия данных относится по меньшей мере один из следующих алгоритмов: кодирование по Хаффману, VLC, энтропийное кодирование, арифметическое кодирование, интервальное кодирование.

Опционально кодер способен разделять входные данных (D1) на множество

фрагментов данных и применять параллельную архитектуру процессоров для по  
существо параллельной обработки множества фрагментов данных.

Опционально генератор в кодере способен генерировать один или более наборов  
индексов на основе множества значений данных, которые объединяются друг с другом.  
5 Кроме того, опционально в кодере индексы выводятся из одного или более RGB-  
пикселей или YUV-пикселей, содержащих значения пикселей R, G и B или значения  
пикселей Y, U и V. Кроме того, опционально кодер способен динамически осуществлять  
переключение между объединением фрагментов данных в некодированном виде или в  
10 кодированном виде в кодированные данные (E2), в зависимости от достижимого  
коэффициента сжатия фрагментов данных при включении их в кодированные данные  
(E2). Опционально кодер способен встраивать в кодированные данные (E2) по меньшей  
мере один завершающий бит для указания на то, что символ относится к "изменению  
кодовой таблицы" или к "концу данных".

Опционально генератор в кодере способен генерировать для заданного фрагмента  
15 данных по существу только достаточное количество индексов, требуемых для ссылки  
на один или более символов, представленных в заданном фрагменте данных.

Опционально доставляемые кодовые таблицы сжимаются, например, путем  
применения кодирования по Хаффману, и для такого способа сжатия кодовых таблиц  
иногда опционально требуются свои собственные одна или более соответствующих  
20 кодовых таблиц.

Опционально доставляемые кодовые таблицы могут повторно использоваться в том  
же кадре данных или в следующих кадрах данных, то есть при кодировании фрагментов  
данных может повторно использоваться любая ранее доставленная таблица для другого  
фрагмента данных в этом или в предшествующих кадрах данных.

Опционально при реализации кодера предпочтительно применяются оптимальные  
25 реализации для доставки таблицы, например, в кодированных данных или путем  
включения в кодированные данные одного или более идентификационных кодов,  
указывающих, откуда можно получить доступ к таблице, например, из одной или более  
баз данных, одной или более баз данных прокси-сервера и т.д.

Опционально для обеспечения более эффективного кодирования данных, доставки  
30 кодированных данных и декодирования кодированных данных предпочтительно  
требуется, чтобы доставленные и/или ссылочные таблицы, как указано выше, хранились,  
например, для последующего использования, например, при доставке индекса  
сохраненной таблицы. С помощью такого подхода можно уменьшить объем данных,  
35 которые, например, требуются для передачи из кодера в соответствующий декодер в  
соответствии с раскрытием настоящего изобретения.

Согласно третьему аспекту, предлагается компьютерное программное изделие,  
содержащее машиночитаемый носитель информации, на котором хранятся  
машиночитаемые инструкции, исполняемые вычислительным устройством, в состав  
40 которого входят аппаратные средства обработки для выполнения способа,  
соответствующего первому аспекту.

Согласно четвертому аспекту предусматривается способ декодирования  
кодированных данных (E2), сгенерированных кодером, соответствующим второму  
аспекту; при этом способ, выполняемый в декодере для декодирования кодированных  
данных (E2), сгенерированных кодером, для генерации соответствующих  
45 декодированных данных (D3), включает:

(i) прием кодированных данных (E2) и выделение из них одного или более наборов  
индексов вместе с по меньшей мере одним из следующего: одной или более кодовыми

таблицами, одной или более частотными таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более из таких таблиц;

(ii) вычисление, на основе одного или более наборов индексов, соответствующих символам в по меньшей мере одном из следующего: одном или более фрагментах данных, сжатых символов элементов в одной или более кодовых таблицах, одной или более частотных таблицах, одной или более таблицах длин кодовых слов, одной или более таблицах вероятностей;

(iii) восстановление из символов одного или более фрагментов данных с использованием информации из по меньшей мере одного из следующего: одной или более кодовых таблиц, одной или более частотных таблиц, одной или более таблиц длин кодовых слов, одной или более таблиц вероятностей; и

(iv) объединение и/или преобразование одного или более фрагментов данных для генерации декодированных данных (D3).

Опционально в этом способе по меньшей мере одна из одной или более таблиц заранее задана.

Опционально способ включает перекодирование декодированных данных (D3) для генерации соответствующих перекодированных данных (D4) и/или генерации соответствующих перекодированных данных (D4) из кодированных данных (E2).

Опционально способ включает восстановление одной или более таблиц на основе их симметрии и соответствующей информации, указывающей на симметрию.

Опционально способ включает прием по меньшей мере одной или более таблиц таким образом, чтобы по меньшей мере одна или более таблиц сохранялась для последующего повторного использования.

Опционально способ включает применение одного или более алгоритмов распаковки данных, выполняемых на шаге (iv), для генерации декодированных данных (D3). Кроме того, опционально, к одному или более алгоритмам распаковки данных относится по меньшей мере один из следующих алгоритмов: декодирование по Хаффману, декодирование VLC, энтропийное декодирование, арифметическое декодирование, интервальное декодирование.

Опционально способ включает объединение множества, состоящего из одного или более фрагментов данных, и генерацию декодированных данных (D3) путем применения параллельной архитектуры процессоров для по существу параллельной обработки множества фрагментов данных.

Опционально способ включает генерацию одного или более наборов индексов на основе множества значений данных, которые объединяются друг с другом. Кроме того, опционально согласно способу индексы выводятся из одного или более RGB-пикселей или YUV-пикселей, содержащих значения пикселей R, G и B или значения пикселей Y, U и V. Кроме того, опционально способ включает динамическое переключение между генерацией фрагментов данных в некодированном виде или в кодированном виде в кодированные данные (E2), в зависимости от достижимого коэффициента распаковки фрагментов данных при включении их в кодированные данные (E2).

Опционально согласно способу декодер способен выделять из кодированных данных (E2) по меньшей мере один завершающий бит, указывающий на то, что символ относится к "изменению кодовой таблицы" или к "концу данных".

Опционально способ включает генерацию заданного фрагмента данных по существу только из достаточного количества индексов, требуемых для ссылки на один или более символов, представленных в заданном фрагменте данных.

Опционально способ включает распаковку одной или более кодовых таблиц, внесенных в кодированные данные (E2). Кроме того, опционально способ включает распаковку одной или более кодовых таблиц путем декодирования по Хаффману. Кроме того, опционально согласно способу при распаковке одной или более кодовых таблиц применяется одна или более вспомогательных кодовых таблиц.

Опционально способ включает прием одной или более кодовых таблиц таким образом, чтобы одна или более кодовых таблиц использовалась в декодере для декодирования переданных впоследствии данных.

Опционально способ включает выделение из кодированных данных (E2) одного или более идентификационных кодов, указывающих, откуда допускается доступ к одной или более кодовых таблиц через одну или более баз данных и/или одну или более баз данных на прокси-сервере.

Опционально способ включает декодирование одного или более следующих типов данных: захваченные звуковые сигналы, захваченные видеосигналы, захваченные изображения, текстовые данные, сейсмографические данные, сигналы датчиков, данные, преобразованные из аналоговой в цифровую форму (ADC, analog-to-digital), данные биомедицинского сигнала, календарные данные, экономические данные, математические данные, двоичные данные.

Опционально способ включает прием кодированных данных (E2) из множества источников данных и объединение данных, полученных из источников, для восстановления кодированных данных (E2).

Согласно пятому аспекту предлагается компьютерное программное изделие, содержащее машиночитаемый носитель информации, на котором хранятся машиночитаемые инструкции, исполняемые вычислительным устройством, в состав которого входят аппаратные средства обработки для выполнения способа, соответствующего четвертому аспекту.

Согласно шестому аспекту, предлагается декодер для декодирования кодированных данных (E2), сгенерированных кодером, соответствующим второму аспекту; при этом декодер, выполняющий декодирование кодированных данных (E2), сгенерированных кодером, для генерации соответствующих декодированных данных (D3), способен:

(i) принимать кодированные данные (E2) и выделять из них информацию одного или более наборов индексов вместе с по меньшей мере одним из следующего: одной или более кодовыми таблицами, одной или более частотными таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более из этих таблиц;

(ii) вычислять на основе одного или более наборов индексов соответствующие символы в по меньшей мере одном из следующего: одном или более фрагментах данных, сжатых символах элементов в одной или более кодовых таблицах, одной или более частотных таблицах, одной или более таблицах длин кодовых слов, одной или более таблицах вероятностей;

(iii) восстанавливать из символов один или более фрагментов данных с использованием информации из по меньшей мере одного из следующего: одной или более кодовых таблиц, одной или более частотных таблиц, одной или более таблиц длин кодовых слов, одной или более таблиц вероятностей; и

(iv) объединять и/или преобразовывать один или более фрагментов данных для генерации декодированных данных (D3).

Опционально декодер также содержит транскодер для перекодирования декодированных данных (D3) для генерации соответствующих перекодированных



данных (D4) и/или генерации соответствующих перекодированных данных (D4) на основе кодированных данных (E2).

5 Опционально декодер способен восстанавливать одну или более таблиц на основе их симметрии и соответствующей информации, указывающей на симметрию и предоставляемой декодеру.

Опционально декодер способен принимать по меньшей мере одну из одной или более таблиц таким образом, чтобы по меньшей мере одна из одной или более таблиц сохранялась для последующего повторного использования.

10 Опционально декодер способен применять один или более алгоритмов распаковки данных, выполняемых на шаге (iv), для генерации декодированных данных (D3). Кроме того, опционально в декодере выполняется по меньшей мере один или более следующих алгоритмов распаковки данных: декодирование по Хаффману, декодирование VLC, энтропийное декодирование, арифметическое декодирование, интервальное декодирование.

15 Опционально декодер способен объединять множество, состоящее из одного или более фрагментов данных, для генерации декодированных данных (D3) путем применения параллельной архитектуры процессоров для по существу параллельной обработки множества фрагментов данных.

20 Опционально декодер способен генерировать один или более наборов индексов на основе множества значений данных, которые объединяются друг с другом. Кроме того, опционально в декодере индексы выводятся из одного или более RGB-пикселей или YUV-пикселей, содержащих значения пикселей R, G и B или значения пикселей Y, U и V. Кроме того, опционально декодер способен осуществлять динамическое переключение между генерацией фрагментов данных в некодированном виде или в кодированном виде в кодированные данные (E2), в зависимости от достижимого коэффициента распаковки фрагментов данных при включении их в кодированные данные (E2).

30 Опционально декодер способен выделять из кодированных данных (E2) по меньшей мере один завершающий бит, указывающий на то, что символ относится к "изменению кодовой таблицы" или к "концу данных".

Опционально декодер способен генерировать заданный фрагмент данных по существу только из достаточного количества индексов, требуемых для ссылки на один или более символов, представленных в заданном фрагменте данных.

35 Опционально декодер способен распаковывать одну или более кодовых таблиц, включенных в кодированные данные (E2). Кроме того, опционально декодер способен распаковывать одну или более кодовых таблиц путем декодирования по Хаффману. Кроме того, опционально в декодере при распаковке одной или более кодовых таблиц применяется одна или более вспомогательных кодовых таблиц.

40 Опционально декодер способен принимать одну или более кодовых таблиц таким образом, чтобы одна или более кодовых таблиц использовалась в декодере (60) для декодирования переданных впоследствии данных.

45 Опционально декодер способен выделять из кодированных данных (E2) один или более идентификационных кодов, указывающих, откуда допускается доступ к одной или более кодовых таблиц через одну или более баз данных и/или одну или более баз данных на прокси-сервере.

