

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06K 9/38 (2019.05); G06K 9/4642 (2019.05); G06T 9/00 (2019.05); H04N 19/14 (2019.05); H04N 19/90 (2019.05)

(21)(22) Заявка: 2019101654, 16.07.2014

(24) Дата начала отсчета срока действия патента:
16.07.2014Дата регистрации:
29.08.2019

Приоритет(ы):

(30) Конвенционный приоритет:
24.07.2013 IT TO2013A000629Номер и дата приоритета первоначальной заявки,
из которой данная заявка выделена:
2016105699 24.07.2013

(43) Дата публикации заявки: 11.03.2019 Бюл. № 8

(45) Опубликовано: 29.08.2019 Бюл. № 25

Адрес для переписки:

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

(72) Автор(ы):

ПАСЧАЛАКИС, Ставрос (GB)

(73) Патентообладатель(и):

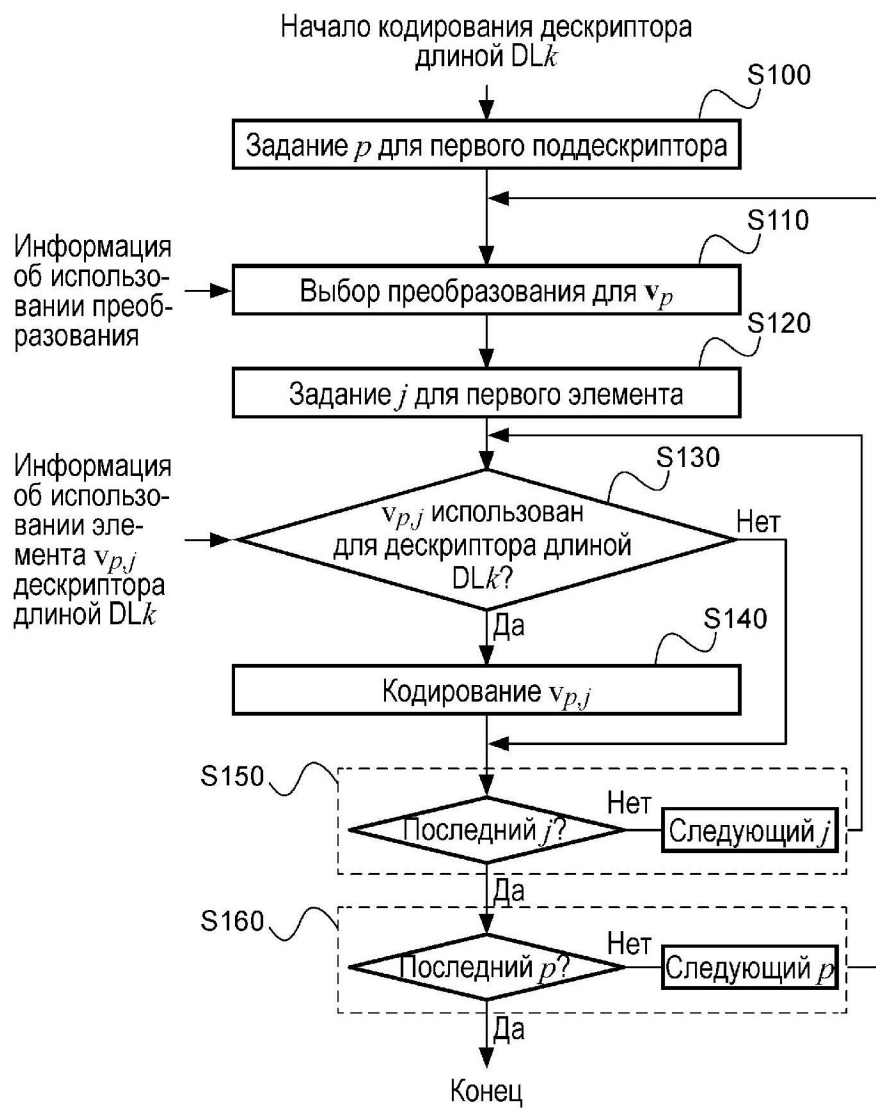
Нью Лак Глобал Лимитед (VG)

(56) Список документов, цитированных в отчете
о поиске: WO 2011044497 A2, 14.04.2011. US
8054170 B1, 08.11.2011. US 2009172730 A1,
02.07.2009. US 20130155063 A1, 20.06.2013. RU
2427911 C1, 27.08.2011.(54) УСТРОЙСТВО ОБРАБОТКИ ИЗОБРАЖЕНИЙ И СПОСОБ КОДИРОВАНИЯ ДЕСКРИПТОРА
ИЗОБРАЖЕНИЯ НА ОСНОВЕ ГИСТОГРАММ ГРАДИЕНТОВ

(57) Реферат:

Изобретение относится к области обработки изображений. Технический результат заключается в повышении производительности, снижении вычислительной сложности и количества информации, необходимой для создания масштабируемых битовых потоков. Способ кодирования дескриптора изображения на основе гистограмм градиентов в преобразованный дескриптор, содержащий преобразованные поддескрипторы, при этом каждая гистограмма градиентов содержит множество бинов гистограммы, и каждый поддескриптор содержит

набор значений. Набор значений поддескриптора получают посредством преобразования каждого из бинов гистограммы соответствующей гистограммы градиентов и выполнения скалярного квантования полученных преобразованных значений. Упомянутые поддескрипторы генерируют в соответствии со списком порядка использования элементов, определяющим индекс поддескрипторов из упомянутого набора поддескрипторов и индекс элементов из упомянутого набора значений. 2 н. и 20 з.п. ф-лы, 38 ил.



ФИГ. 5



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) ABSTRACT OF INVENTION

(52) CPC

G06K 9/38 (2019.05); *G06K 9/4642* (2019.05); *G06T 9/00* (2019.05); *H04N 19/14* (2019.05); *H04N 19/90* (2019.05)

(21)(22) Application: **2019101654, 16.07.2014**

(24) Effective date for property rights:
16.07.2014

Registration date:
29.08.2019

Priority:

(30) Convention priority:
24.07.2013 IT TO2013A000629

Number and date of priority of the initial application,
from which the given application is allocated:
2016105699 24.07.2013