Опционально декодер способен декодировать один или более следующих типов данных: захваченные звуковые сигналы, захваченные видеосигналы, захваченные изображения, текстовые данные, сейсмографические данные, сигналы датчиков, данные,

преобразованные из аналоговой в цифровую форму (ADC), данные биомедицинского сигнала, календарные данные, экономические данные, математические данные, двоичные данные.

5 Опционально декодер способен принимать кодированные данные (E2) из множества источников данных и объединять данные, полученные из источников, для восстановления кодированных данных (E2).

В соответствии с седьмым аспектом предлагается кодек, который содержит по меньшей мере один кодер, соответствующий второму аспекту, для кодирования входных данных (D1) с целью генерации соответствующих кодированных данных (E2) и по  
10 меньшей мере один декодер, соответствующий шестому аспекту, для декодирования кодированных данных (E2) с целью генерации декодированных выходных данных (D3).

Опционально кодек реализован таким образом, что по меньшей мере один кодер и по меньшей мере один декодер пространственно разделены друг с другом и связаны друг с другом через сеть передачи данных. Кроме того, опционально кодек реализован  
15 таким образом, что сеть передачи данных сконфигурирована как одноранговая сеть связи. Кроме того, опционально кодек реализован таким образом, что его кодер и декодер симметрично выполняют свои функции в процессе обработки данных; другими словами, функции обработки, выполняемые в кодере, в декодере реализованы в виде соответствующих обратных функций, выполняемых в обратном порядке.

20 Следует принимать во внимание, что признаки настоящего изобретения допускается комбинировать в различных сочетаниях без нарушения объема настоящего изобретения, определенного прилагаемой формулой изобретения.

#### КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Варианты осуществления, приведенные в настоящем раскрытии изобретения,  
25 описываются ниже только на примерах со ссылкой на прилагаемые чертежи, на которых:  
на фиг. 1 показаны известные кодер и декодер, предназначенные для кодирования и декодирования данных;

на фиг. 2 показан способ кодирования данных в соответствии с вариантом раскрытия настоящего изобретения;

30 на фиг. 3А показан вариант осуществления кодера и декодера, вместе образующих кодек, в соответствии с раскрытием настоящего изобретения; и

на фиг. 3В показан альтернативный вариант реализации кодера и декодера, вместе образующих кодек, в соответствии с раскрытием настоящего изобретения, в котором декодированные данные D3 в декодере перекодируются для генерации  
35 перекодированных данных D4.

На прилагаемых чертежах подчеркнутые числа используются для обозначения элементов, поверх которых расположены эти числа, или элементов, рядом с которыми находятся эти числа. Если число не подчеркнуто, но рядом с ним расположена стрелка, то неподчеркнутое число используется для идентификации общего элемента, на который  
40 указывает стрелка.

#### ОПИСАНИЕ ВАРИАНТОВ ОСУЩЕСТВЛЕНИЯ ИЗОБРЕТЕНИЯ

В целом раскрытие настоящего изобретения относится, например, к кодерам, декодерам, кодерам и связанным с ними способами функционирования. Кроме того, варианты раскрытия настоящего изобретения позволяют на практике повысить  
45 эффективность доставки кодовых таблиц, частотных таблиц, таблиц длин кодовых слов или таблиц вероятностей по сравнению с известными способами. Кроме того, варианты раскрытия настоящего изобретения также позволяют осуществлять доставку одного или более деревьев Хаффмана таким образом, чтобы использовалось меньшее

количество битов для доставки одной или более таблиц, благодаря чему обеспечивается  
 5 больший коэффициент сжатия данных, достижимый в процессе кодирования данных,  
 в особенности если объем кодированных данных вместе с одной или более таблицами  
 относительно мал. Кодовые таблицы, частотные таблицы, таблицы длин кодовых слов  
 или таблицы вероятностей требуются для множества различных способов энтропийного  
 кодирования, например: для способов кодирования переменной длины (VLC), таких  
 как кодирование по Хаффману, арифметическое кодирование, интервальное  
 кодирование и других способов. Как кодеры, например передатчики, так и декодеры,  
 например приемники, предпочтительно применяют способы, описываемые ниже.

10 Варианты раскрытия настоящего изобретения, описываемые ниже, относятся к  
 области, в которой объем сохраняемых и передаваемых данных со временем быстро  
 возрастает. Для хранения и передачи таких данных требуется значительный объем  
 запоминающих устройств, широкая полоса частот и высокий уровень энергии. Большая  
 15 часть данных в этой области представляет собой, помимо прочего, захваченные звуковые  
 сигналы, захваченные видеосигналы, захваченные изображения, текстовые данные,  
 сейсмографические данные, данные сигналов датчиков, данные, преобразованные из  
 аналоговой в цифровую форму (ADC), данные биомедицинского сигнала, календарные  
 данные, экономические данные, математические данные, двоичные данные. Варианты  
 раскрытия настоящего изобретения позволяют снизить объем кодированных данных  
 20 для всех указанных типов данных, а также для других типов данных, что позволяет  
 эффективно доставлять кодовые таблицы, частотные таблицы, таблиц длин кодовых  
 слов или таблицы вероятностей путем использования меньших фрагментов данных,  
 что эффективно уменьшает энтропию данных, то есть размер данных.

Кроме того, небольшие фрагменты данных могут эффективней обрабатываться с  
 25 использованием параллельных процессов для более быстрого вывода результатов, и  
 такой параллелизм является общим средством в современной микропроцессорной  
 архитектуре, в особенности в разрабатываемых конфигурациях микропроцессоров,  
 например в массивах процессоров данных и в высокоскоростных конфигурациях  
 процессоров RISC (компьютеров с сокращенным набором команд).

30 Для заданного способа кодирования соответствующая кодовая таблица содержит  
 информацию, указывающую длины кодовых слов, например, выраженные в битах,  
 коды для представления кодовых слов и индексы кодовых слов. Кодовая таблица также  
 может генерироваться на основе длин кодовых слов. Индексы кодовых слов  
 представляют значения соответствующих исходных символов, кодированных с  
 35 использованием кодовых слов. Подобным образом частотная таблица содержит частоты  
 появления символов и индексы символов. Индексы символов представляют значения  
 исходных символов, кодированных, соответственно, индексами. Частотная таблица  
 может быть преобразована в таблицу вероятностей, которая может использоваться в  
 качестве приблизительной оценки частотной таблицы. Также может выполняться  
 40 двухстороннее преобразование частотной таблицы и длин кодовых слов.

При доставке указанных выше таблиц одним из очень важных параметров является  
 максимальный индекс таблицы. Максимальный индекс таблицы определяет количество  
 различных символов или количество возможных различных символов, доступных в  
 доставленной таблице, а также - во входных данных. Например, если заданы следующие  
 45 данные:

4, 3, 0, 1, 0, 4, 3, 4,

то в данном примере максимальный индекс равен 4, а минимальный индекс равен  
 0, это означает, что в данных потенциально содержатся 5 (максимальное значение -

минимальное значение  $+1=4-0+1$  различных символов (0, 1, 2, 3, 4). С учетом того, что в наборе данных в действительности существуют только четыре различных символа (0, 1, 3, 4), таблица опционально также доставляется с использованием значения '3' в качестве максимального индекса, а именно: количества доступных различных символов, вместо значения '4', а именно: количества возможных различных символов. Если значение 3 используется в качестве максимального значения индекса таблицы, то некоторый другой механизм требуется для доставки информации, касающейся символов, используемых для индексов каждой таблицы.

Если символы упорядочены, опционально доставляется фактический максимальный индекс (4) и биты доступности, например 11011 в данном примере. В результате такой доставки выполняется преобразование вида "индекс таблицы 0 = символ 0", который помечен как индекс, при этом связанная пара символов равна (0, 0). Таким же образом, оставшиеся пары индексов и символов представляют собой следующую последовательность: (1, 1), (2, 3) и (3, 4). Кроме того, опционально возможно использовать пары индексов и символов непосредственно с целью определения используемых индексов для различных индексов таблицы и затем - для доставки максимального индекса таблицы, равного '3'. Этот способ очень полезен, например, если символы в доставляемой таблице отсортированы на основе своих частот.

Например, в указанном выше случае пары индексов и символов представляют собой следующую последовательность: (0, 4), (1, 0), (2, 3) и (3, 1). Иногда используемые пары индексов и символов задаются предварительно, в то время как в других случаях доставляется индекс используемой таблицы пар символов и индексов. В других различных ситуациях пары индексов и символов доставляются вместе с кодированными данными (E2). В некоторой другой ситуации декодер 60 способен извлекать используемые пары индексов и символов из известного местоположения; а в некоторой отличной ситуации декодер 60 способен извлекать используемые пары индексов и символов из местоположения, информация о котором доставлена вместе с кодированными данными (E2).

В некоторых ситуациях выгодно аппроксимировать таблицу, подлежащую передаче, с помощью симметрии. С помощью симметрии можно доставлять подлежащую использованию кодовую таблицу с помощью меньшего объема данных по сравнению с объемом, требуемым для доставки всей таблицы, независимо от того, основана ли таблица на длинах кодов, вероятностях или частотах появления. Кроме того, генерация симметричной таблицы как в кодере 50, так и в декодере 60 выполняется быстрее, поскольку такая таблица содержит меньше элементов, то есть меньшее количество символов. Однако при использовании симметрии таблица становится менее оптимальной, но экономия, достигаемая в размере данных доставляемой таблицы, полностью или в значительной степени компенсирует эту потерю, в особенности, если передается достаточно небольшой объем данных.

В последнем приведенном выше примере опционально используется симметрия при доставке таблицы, поскольку значение '0' символа более вероятно, чем значение '1', и соответственно значение '4' символа более вероятно, чем значение '3'. Кроме того, вероятности значений '0' и '4' символов близки друг к другу, и, соответственно, вероятности значений '1' и '3' символов также близки друг к другу. Однако значение '2' вообще отсутствует в данных, поэтому - это самое маловероятное значение, независимо, в каком направлении осуществляется его проверка.

При использовании симметрии таблица кодирования может, таким образом, всегда генерироваться на основе сумм появлений симметрично соответствующих элементов.

В таком случае в целом существует 5 появлений значений '0' и '4' символов, и, соответственно, в совокупности 3 появления значений '1' и '3' символов. Элемент '2' вообще отсутствует в данных, и, таким образом, необходимость задавать для него собственный символ отсутствует. Однако в некоторых ситуациях для элемента '2' также может генерироваться символ, и в этом случае он должен включаться как в правостороннюю, так и в левостороннюю таблицы при использовании симметрии.

Таким образом, таблица интервального кодирования опционально генерируется таким образом, чтобы количество появлений для интервалов составляло 5, 3 и (0), и, следовательно, будет использоваться интервальная таблица 5, 3, 0, 3, 5 для символов от нуля до четырех, хотя оптимальная интервальная таблица, основанная на частоте появлений, безусловно имеет следующий вид: 3, 2, 0, 1, 2, который требует доставки четырех значений частоты появлений, по сравнению с приведенной выше опциональной реализацией, основанной на симметрии, в которой требуется передавать, то есть доставлять, только два значения частоты появления.

Та же самая идея, основанная на симметрии, опционально используется в других способах кодирования, таких как кодирование по Хаффману, и в этом случае основанной на симметрии таблицей может быть, например, таблица, в которой левосторонние значения принимают код '0', а правосторонние значения принимают код '1'. Таким образом, кодовые слова Хаффмана, например, могут быть представлены следующим образом: 01, 00, не существует, 10, 11. В альтернативном варианте, если на будущее требуется зарезервировать позицию для значения '2', то коды выглядят следующим образом: 01, 001, 000/100, 101, 11. В этой реализации, в принципе, должны быть переданы/доставлены только две длины кодов (1 и 1) или, в противном случае, - три длины кодов (1, 2, 2), и используемая таблица полностью восстанавливается в декодере 60, если известно, что в таблице используется симметрия. В более длинных таблицах преимущества даже еще более очевидны.

Следует принимать во внимание, что информация о том, используется ли в таблице симметрия, в одном из вариантов уже известен заранее, или, в противном случае, она передается/доставляется таким же образом, как информация о том, какая таблица подлежит использованию: оптимальная или заранее заданная, или таблица, сгенерированная динамически из предшествующих таблиц. Доставка фрагмента информации об использовании в таблице симметрии выполняется путем передачи в декодер 60 индекса подлежащей использованию таблицы.

Например:

- (i) индекс '0' таблицы означает, что таблица полностью передается/доставляется;
- (ii) индекс '2' таблицы означает, что таблица симметрична и доставляется только половина таблицы;
- (iii) индексы 2...63 таблицы означают, что используется заранее заданная таблица;
- и
- (iv) индексы 64...127 таблицы означают, что используется предварительно доставленная и сохраненная динамическая таблица.

Следует принимать во внимание, что симметричные таблицы также могут применяться как заранее заданные или динамически сохраненные таблицы.

Опционально применяются различные способы кодирования, например, с использованием ODelta-кодирования или без него, при этом в случае ODelta-кодирования задействуется кодирование данных дифференциально в последовательность значений 0 и 1, и используется циклический перенос при подсчете. Кроме того, при использовании этих различных способов кодирования предпочтительно следует применять их в

совокупности с индексом используемой таблицы, а также опционально определять, выполнялся ли с данными способ ODelta перед кодированием, и должен ли декодер 60 соответственно выполнять обратную операцию после декодирования.

В таком случае, например, индексы таблицы в остальном одинаковы, но добавляется значение 128 (десятичное) для указания на то, что использовалось ODelta-кодирование. Если такая вставка не выполнена, то ODelta-кодирование перед кодированием данных не осуществлялось. Безусловно, к значению индекса таблицы могут также добавляться и другие значения; однако важно принимать во внимание, что индекс таблицы выражает, какой вид таблицы используется и какой вид дополнительных данных передается/ доставляется с индексом таблицы вместе с кодированными данными.

На фиг. 2 и фиг. 3А показаны шаги способа кодирования входных данных D1 для генерации соответствующих кодированных выходных данных E2; шаги способа указаны числами, кратными 10, и на этих шагах обычно выполняются следующие операции: на шаге 20 генерируется одна или более частотных таблиц 25, на шаге 30 генерируется одна или более кодовых таблиц 35, на шаге 40 выполняется анализ входных данных D1 для выбора наиболее подходящего способа кодирования, без ограничения этим способом. При реализации такого способа 10 кодирования входных данных D1 должен существовать некоторый доступный механизм, который изменяет один или более символов во входных данных D1 на соответствующие индексы, например, заданный индекс, включенный в индексы, равен значению пикселя, например, в изображении, содержащем массив пикселей. Индекс также может равняться значению пикселя минус наименьшее значение пикселя, присутствующего в изображении, содержащем массив пикселей. В такой ситуации для выполнения способа также требуется доставка наименьшего значения пикселя в кодированных выходных данных E2, подлежащих доставке некоторым образом из кодера 50 в соответствующий декодер 60 путем выполнения способа 10 или путем применения обратного способа, поскольку в противном случае декодер 60 не способен декодировать соответствующий заданный символ в его первоначальное значение с помощью заданного для него индекса. Индекс также может создаваться на основе множественной информации, например, с помощью одного или более дискретных косинусных преобразований (DCT, Discrete Cosine Transform), отдельных коэффициентов AC, содержащих абсолютные значения коэффициентов AC, знаки коэффициентов AC, последовательность нулевых коэффициентов AC между ними и предыдущим ненулевым коэффициентом AC/DC и флаг индикации, представляющий информацию, относящуюся к текущему коэффициенту AC, и последний ненулевой коэффициент AC. Индекс также допускается создавать на основе множества значений пикселей, которые объединяются вместе, например, на основе 24-битного RGB-пикселя, который содержит 8-битные значения пикселей R, G и B, или на основе 10-битового значения, содержащего два 5-битовых значения пикселя Y.

На фиг. 3В показано перекодирование данных D3 в декодере 60 для генерации соответствующих перекодированных данных D4, опционально применяемое с помощью транскодера 70, например, в условиях многоадресной передачи, если:

- (i) существует множество устройств, каждое из которых оснащено декодером 60 для приема кодированных данных E2;
- (ii) по меньшей мере некоторые устройства из этого множества отличаются друг от друга и выводят данные в различных форматах, отличающихся, например, структурой отображения на экране, разрешением, форматом изображения, емкостью буфера драйвера дисплея и т.д.

Перекодирование данных D3 требуется для генерации соответствующих данных D4, которые могут согласованно обрабатываться аппаратурой и/или поддерживающими программными уровнями устройств, которые обычно основаны на вычислительной аппаратуре, и представляют собой, например, смартфоны, специальное научное оборудование, телевизионное оборудование, устройство высокой точности, устройство для видеоконференций и т.п. Транскодер 70 реализуется в виде программного обеспечения и/или специализированной аппаратуры обработки данных, например ASIC.

Как отмечается ниже, при реализации способа 10 всегда должен включаться шаг изменения одного или более значений символов на один или более соответствующих индексов, и этот шаг или его инверсия должен сообщаться декодеру 60, или, в противном случае, оба эти шага должны предварительно задаваться как для кодера 50, так и для декодера 60. Наиболее простой подход к реализации такой связи заключается в установлении прямой взаимосвязи между заданным значением символа и соответствующим ему значением индекса, например, в установке значения индекса, равного соответствующему значению пикселя или числу, получаемому на основе битов, представляющих следующую последовательность: S = флаг знака, V = 10-битовое значение коэффициента, R = 6-битовое значение ненулевой последовательности и L = флаг завершения последовательности, например:

**S V V V V V V V V V V R R R R R R L**

По многим причинам использовать непосредственную взаимосвязь не всегда представляется возможным, например, при использовании непосредственную взаимосвязи:

- (a) невозможно или неэффективно кодировать либо декодировать данные, индексы, частоты, вероятности или длины символов;
- (b) количество различных индексов становится слишком большим; или
- (c) не для всех символов доступна частотная информация, и по этой причине некоторые алгоритмы не способны генерировать коды для этих символов.

Некоторые такие проблемы, (a)-(c), могут, по меньшей мере частично, решаться с использованием управляющих кодов или некоторой логической схемы, генерирующей частотную информацию для всех символов. Достаточно часто по-прежнему предпочтительней использовать некоторый другой подход для преобразования символов в индексы. Согласно одному из подходов всегда обеспечивается наличие некоторой справочной таблицы (LUT, Look-Up-Table), в которой указаны индексы, используемые для доступных символов; в данном случае для уменьшения размера кодовых таблиц гораздо предпочтительней использовать управляющий код. LUT должна быть доступна в кодере 50 и в декодере 60, или она должна доставляться из кодера 50 в декодер 60, или наоборот. Если для обеспечения более высокой степени сжатия требуется более оптимальное кодирование, то предпочтительнее следует использовать множество таблиц, которые могут выбираться на основе индекса доступных LUT. Однако это иногда не практично, поскольку комбинации частоты или длины кодового слова столь велики, что не имеет смысла сохранять все различные таблицы в памяти данных, либо доставка таких LUT требует большого объема данных, передаваемых между кодером 50 и декодером 60. Таким образом, способы 10, соответствующие раскрытию настоящего изобретения, позволяют достичь эффективного способа передачи частот, длин кодовых слов или вероятностей из кодера 50 в декодер 60 с использованием одного или более подходящих процессов преобразования символов в индексы. Всегда предпочтительней использовать длину кодовых слов, а не частоты, если с помощью способа кодирования невозможно применять более точную информацию, предоставляемую посредством

частотной информации, например, способы кодирования VLC не могут использовать частотную информацию, однако, в отличие от них, способы арифметического и интервального кодирования способны использовать частотную или вероятностную информацию.

5 Далее более подробно описывается пример осуществления, соответствующий раскрытию настоящего изобретения, в котором в целях кодирования применяется длина кодовых слов; однако также допустимо соответствующее осуществление, согласно которому применяется частотная или вероятностная информация.

10 В кодированные данные E2, показанные на фиг. 2, например в кодированный поток данных, включаются значения данных, которые могут соответствовать 20 символам, то есть значения от 0 до 19, но только 8 значений этих данных, а именно: минимальное значение = 2 и максимальное значение = 19, фактически доступны в текущем потоке данных. Эти минимальное и максимальное значения, как описывается далее, опционально также доставляются отдельно для обеспечения дополнительной экономии при доставке таблицы. Соответственно, при том что сумма частот равна 148, 15 минимальная длина кодовых слов составляет 1, а максимальная длина равна 6, и индексы для символов, например, представлены ниже в Табл. 1, можно определить, что без сжатия этих символов потребуется  $148 * 5 \text{ бит} = 740 \text{ бит}$  для передачи битового потока.

Таблица 1: Пример кодирования битового потока

Значение	Частота	CWLen	Част.1 (биты)	Частота2	Част.3 (биты)	Индекс1	Индекс2
2	7	4	4(4)	84	128(4)	2	0
4	2	6	1(6)	24	32(6)	4	1
7	61	1	32(1)	972	1024(1)	7	2
9	1	6	1(6)	12	16(7)	9	3
12	35	2	16(2)	420	512(2)	12	4
13	9	4	4(4)	108	128(4)	13	5
14	5	5	2(5)	60	64(5)	14	6
19	8	4	4(4)	96	128(4)	19	7

Для режима кодирования, представленного в Табл. 1, если не существует подходящая доступная частотная или кодовая таблица, например, заранее заданная или указываемая с помощью ссылочного индекса в кодере 50 и в декодере 60, то существует несколько 35 способов, потенциально доступных для передачи этих необходимых кодовых слов, то есть длин и кодов, из кодера 50 в декодер 60.

В первом примере способ модифицирует частоты данных, а затем генерирует соответствующую кодовую таблицу, при этом наименее вероятному символу, то есть наиболее длинному кодовому слову, выделяется один дополнительный бит, и всем 40 символам, которые не доступны в данных, выделяются настолько большие длины кодового слова, что они не влияют на кодирование, но позволяют кодору 50 и декодеру 60 создавать одинаковые частотные таблицы и одинаковое дерево Хаффмана. С учетом 12 отсутствующих символов такая модификация частот может быть реализована, например, путем умножения исходных частот на 12 и установки значения частоты, 45 равным 1, для всех символов, не имеющих какого-либо действительного значения частоты, то есть имеющих частоту = 0. Модифицированные значения частот представлены, например, в столбце "Частота 2" в Табл. 1 для всех символов с доступными частотами. На основе этих новых частот длины кодовых слов для всех 20



символов могут быть представлены следующим образом:

11, 11, 4, 11, 6, 11, 11, 1, 11, 7, 11, 11, 2, 4, 5, 10, 10, 10, 10, 4

Согласно такой кодовой таблице требуется  $(7*4+2*6+81*1+1*7+35*2+9*4+5*5+8*4)=291$  бит для доставки кодированных символов из кодера 50 в декодер 60. Для корректного кодирования данных и последующего их декодирования кодер 50 и декодер 60 должны использовать одинаковые частоты, при этом такие частоты могут создаваться на основе этих новых длин кодовых слов, и результат представлен в столбце "Част3" ( $=2^{(\text{maxbitlen} - \text{bitlen})}$ ) для символов с частотой; где "maxbitlen" является аббревиатурой от "maximum bit length" (максимальная длина в битах), а "bitlen" является аббревиатурой от "bit length" (длина в битах). Другим символам назначаются частоты 2 (bitlen = 10) или 1 (bitlen = 11).

Первый способ позволяет генерировать кодовую таблицу, которая содержит все возможные символы, то есть 20 символов в приведенном выше примере, при этом преимущество достигается в том случае, если одинаковая кодовая таблица используется также для других подобных типов данных. Такие виды длин также могут дополнительно сжиматься с использованием любого способа сжатия без какой-либо дополнительной информации, и поэтому кодовую таблицу легко доставлять между кодером 50 и декодером 60 потенциально во всех ситуациях. Например, без сжатия для этой кодовой таблицы требуется 4 бита для представления всех длин кодовых слов  $\Rightarrow 20$  длин \* 4 бита / длина = 80 бит.

Этот первый способ неэффективен по сравнению с оптимальными кодами Хаффмана, длиной только 1 бит для каждого наименее вероятного символа, то есть = 1 бит в этом первом способе из примера. Кроме того, существуют также такие биты, которые требуются для сжатия и доставки длин кодовых слов, то есть - кодовой таблицы, из кодера 50 в декодер 60. Число 20, а именно: количество всех возможных символов, доставляется дополнительно, либо оно также может быть известно декодеру 60.

Далее согласно приведенному примеру описывается второй способ для иллюстрации альтернативного осуществления настоящего изобретения. С помощью второго способа генерируются длины кодовых слов только для доступных символов, то есть для тех символов, для которых значение частоты  $>0$ . Индекс, используемый для генерации кодовой таблицы Хаффмана, приведен в столбце "Индекс 2" (см. Табл. 1), но индекс, который должен быть доставлен из кодера 50 в декодер 60, приведен в столбце "Индекс 1", то есть = значение символа (см. Табл. 1). Такие сгенерированные длины кодовых слов указаны в столбце "CWLen" (см. Табл. 1). С использованием кодовой таблицы такого вида требуется  $(7*4+2*6+81*1+1*6+35*2+9*4+5*5+8*4)=290$  бит для доставки кодированных символов. На основе этих длин кодовых символов кодер 50 и декодер 60 способны создавать таблицу с частотами, представленными в столбце "Част. 1" в Табл. 1.

Согласно первому способу доставки кодовых таблиц такого вида доставляются длина и индекс кодового слова в виде пары чисел, представленных следующим образом: (2, 4), (4, 6), (7, 1), (9, 6), (12, 2), (13, 4), (14, 5), (19, 4), где пары заключены в скобки.

Для такого способа доставки требуется 5 бит для каждого индекса и 3 бита для каждой длины  $\Rightarrow 8$  бит для каждой пары, и всего  $8*8$  бит = 64 бита.

Для этих индексов также может использоваться дельта-кодирование, и тогда пары выглядят следующим образом:

(2, 4), (2, 6), (3, 1), (2, 6), (3, 2), (1, 4), (1, 5), (5, 4)

Теперь следует принимать во внимание, что для передачи кодовой таблицы требуется

только 3 бита для каждого индекса и 3 бита для каждой длины кодового слова, то есть 6 бит для каждой пары и в целом:  $8 \cdot 6 \text{ бит} = 48 \text{ бит}$ .

Эти значения индексов и длин кодовых слов предпочтительно разделяются на собственные соответствующие потоки данных, что часто позволяет улучшить степень сжатия по сравнению с комбинированными 8-или 6-битными значениями. Теперь потоки выглядят следующим образом:

2, 4, 7, 9, 12, 13, 14, 19 и

4, 6, 1, 6, 2, 4, 5, 4

$\Rightarrow 8 \cdot 5 \text{ бит} + 8 \cdot 3 \text{ бита} = 64 \text{ бита}$  в целом.

Если для индексов первого потока применяется дельта-кодирование, то генерируются следующие потоки:

2, 2, 3, 2, 3, 1, 1, 5 и 4, 6, 1, 6, 2, 4, 5, 4

$\Rightarrow 8 \cdot 3 \text{ бита} + 8 \cdot 3 \text{ бита} = 48 \text{ бит}$  в целом.

Это наилучший, то есть оптимальный, способ доставки, при котором обеспечивается большое количество возможных символов, но данные содержат только небольшое количество взаимно различных символов.

Все эти упомянутые выше потоки данных могут сжиматься и доставляться из кодера 50 в декодер 60 в процессе функционирования. Этот способ не характеризуется неэффективностью по сравнению с оптимальными кодами Хаффмана, однако в рамках этого способа раскрытия настоящего изобретения все еще потребляется значительное количество битов для доставки потоков, содержащих информацию об индексах и длинах кодовых слов. Кроме того, должно доставляться значение 8, а именно: количество доступных символов, поскольку в противном случае декодер 60 не способен оценить, как много значений или пар должно декодироваться в кодовую таблицу. В этом случае число 20, то есть количество всех возможных символов, доставлять между кодером 50 и декодером 60 не требуется.

Некоторые описанные выше способы можно применять совместно для обеспечения особенно эффективного кодирования входных данных D1 в кодере 50 с целью генерации соответствующих кодированных данных E2, так же как и для выполнения обратного процесса в декодере 60. Опционально в рамках всех этих решений создаются коды Хаффмана только для доступных символов, например, способом, схожим со вторым способом. Кроме того, частотная таблица, сгенерированная на основе этих длин данных, указана в столбце "Част. 1" (см. Табл. 1).

По сравнению с первым способом второй способ также позволяет устанавливать нулевую длину кодового слова, если оно недоступно, однако по-прежнему необходимо доставлять такую информацию из кодера 50 в декодер 60; это возможно реализовать вследствие того, что фактическая длина кодового слова никогда не может быть нулевой. В результате формируется следующий поток длин:

0, 0, 4, 0, 6, 0, 0, 1, 0, 6, 0, 0, 2, 4, 5, 0, 0, 0, 0, 4,

в котором первоначально требуется только 3 бита для всех длин кодовых слов  $\Rightarrow 20 \cdot 3 \text{ бита} = 60 \text{ бит}$ .

Такие типы длин также могут сжиматься с использованием любого способа сжатия без любой дополнительной информации, и поэтому кодовую таблицу легко доставлять между кодером 50 и декодером 60 во всех ситуациях. Кроме того, эти данные также часто содержат многочисленные нулевые значения, и поэтому они могут легко сжиматься, например, с использованием VLC или RLE. Число 20, то есть количество всех возможных символов, дополнительно доставляется из кодера 50 в декодер 60, или оно должно быть известно декодеру 60 заранее.

В другом варианте осуществления используются биты для указания, какие символы данных доступны, а какие - нет. В ситуации, когда генерируются два потока данных, соответственно содержащих биты и длины кодовых слов, эти потоки выглядят следующим образом:

5 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1

и

4, 6, 1, 6, 2, 4, 5, 4,

при этом требуется 1 бит для каждого символа и 3 бита для каждой длины доступного кодового слова =>  $20 * 1 \text{ бит} + 8 * 3 \text{ бита} = 44 \text{ бита}$  в целом. Такой способ обычно является

10 наилучшим, то есть оптимальным, способом доставки.

Такие потоки далее сжимаются с использованием модификатора энтропии (EM, Entropy Modifier) и/или VLC. Как указано выше, число 20, то есть количество всех возможных символов, дополнительно доставляется между кодером 50 и декодером 60, или оно указывается кодеру 60 заранее. Кроме того, число 8, то есть количество

15 доступных символов, не требуется доставлять между кодером 50 и декодером 60, поскольку его можно вычислить на основе битов со значением 1.

Для всех ранее описанных способов, соответствующих раскрытию настоящего изобретения, кодер 50 и декодер 60 должны содержать информацию о том, сколько может использоваться различных символов, например 20 символов в приведенном

20 выше примере, или о том, сколько доступно различных символов, например 8 в приведенном выше примере. Если значение, то есть количество различных символов, недоступно в декодере 60, оно должно доставляться в декодер 60. Дополнительно некоторые данные могут сохраняться в таблице, которые доставляются путем передачи небольшого объема дополнительной информации, для указания диапазона, в котором

25 значения данных доступны, например, возможно доставлять значения 2 (минимальное) и 19 (максимальное) для указания диапазона, в который должны включаться значения.

В этом примере при такой доставке часто не удается сэкономить передаваемые биты, однако в ситуации, когда, например, 8-битовые пиксели содержат только значения от

30 60 до 200, достигается значительная экономия битов, передаваемых из кодера 50 в декодер 60. Доставка такого диапазона позволяет избавиться от необходимости передачи из кодера 50 в декодер 60 всех битов или значений, которые в противном случае

использовались бы для меньшего, чем наименьшее, значения, и большего, чем наибольшее, значение. Кроме того, следует принимать во внимание, что при доставке

35 диапазона из кодера 50 в декодер 60 не требуется доставлять значение первого и последнего индекса в ситуации, когда значения индекса в противном случае передавались бы с использованием дельта-кодирования или без такового. Тот же подход применим также к первому и последнему 1-битному значению в последнем примере. Доставка

минимального и максимального значений опционально также используется в случае применения других способов, например ODelta-кодирования, способа, раскрытого в

40 заявке на патент GB 1303661.1, поданной Gurulogic Microsystems Oy 1<sup>го</sup> марта 2013 года и включенной в настоящую заявку посредством ссылки. Основное преимущество доставки минимального и максимального значений достигается, если все способы кодирования с изменяющейся и реализуемой энтропией используют одинаковую

информацию, которая доставляется только один раз.

Способы, описанные выше, опционально применяются выборочно, например, в зависимости от того, как много символов должно быть кодировано в заданном

фрагменте данных, выделенном из общего набора данных, подлежащего передаче из кодера 50 в декодер 60. Таким образом, выбор ранее разьясненных способов зависит

45

от того, сколько доступно различных символов, сколько из них фактически используется, какова частота наименее вероятного символа, и сколько индексов доступных символов распределено через возможные символы.

На основе значений частот для символов, приведенных в Табл. 1, также могут  
 5 рассчитываться масштабированные вероятности. Пусть, например, используется 148  
 символов. Масштабированные вероятности в этом примере предпочтительно  
 рассчитываются с использованием двух различных множителей вероятностей, а именно:  
 256 и 32. При использовании множителя 256 вероятностей для первого символа значение  
 масштабированной вероятности для него вычисляется следующим образом: Round(256\*7/  
 10 148)=12, где "Round" - функция округления до ближайшего большего целого. Все  
 вычисленные с помощью множителя 256 масштабированные вероятности выглядят  
 следующим образом:

12, 3, 140, 2, 61, 16, 9, 14

Сумма значений масштабированных вероятностей составляет 257, то есть превышает  
 15 256, и предпочтительно следует уменьшить некоторое значение на 1. Такое уменьшение  
 преимущественно реализуется для того, чтобы на фактическое кодирование оказывалось  
 бы наименьшее влияние. Например, в этом случае значение 2, то есть наименьшее  
 значение, может быть уменьшено до 1, или значение 9, то есть наименьшее значение,  
 округленное в большую сторону, может быть соответственно уменьшено до 8, таким  
 20 образом значения масштабированных вероятностей с множителем 256 для  
 интервального или арифметического кодирования выглядят следующим образом:

12, 3, 140, 1, 61, 16, 9, 14 (сумма = 256 = множитель вероятностей).

Доставка масштабированных вероятностей (с множителем 256) может выполняться  
 следующим образом:

25 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1

и

12, 3, 140, 1, 61, 16, 9, 14

Если значение множителя вероятностей составляет 32, то значения масштабированных  
 вероятностей выглядят следующим образом:

30 2, 0, 18, 0, 8, 2, 1, 2

и после коррекции суммы получается следующий результат:

1, 0, 18, 0, 8, 2, 1, 2

Теперь следует отметить, что для некоторых значений масштабированных  
 вероятностей вычисляется нулевое значение. Это означает, что такие значения должны  
 35 доставляться, например, с использованием управляющего символа. Для управляющего  
 символа требуется вычислить масштабированную вероятность, и она может быть  
 меньше значения 1. В этом случае выделяется значение 1, поскольку Round (32\*(2+1)/  
 148)=1. Теперь этот управляющий символ должен добавляться к другим символам, и  
 новый набор символов выглядит следующим образом: "управляющий символ", 2, 7,  
 40 12, 13, 14 и 19. Эти новые символы предпочтительно распределены индексам в  
 диапазоне от 0 до 6. Масштабированные вероятности для новых символов, если  
 вероятность управляющего символа уменьшается для одного или более других символов,  
 для интервального или арифметического кодирования выглядят следующим образом:

1 и 1, 18, 8, 2, 1, 1 (сумма = 32 = множитель вероятностей). Доставка

45 масштабированных вероятностей (с множителем 32 и управляющим кодом) может  
 выполняться следующим образом:

0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1

и

1, 1, 18, 8, 2, 1, 1

Следует принимать во внимание, что первый символ является управляющим, и биты указывают другие значения символов.

Если определен управляющий символ, эта таблица также может лучше использоваться в последствии, как указано выше. Кроме того, теперь символы, которые отсутствовали в этих данных, могут доставляться с использованием управляющего символа, если они будут входить в состав последующих данных.

Применение управляющих кодов при интервальном кодировании описывается в другой заявке на патент, поданной 20 февраля 2014 года заявителем Gurulogic Microsystems Oy под заголовком "Encoder, decoder and method" (кодер, декодер и способ) с номером заявки GB 1403038.1, включенной в настоящую заявку посредством ссылки.

Как указывалось выше при пояснении Табл. 1, опционально также можно использовать заранее заданную таблицу или таблицу, описанную индексом, а не доставленную таблицу, если данные D1 кодированы в кодере 50 и впоследствии декодируются в декодере 60. Это означает, что используемая таблица кодов, частот, вероятностей или длин кодов известна заранее кодеру 50 и декодеру 60, или таблица выбирается из ограниченного набора альтернативных таблиц, и кодер 50 доставляет выбранные данные декодеру 60. Заранее заданная таблица опционально доступна локально декодеру 60.

Таблица может предварительно сохраняться на основе доставленных параметров, например индекса таблицы и/или максимального индекса таблицы. В альтернативном варианте таблица может генерироваться с помощью алгоритма, реализованного в функции инициализации, или с помощью алгоритма в предварительно сохраненной таблице; такое создание, то есть генерация, таблицы также основано на доставляемых параметрах, например на индексе таблицы и/или максимальном индексе таблицы, и/или на минимальном индексе таблицы. Вместо предварительно сохраненной таблицы также опционально возможно использовать таблицу, доставленную ранее в процессе кодирования данных D1 для генерации кодированных данных E2.

Например, если используется кодирование VLC, обычно длина кодов сохраняется заранее для различных индексов таблиц, и кодовая таблица может генерироваться на основе этих значений длин с использованием полной таблицы или с использованием только части значений таблицы. Используемая часть может определяться на основе поставленных параметров таблицы или на основе индекса таблицы. Таким же образом, если используется интервальное кодирование, обычно таблица вероятностей генерируется на основе формы таблицы, например доставляется с помощью индекса таблицы и длины таблицы, например, доставляется в виде максимального индекса таблицы.

Далее описывается другой пример способа, соответствующего раскрытию настоящего изобретения, который позволяет выполнять эффективную доставку кода без использования отдельного управляющего символа, но при этом по-прежнему позволяет очень эффективно кодировать все символы для текущего заданного фрагмента данных и для последующих фрагментов данных, частоты символов которого немного отличаются. Этот другой способ может быть реализован таким образом, чтобы все символы, то есть символы с доступными и недоступными значениями, распределялись по меньшей мере для масштабированной вероятности со значением один. Если значение масштабированной вероятности равно 1, то бит доступности равен 0, и для других значений масштабированной вероятности бит доступности равен 1. Значение масштабированной вероятности затем требуется доставить только для символов, бит

доступности которых равен 1. Более подробная информация содержится в заявке на патент, поданной заявителем Gurulogic Microsystems Oy под номером GB 1403038.1, которая включена в данную заявку посредством ссылки, однако доставка таблицы согласно приведенному выше примеру выглядит следующим образом:

5 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0

и

10, 4 (сумма =  $18 \cdot 1 + 10 + 4 = 32$  = множитель вероятностей).

Этот пример в данном случае показывает предпочтительное решение, благодаря которому достигается производительность, почти аналогичная интервальному кодированию, в отличие от кодирования по Хаффману для кодирования энтропии данных. Такое решение позволяет с высокой эффективностью доставлять таблицы вероятностей кодов. С помощью других типов данных или других типов значений множителя вероятностей это решение, очевидно, во многих ситуациях является наилучшим, то есть оптимальным, применяемым способом кодирования. По этой причине доставка таких кодовых таблиц далее описывается более подробно.

Опционально кодер 50 и декодер 60 могут сохранять все таблицы, подлежащие использованию этими устройствами, статически или динамически обновлять их. Если таблицы сохранены, они предпочтительно идентифицируются в данных, переданных из кодера 50 в декодер 60, посредством индекса, который уникально идентифицирует свою соответствующую сохраненную таблицу. Такая индексация таблиц потенциально позволяет достичь значительной экономии служебных данных, требуемых, в противном случае, для доставки кодовой таблицы из кодера 50 в декодер 60.

Использование предварительно сохраненной таблицы можно оценить в следующем примере. Для доставляемой таблицы последняя таблица вероятностей, представленная в предшествующем примере (1, 1, 1, 1, 1, 1, 1, 10, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1), выбирается для добавления с целью последующего использования, с этой целью для таблицы выделяется значение индекса, например, "17". Теперь требуется новый подлежащий кодированию фрагмент данных, значения символов и частоты которого представлены в Табл. 2.

30 **Таблица 2: Пример вторых значений символов и частот для повторного использования таблицы**

Значение	Частота
1	3
5	1
7	68
8	1
10	4
12	32
14	3
18	1

Теперь все доступные таблицы вероятностей для интервального кодирования могут оцениваться на основе Табл. 2, и существует большая возможность того, что таблица 17 выбирается как наиболее вероятная таблица для использования с целью доставки этих новых данных из кодера 50 в декодер 60 при интервальном кодировании. По меньшей мере, легко видеть, что доставка новой таблицы вероятностей требует гораздо больше данных, чем объем дополнительных данных, создаваемых таблицей 17, для

достижения идеального результата энтропийного кодирования. По этой причине таблица 17 или некоторая другая таблица вероятностей может использоваться повторно вместо поставки новой более оптимизированной таблицы вероятностей. Если таблица используется повторно, ее индекс 17 доставляется из кодера 50 в декодер 60, и затем значения интервального кодирования могут доставляться из кодера 50 в декодер 60. Если таблица не может использоваться повторно, то из кодера 50 в декодер 60 вначале доставляется значение, например 0, или следующий свободный индекс таблицы, который определяет новую доставку таблицы, а затем таблица также доставляется из кодера 50 в декодер 60, после чего значения интервального кодирования могут доставляться из кодера 50 в декодер 60. Часто также требуется доставлять количество кодированных символов, например, обычно перед интервально кодированными данными, что позволяет корректно декодировать данные в декодере 60.

Опционально можно также модифицировать кодовую таблицу, которая уже используется, и передавать только изменения, выполненные в ней, в результате чего формируется новая кодовая таблица. Кроме того, опционально уже доставленная кодовая таблица может использоваться адаптивным образом после доставки/приема.

Опционально кодер 50 и декодер 60 способны создавать аналогичные полные таблицы, которые также позволяют выполнять кодирование других символов для последующего использования аналогичных типов данных, в которых могут содержаться отсутствующие в настоящий момент символы. Эти таблицы могут сохраняться, и им предоставляется новый индекс кодовой таблицы. Можно сохранять оба типа таблиц: одну завершенную и одну незавершенную. Кроме того, возможно сохранять только исходную таблицу, и если в следующий момент времени требуется завершенная таблица, она может быть указана отдельно при передаче данных из кодера 50 в декодер 60.

Решение, согласно которому предпочтительно сохраняются завершенные таблицы, более предпочтительно, поскольку оно упрощает принятие решений и не требует дополнительной доставки информации о том, требуется ли завершить формирование таблицы. Завершение формирования таблицы также может быть реализовано таким образом, чтобы таблица, содержащая все значения, заполнялась частотами, уменьшающимися, например, от 4 до 1 так, чтобы следующие символы распределялись относительно коротким символам, а последние символы - распределялись относительно длинным символам. В случае применения такого подхода порядок символов соответствует порядку доступных символов в последующем потоке данных.

В описанных выше примерах предпочтительно применяется доставка длин кодовых слов и доставка масштабированных вероятностей, хотя подобные технологии также могут использоваться для доставки значений частот, требуемых для кодирования по Хаффману, арифметического и интервального кодирования. Предпочтительно лучший для применения способ кодирования использует младшие биты при добавлении к кодированной информации служебной информации, которая требуется для доставки кодовой таблицы, длин кодовых слов, таблицы вероятностей или частотной таблицы, благодаря чему, например, обеспечивается передача из кодера 50 в декодер 60 меньших частей данных с использованием способов кодирования, специально оптимизированных для характера и/или типа данных содержимого, включенного в небольшие части данных, то есть во фрагменты данных. По этим причинам наилучший результат при использовании арифметического и интервального кодирования может быть достигнут, если значения частот квантуются, по меньшей мере до определенной степени, для того чтобы частотная таблица представляла почти корректные значения, которые могут доставляться посредством очевидно меньшего количества битов по сравнению с точной

таблицей, и, таким образом, в этом случае оптимальность снижается лишь в малой степени при кодировании символов, например в случае применения энтропийного кодирования. Кроме того, доставка таблиц масштабированных вероятностей позволяет очень эффективно и почти оптимально реализовать интервальное и арифметическое кодирование.

5 Если из кодера 50 в декодер 60 необходимо передать только небольшой объем данных, то обычно данные лучше передавать по существу без применения любой формы кодирования. Однако, если объем данных возрастает, то предпочтительно следует применять кодирование по Хаффману с приблизительно корректными длинами кодовых слов. По мере возрастания объема данных, подлежащих передаче из кодера 50 в декодер 10 60, постепенно большие преимущества приносит использование более точных кодовых таблиц. Кроме того, при наличии значительного объема данных, подлежащих арифметическому или интервальному кодированию, наилучшие результаты кодирования получают, если применяется более эффективное энтропийное кодирование; таким 15 образом, достигаются преимущества при кодировании битов, требуемых для доставки частоты или вероятности, относительно битов, требуемых для доставки кодовой таблицы. Доставка индекса таблицы вероятностей, частотной таблицы, таблиц длин кодовых слов или кодовой таблицы всегда выполняется аналогично, и если используется индекс, способ, использующий наилучшую доступную таблицу, позволяет достичь 20 самый высокий уровень сжатия.

Информация о выбранном для использования способе кодирования также должна доставляться из кодера 50 в декодер 60, если она не зафиксирована в зависимости от данных или их объема.

Кодер 50 и декодер 60 вместе образуют кодек 100. На практике может использоваться 25 один кодер 50 и один или более декодеров 60, например, в ситуации, когда кодер 50 генерирует кодированные данные E2, которые в широкополосном режиме передаются в множество декодеров 60, то есть осуществляется "многоадресная передача", например, через беспроводную, оптоволоконную сеть связи и т.п. Кроме того, могут возникнуть ситуации, например, в одноранговой сети передачи данных, 30 когда декодер 60 получает свои кодированные данные E2 из нескольких кодеров 50, связанных в одноранговой сети, в которой кодированные данные E2 предоставляются в отдельных кодированных фрагментах данных, объединяемых в декодере 60; такая конфигурация предоставляет преимущества, поскольку определенные части кодированных данных E2 могут передаваться в декодер 60 в большей степени локально, 35 что облегчает нагрузку по данным на каналах дальней связи, используемых для реализации таких одноранговых сетей передачи данных. Кодер 50 и декодер 60 допускают реализацию в виде специализированной цифровой аппаратуры, вычислительного устройства, которое способно выполнять одну или более программ, записанных на носителе информации, или в виде комбинации этих устройств. Кодер 50 40 и декодер 60 используются в устройстве записи и/или воспроизведения звукового сигнала, в устройстве записи и/или воспроизведения видеосигнала, в персональных компьютерах, смартфонах, цифровых камерах, видеокамерах, телевизионных приемниках, Интернет-терминалах, научном оборудовании, системах наблюдения и/или обеспечения безопасности, спутниках, выполняющих функции наземного контроля, 45 системах наблюдения за сейсмической активностью, а также в других системах.

Варианты раскрытия настоящего изобретения позволяют более эффективно доставлять таблицы, например кодовые таблицы, частотные таблицы, таблицы вероятностей или длины кодовых слов, благодаря чему это изобретение становится



более привлекательным для разделения данных на меньшие фрагменты данных, которые, например, затем могут отдельно кодироваться оптимальным образом. Кроме того, таблицы опционально кодируются с использованием способа энтропийного кодирования. Для этих меньших фрагментов данных дополнительно требуются  
5 собственные соответствующие кодовые таблицы, частотные таблицы, таблицы вероятностей или длины кодовых слов, и при наличии множества доступных различных таблиц предпочтительно для различных фрагментов данных доставлять только индексы заданных таблиц. В противном случае новую таблицу также следует доставлять в декодер 60. При доставке заданной таблицы часто полезно сохранять ее в памяти  
10 данных декодера 60 для последующего использования со своим собственным уникальным ссылочным индексом.

В примере осуществления, соответствующем раскрытию настоящего изобретения, каждый блок данных доставляется как отдельный блок данных с дополнительной информацией, описывающей количество блоков данных, принадлежащих взаимно  
15 одинаковому массиву данных; такая передача блоков данных часто бывает достаточно неэффективной, поскольку для всех блоков данных требуется доставлять идентификатор применяемого способа кодирования, количество символов, и используемые кодовую таблицу, частотную таблицу или таблицу вероятностей. Кроме того, количество блоков данных, принадлежащих взаимно одинаковому массиву данных, также необходимо  
20 доставлять из кодера 50 в декодер 60.

Опционально в кодовой таблице используется вставка одного или более добавочных символов, имеющих свое собственное соответствующее значение. Обычно в больших кодовых таблицах "управляющему" символу предпочтительно назначается собственное кодовое слово. Кроме того, предпочтительно также назначать собственное доступное  
25 слово для символа "конец коэффициентов" в кодовой таблице, которая используется для DCT-коэффициентов в формате JPEG. Это означает, что способ уже известен декодеру 60, следовательно, способ может использоваться очень эффективным образом путем добавления новых символов кодирования, которые могут при необходимости применяться, например, для обозначения "конца кодовой таблицы", "конца данных",  
30 а также для "управляющего символа". Эти дополнительные символы могут генерироваться таким образом, чтобы их частота равнялась 1 каждый раз при их использовании. Если используется доступная таблица, то добавляется соответствующий идентификатор в виде символа, который разделяет код с одним из символов с наиболее длинным кодовым словом. Если имеются оставшиеся данные, например, новый фрагмент  
35 данных после текущего фрагмента данных, кодер 50 предпочтительно использует символ "изменение таблицы кодирования", а не символ "конец данных". Если доставляется символ "изменение таблицы кодирования", после него доставляется индекс новой кодовой таблицы. Значение индекса, определяющего новую кодовую таблицу, например, применяется, если отсутствует доступная таблица, а если таблицы уже  
40 доступны, то индексу присваивается значение от 1 до общего количества таблиц. Опционально этот индекс для новой кодовой таблицы принимает значение, в котором используется столько битов, сколько требуется для представления доступных или подходящих таблиц для данных. Если доставляется значение 0 в качестве индекса для новой кодовой таблицы, то требуется доставка кодовой таблицы перед кодированием  
45 следующих символов в потоке данных, передаваемых из кодера 50 в декодер 60. В противном случае после индекса, идентифицирующего новую кодовую таблицу, сразу же могут кодироваться новые символы в пределах этой новой кодовой таблицы. Если кодируется последний фрагмент данных и последнее значение доставляется из кодера

50 в декодер 60, то кодер 50 доставляет символ "конец данных". В этом случае действителен только символ "конец данных", и символ "изменение кодовой таблицы" не используется. Если доставляется символ "конец данных", то после этого доставку каких-либо данных продолжать не требуется. Этот символ "конец данных" позволяет отменить доставку ряда значений данных для каждого фрагмента данных. Кроме того, в декодер 60 также не должна доставляться информация о способе кодирования, поскольку только используемые кодовая таблица, частотная таблица, таблица вероятностей или таблица длин кодовых слов изменяется для различных фрагментов данных. Таким образом, общий объем служебных данных, подлежащих передаче из кодера 50 в декодер 60, значительно меньше, если кодовая таблица изменяется в процессе кодирования и последующего декодирования данных. Для обнаружения, относится ли символ к "изменению кодовой таблицы" или к "завершению данных", требуется один завершающий бит, или для этого возможно генерировать оба символа с частотой 1 в кодовой таблице.

Иногда предпочтительно передавать количество значений данных, объем кодированных данных или использовать символ "конец данных" в зависимости от данных, объема данных, используемого способа кодирования и реализации декодера 60 и кодера 50. Кроме того, дополнительно предпочтительно использовать параллелизм при обработке данных в кодере 50 и/или декодере 60, а именно: после поставки объема кодированных данных декодер 60 может простым образом разделять данные для различных процессоров, процессов и потоков. Обычно часто существуют разнообразные подходы, наилучшим образом подходящие для доставки информации, в зависимости от количества значений данных, которые требуется декодировать, и в таком случае не требуется доставлять информацию о соответствующем выборе; однако, если доступно множество наилучших выборов, кодер 50 выбирает способ и доставляет в декодер 60 информацию о соответствующем решении, касающуюся наиболее подходящего выбора.

Из вышеизложенного понятно, что декодер 60 реализует по существу инверсию функций кодирования, выполняемых в кодере 50, если данные D1 и D3 должны быть по существу взаимно аналогичными, как, например, показано на фиг. 3А. Однако во многих практических ситуациях, например при многоадресной передаче кодированных данных E2 во множество взаимно различных устройств, требуется использовать транскодер 70 для перекодирования данных D3 с целью генерации соответствующих перекодированных данных D4, совместимых с заданным устройством, содержащим декодер 60 и связанный с ним транскодер 70, как показано на фиг. 3В. Опционально как декодер 60, так и транскодер 70 реализуются с помощью вычислительного оборудования; опционально транскодер 70 реализуется в специализированном оборудовании перекодирования, например в аппаратном коммутаторе и т.п. Варианты осуществления, соответствующие раскрытию настоящего изобретения, позволяют реализовать кодирование и декодирование данных без потерь или с потерями.

Опционально декодер 60 также способен выполнять перекодирование данных, например, выводить данные на дисплей, отличный от того, который требуется для представления данных (D1). В таком случае данные, обработанные кодеком 100, никогда не декодируются в свой исходный формат. Вместо этого кодированные данные E2, например, непосредственно преобразуются в некоторый другой формат, в котором они затем могут быть визуализированы, например, на экране или сохранены в файле. Примером реализации такого перекодирования является ситуация, в которой данные D1 изначально находились в формате YUV, а затем были сжаты и переданы в приемник, который поблочно восстанавливает данные, выполняет цветное преобразование и

масштабирует их в изображение RGB, подходящее для визуализации на экране, даже без реконструкции результирующего изображения YUV с максимальным разрешением.

Декодер 60 способен выполнять способ декодирования кодированных данных (E2), сгенерированных кодером 50, с целью генерации соответствующих декодированных данных (D3), при этом способ включает следующие шаги:

(i) прием кодированных данных (E2) и выделение из них одного или более наборов индексов вместе с одной или более частотными таблицами и/или одной или более кодовыми таблицами, и/или одной или более таблицами длин кодовых слов, и/или одной или более таблицами вероятностей, и/или с информацией, указывающей на одну или более из таких таблиц;

(ii) вычисление на основе одного или более наборов индексов соответствующих символов в одном или более фрагментах данных и/или сжатых символов элементов одной или более кодовых таблиц и/или одной или более частотных таблиц, и/или одной или более таблиц длин кодовых слов, и/или одной или более таблиц вероятностей; и

(iii) восстановление из символов одного или более фрагментов данных с использованием информации из одной или более кодовых таблиц и/или одной или более частотных таблиц, и/или одной или более таблиц длин кодовых слов, и/или одной или более таблиц вероятностей; и

(iv) объединение и/или преобразование одного или более фрагментов данных для генерации декодированных данных (D3).

Опционально способ включает прием по меньшей мере одной или более таблиц таким образом, чтобы по меньшей мере одна или более таблиц сохранялась для последующего повторного использования.

Опционально способ включает применение одного или более алгоритмов распаковки данных, выполняемых на шаге (iv), для генерации декодированных данных (D3). Кроме того, опционально согласно способу к одному или более алгоритмам распаковки данных относится по меньшей мере один из следующих алгоритмов: декодирование по Хаффману, декодирование VLC, энтропийное декодирование, арифметическое декодирование, интервальное декодирование.

Опционально способ включает объединение множества, состоящего из одного или более фрагментов данных, для генерации декодированных данных (D3) путем применения параллельной архитектуры процессоров для по существу параллельной обработки множества фрагментов данных.

Опционально способ включает генерацию одного или более наборов индексов на основе множества значений данных, которые объединяются друг с другом. Кроме того, опционально согласно способу индексы выводятся из одного или более RGB-пикселей, содержащих значения пикселей R, G и B или значения пикселей Y, U и V. Кроме того, опционально способ включает динамическое переключение между генерацией фрагментов данных в некодированном виде или в кодированном виде в кодированные данные (E2), в зависимости от достижимого коэффициента распаковки фрагментов данных при включении их в кодированные данные (E2).

Опционально согласно способу декодер 60 способен выделять из кодированных данных (E2) по меньшей мере один завершающий бит, указывающий на то, что символ относится к "изменению кодовой таблицы" или к "концу данных".

Опционально способ включает генерацию заданного фрагмента данных по существу только из достаточного количества индексов, требуемых для ссылки на один или более символов, представленных в заданном фрагменте данных.

Опционально способ включает распаковку одной или более кодовых таблиц,

внесенных в кодированные данные (E2). Кроме того, опционально способ включает распаковку одной или более кодовых таблиц путем декодирования по Хаффману. Кроме того, опционально согласно способу при распаковке одной или более кодовых таблиц применяется одна или более вспомогательных кодовых таблиц.

5 Опционально способ включает прием одной или более кодовых таблиц таким образом, чтобы одна или более кодовых таблиц использовалась в декодере (60) для декодирования переданных впоследствии данных.

Опционально способ включает выделение из кодированных данных (E2) одного или более идентификационных кодов, указывающих, откуда допускается доступ к одной  
10 или более кодовых таблиц через одну или более баз данных и/или одну или более баз данных на прокси-сервере.

Опционально способ включает декодирование одного или более следующих типов данных: захваченные звуковые сигналы, захваченные видеосигналы, захваченные изображения, текстовые данные, сейсмографические данные, сигналы датчиков, данные,  
15 преобразованные из аналоговой в цифровую форму (ADC), данные биомедицинского сигнала, календарные данные, экономические данные, математические данные, двоичные данные.

Опционально способ включает прием кодированных данных (E2) из множества источников данных и объединение данных, полученных из источников, для  
20 восстановления кодированных данных (E2).

Декодер 60 способен выполнять указанный выше способ декодирования кодированных данных (E2), для генерации декодированных данных (D3); декодер 60 предназначен для декодирования кодированных данных (E2), сгенерированных кодером 50, при этом декодер 60 способен:

25 (i) принимать кодированные данные (E2) и выделять из них один или более наборов индексов вместе с одной или более кодовыми таблицами и/или одной или более частотными таблицами, и/или одной или более таблицами длин кодовых слов, и/или одной или более таблицами вероятностей, и/или с информацией, указывающей на одну или более из таких таблиц;

30 (ii) вычислять на основе одного или более наборов индексов соответствующие символы в одном или более фрагментах данных и/или сжатые символы элементов в одной или более кодовых таблицах и/или одной или более частотных таблицах, и/или одной или более таблицах длин кодовых слов, и/или одной или более таблицах вероятностей; и

35 (iii) восстанавливать из символов один или более фрагментов данных с использованием информации из одной или более кодовых таблиц и/или одной или более частотных таблиц, и/или одной или более таблиц длин кодовых слов, и/или одной или более таблиц вероятностей; и

40 (iv) объединять и/или преобразовывать один или более фрагментов данных для генерации декодированных данных (D3).

Опционально декодер 60 также содержит транскодер 70 для перекодирования декодированных данных (D3) с целью генерации соответствующих перекодированных данных (D4) и/или генерации соответствующих перекодированных данных (D4) на основе кодированных данных (E2).

45 Опционально декодер 60 способен принимать по меньшей мере одну из одной или более таблиц таким образом, чтобы по меньшей мере одна из одной или более таблиц сохранялась для последующего повторного использования.

Опционально декодер 60 способен применять один или более алгоритмов распаковки

данных, выполняемых на шаге (iv), для генерации декодированных данных (D3). Кроме того, опционально в декодере 60 выполняется по меньшей мере один или более следующих алгоритмов распаковки данных: декодирование по Хаффману, декодирование VLC, энтропийное декодирование, арифметическое декодирование, интервальное декодирование.

Опционально декодер 60 способен объединять множество, состоящее из одного или более фрагментов данных, для генерации декодированных данных (D3) путем применения параллельной архитектуры процессоров для по существу параллельной обработки множества фрагментов данных.

Опционально декодер 60 способен генерировать один или более наборов индексов на основе множества значений данных, которые объединяются друг с другом. Кроме того, опционально в декодере 60 индексы выводятся из одного или более RGB-пикселей, содержащих значения пикселей R, G и B или значения пикселей Y, U и V. Кроме того, опционально декодер 60 способен осуществлять динамическое переключение между генерацией одного или более фрагментов данных в некодированном виде или в кодированном виде в кодированные данные (E2), в зависимости от достижимого коэффициента распаковки фрагментов данных при включении их в кодированные данные (E2).

Опционально декодер 60 способен выделять из кодированных данных (E2) по меньшей мере один завершающий бит, указывающий на то, что символ относится к "изменению кодовой таблицы" или к "концу данных".

Опционально декодер 60 способен генерировать заданный фрагмент данных по существу только из достаточного количества индексов, требуемых для ссылки на один или более символов, представленных в заданном фрагменте данных.

Опционально декодер 60 способен распаковывать одну или более кодовых таблиц, включенных в кодированные данные (E2). Кроме того, опционально декодер 60 способен распаковывать одну или более кодовых таблиц путем декодирования по Хаффману. Кроме того, опционально в декодере 60 при распаковке одной или более кодовых таблиц применяется одна или более вспомогательных кодовых таблиц.

Опционально декодер 60 способен принимать одну или более кодовых таблиц таким образом, чтобы одна или более кодовых таблиц использовалась в декодере (60) для декодирования переданных впоследствии данных.

Опционально декодер 60 способен выделять из кодированных данных (E2) один или более идентификационных кодов, указывающих, откуда допускается доступ к одной или более кодовым таблицам через одну или более баз данных и/или одну или более баз данных на прокси-сервере.

Опционально декодер 60 способен декодировать один или более следующих типов данных: захваченные звуковые сигналы, захваченные видеосигналы, захваченные изображения, текстовые данные, сейсмографические данные, сигналы датчиков, данные, преобразованные из аналоговой в цифровую форму (ADC), данные биомедицинского сигнала, календарные данные, экономические данные, математические данные, двоичные данные.

Опционально декодер 60 способен принимать кодированные данные (E2) из множества источников данных и объединять данные, полученные из источников, для восстановления кодированных данных (E2). Например, множество исходных данных передают в одноранговой сети передачи данных, в которой кодированные данные (E2) пересылают из кодера 50 в декодер 60.

Различные изменения могут вноситься в варианты осуществления настоящего

изобретения, описанные выше, при условии сохранения объема настоящего изобретения, определенного прилагаемой формулой изобретения. Такие термины как "включающий", "содержащий", "состоящий из", "имеет", "представляет собой", используемые в описании и формуле изобретения, не должны толковаться как исключаящие использование явно  
 5 не описанных блоков, компонентов или элементов. Ссылка на компонент в единственном числе также допускает наличие множества компонентов. Числовые значения, заключенные в скобки в прилагающейся формуле изобретения, помогают разобраться в пунктах формулы изобретения и не должны трактоваться в качестве ограничения заявленного изобретения.

10

## (57) Формула изобретения

1. Способ кодирования входных данных (D1) в кодере (50) для генерации соответствующих закодированных данных (E2), включающий:

- (a) анализ символов, представленных во входных данных (D1) и разделение и/или  
 15 преобразование входных данных (D1) на один или более фрагментов данных;  
 (b) генерацию, выполняемую как функция от появления символов по меньшей мере одного из следующего: одной или более кодовых таблиц, одной или более частотных таблиц, одной или более таблиц длин кодовых слов, одной или более таблиц вероятностей для символов, представленных в одном или более фрагментах данных;  
 20 (c) вычисление одного или более наборов индексов, связывающих символы в каждом фрагменте данных с элементами в по меньшей мере одном из следующего: одной или более кодовых таблицах, одной или более частотных таблицах, одной или более таблицах длин кодовых слов, одной или более таблицах вероятностей;  
 (d) сборку информации одного или более наборов индексов, связывающих символы  
 25 в каждом фрагменте данных, совместно с по меньшей мере одним из следующего: одной или более частотными таблицами, одной или более кодовыми таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более таких таблиц, в закодированные данные (E2);  
 30 (e) сжатие символов в упомянутых одной или более таблицах в закодированные данные (E2).

2. Способ по п. 1, отличающийся тем, что по меньшей мере одна из одной или более таблиц задана заранее.

3. Способ по п. 1 или 2, включающий доставку закодированных данных (E2),  
 35 включающих информацию одного или более набора индексов, связывающих символы в каждом фрагменте данных, совместно с по меньшей мере одним из следующего: одной или более частотными таблицами, одной или более кодовыми таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более упомянутых таблиц, а также на сжатые  
 40 символы.

4. Кодер (50), предназначенный для кодирования входных данных (D1) с целью генерации соответствующих закодированных данных (E2), содержащий:

- (a) анализатор для анализа символов, представленных во входных данных (D1), и для разделения и/или преобразования входных данных (D1) на один или более  
 45 фрагментов данных;  
 (b) генератор для генерации, выполняемой как функция от появления символов по меньшей мере одного из следующего: одной или более кодовых таблиц, одной или более частотных таблиц, одной или более таблиц длин кодовых слов, одной или более

таблиц вероятностей для символов, представленных в одном или более фрагментах данных;

5 (с) процессор для вычисления одного или более наборов индексов, связывающих символы в каждом фрагменте данных с элементами в по меньшей мере одном из следующего: одной или более кодовых таблицах, одной или более частотных таблицах, одной или более таблицах длин кодовых слов, одной или более таблицах вероятностей; и

10 (d) сборщик данных для сборки информации одного или более наборов индексов, связывающих символы в каждом фрагменте данных, совместно с по меньшей мере одним из следующего: одной или более кодовыми таблицами, одной или более частотными таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более таких таблиц, в закодированные данные (E2); и

15 (е) компрессор для сжатия символов в упомянутых одной или более таблицах в закодированные данные (E2).

5. Кодер (50) по п. 4, отличающийся тем, что он выполнен с возможностью доставки закодированных данных (E2), включающих информацию одного или более набора индексов, связывающих символы в каждом фрагменте данных, совместно с по меньшей мере одним из следующего: одной или более частотными таблицами, одной или более кодовыми таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более таких таблиц, а также на сжатые символы.

25 6. Кодер (50) по п. 5, отличающийся тем, что он выполнен с возможностью осуществлять доставку по меньшей мере одной из одной или более таблиц с помощью способа, согласно которому по меньшей мере одна из одной или более таблиц сохраняется для последующего повторного использования.

30 7. Кодер (50) по любому из пп. 4-6, отличающийся тем, что он выполнен с возможностью разделять входные данные (D1) на множество фрагментов данных и применять параллельную архитектуру процессоров для по существу параллельной обработки множества фрагментов данных.

35 8. Кодер (50) по п. 4, отличающийся тем, что он выполнен с возможностью осуществлять динамическое переключение между сборкой фрагментов данных в незакодированном виде или в закодированном виде в закодированные данные (E2), выполняемое как функция от достижимого коэффициента сжатия фрагментов данных при включении их в закодированные данные (E2).

9. Машиночитаемый носитель информации, на котором хранятся машиночитаемые инструкции, исполняемые вычислительным устройством, в состав которого входят аппаратные средства обработки для выполнения способа по любому из пп. 1-3.

40 10. Способ, выполняемый в декодере (60), для декодирования закодированных данных (E2), сгенерированных кодером (50) по любому из пп. 4-8, с целью генерации соответствующих декодированных данных (D3), при этом способ включает:

45 (i) прием закодированных данных (E2) и выделение из них информации одного или более наборов индексов совместно с по меньшей мере одним из следующего: одной или более кодовыми таблицами, одной или более частотными таблицами, одной или более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более из таких таблиц;

(ii) вычисление на основе одного или более наборов индексов соответствующих символов в по меньшей мере одном из следующего: одном или более фрагментах

данных, сжатых символах элементов в одной или более кодовых таблицах, одной или более частотных таблицах, одной или более таблицах длин кодовых слов, одной или более таблицах вероятностей;

5 (iii) восстановление из символов одного или более фрагментов данных с использованием информации из по меньшей мере одного из следующего: одной или более кодовых таблиц, одной или более частотных таблиц, одной или более таблиц длин кодовых слов, одной или более таблиц вероятностей; и

(iv) объединение и/или преобразование одного или более фрагментов данных для генерации декодированных данных (D3).

10 11. Способ по п. 10, отличающийся тем, что по меньшей мере одна из одной или более таблиц задана заранее.

12. Способ по п. 10, включающий перекодирование декодированных данных (D3) для генерации соответствующих перекодированных данных (D4) и/или генерации соответствующих перекодированных данных (D4) на основе закодированных данных  
15 (E2).

13. Способ по любому из пп. 10-12, включающий прием закодированных данных (E2) из множества источников данных и объединение данных, полученных из источников, для восстановления закодированных данных (E2).

14. Декодер (60), предназначенный для декодирования закодированных данных (E2),  
20 сгенерированных кодером (50) по любому из пп. 4-8, с целью генерации соответствующих декодированных данных (D3), при этом декодер (60) выполнен с возможностью:

(i) принимать закодированные данные (E2) и выделять из них информацию одного или более наборов индексов совместно с по меньшей мере одним из следующего: одной или более кодовыми таблицами, одной или более частотными таблицами, одной или  
25 более таблицами длин кодовых слов, одной или более таблицами вероятностей, информацией, указывающей на одну или более из упомянутых таблиц;

(ii) вычислять на основе одного или более наборов индексов соответствующие символы в по меньшей мере одном из следующего: одном или более фрагментах данных, сжатых символах элементов в одной или более кодовых таблицах, одной или более  
30 частотных таблицах, одной или более таблицах длин кодовых слов, одной или более таблицах вероятностей;

(iii) восстанавливать из символов один или более фрагментов данных с использованием информации из по меньшей мере одного из следующего: одной или более кодовых таблиц, одной или более частотных таблиц, одной или более таблиц  
35 длин кодовых слов, одной или более таблиц вероятностей; и

(iv) объединять и/или преобразовывать один или более фрагментов данных для генерации декодированных данных (D3).

15. Декодер (60) по п. 14, включающий также транскодер (70) для перекодировки декодированных данных (D3) с целью генерации соответствующих перекодированных  
40 данных (D4) и/или генерации соответствующих перекодированных данных (D4) на основе закодированных данных (E2).

16. Декодер (60) по п. 14, отличающийся тем, что он выполнен с возможностью принимать по меньшей мере одну из одной или более таблиц таким образом, чтобы по  
45 меньшей мере одна из одной или более таблиц сохранялась для последующего повторного использования.

17. Декодер (60) по любому из пп. 14-16, отличающийся тем, что он выполнен с возможностью объединять множество из одного или более фрагментов данных для генерации декодированных данных (D3) путем применения параллельной архитектуры



процессоров для по существу параллельной обработки множества фрагментов данных.

18. Декодер (60) по п. 14, отличающийся тем, что он выполнен с возможностью осуществлять динамическое переключение между генерацией одного или более фрагментов данных в незакодированном виде или в закодированном виде в закодированные данные (E2), выполняемое как функция от достижимого коэффициента распаковки фрагментов данных при включении их в закодированные данные (E2).

19. Декодер (60) по п. 14, отличающийся тем, что он выполнен с возможностью распаковывать одну или более кодовых таблиц, включенных в закодированные данные (E2).

20. Декодер (60) по п. 14, отличающийся тем, что он выполнен с возможностью принимать одну или более кодовых таблиц так, чтобы одна или более кодовых таблиц использовалась в декодере (60) для декодирования переданных впоследствии данных.

21. Декодер (60) по п. 14, отличающийся тем, что он выполнен с возможностью извлекать из закодированных данных (E2) один или более идентификационных кодов, указывающих, откуда осуществляется доступ к одной или более кодовых таблиц через одну или более баз данных и/или одну или более баз данных на прокси-сервере.

22. Декодер (60) по п. 14, отличающийся тем, что он выполнен с возможностью принимать закодированные данные (E2) из множества источников данных и объединять данные, полученные из источников, для восстановления закодированных данных (E2).

23. Кодек (100), содержащий по меньшей мере один кодер (50) по п. 4 для кодирования входных данных (D1) с целью генерации соответствующих закодированных данных (E2) и по меньшей мере один декодер (60) по п. 14 для декодирования закодированных данных (E2) с целью генерации декодированных выходных данных (D3).

24. Кодек (100) по п. 23, отличающийся тем, что по меньшей мере один кодер (50) и по меньшей мере один декодер (60) пространственно разделены друг с другом и совместно связаны друг с другом через сеть передачи данных.

25. Кодек (100) по п. 23, отличающийся тем, что сеть передачи данных сконфигурирована как одноранговая сеть связи.

26. Кодек (100) по п. 23, отличающийся тем, что кодер (50) и декодер (60) симметрично обрабатывают данные по отношению друг к другу.

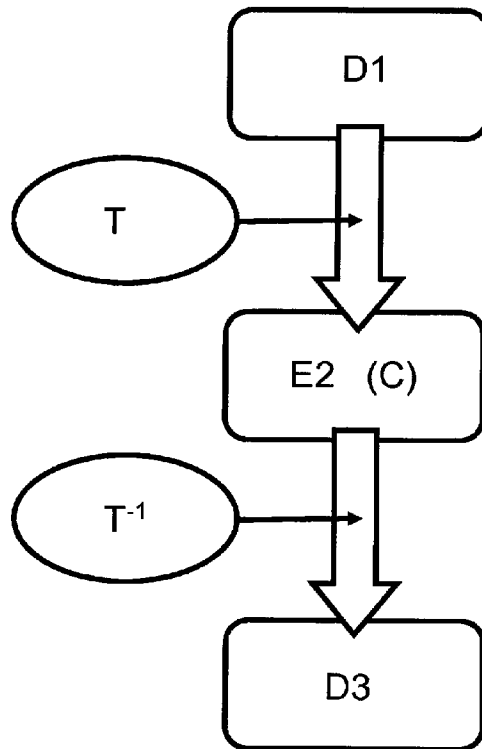
35

40

45

Способы и устройства для кодирования и декодирования  
исходных данных с использованием сжатия символов

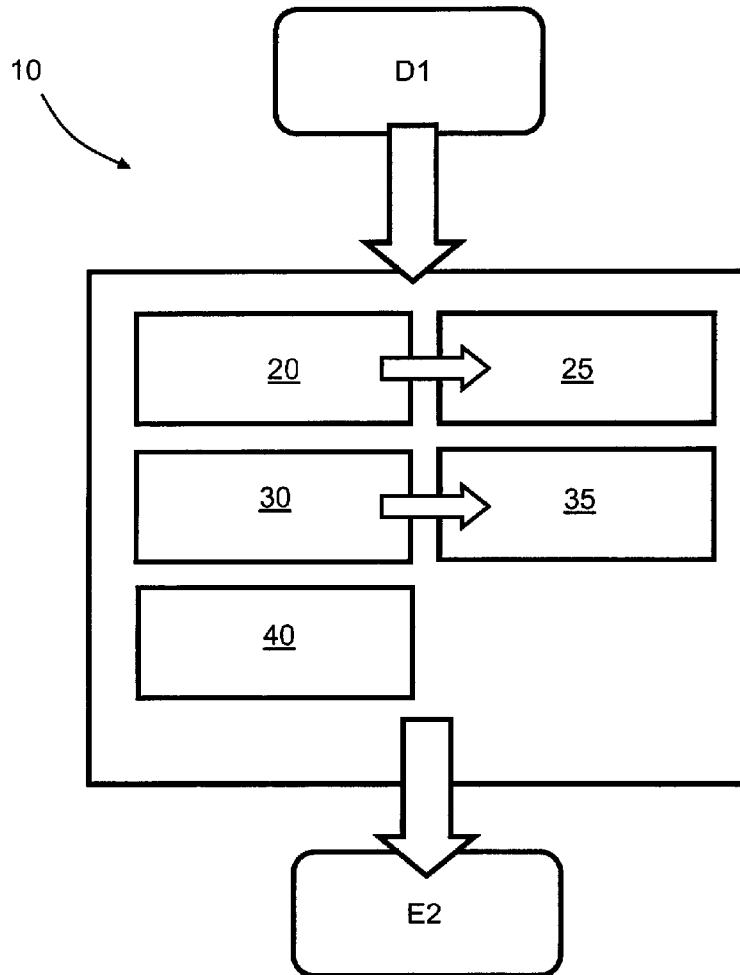
1/4



Фиг.1

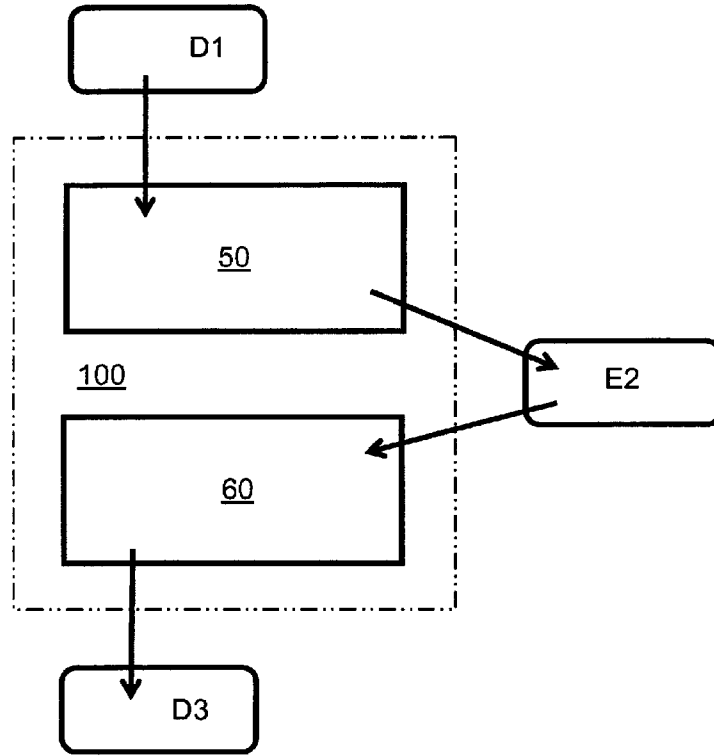
Способы и устройства для кодирования и декодирования исходных данных с использованием сжатия символов

2/4



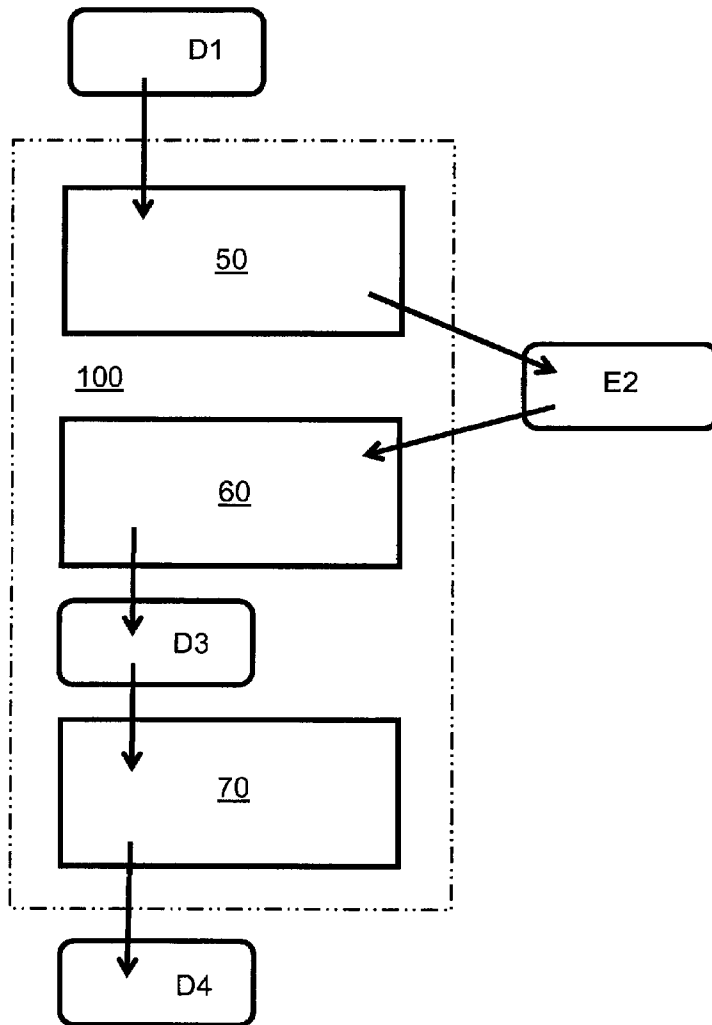
Фиг.2

3/4



Фиг.3А

4/4



Фиг.3В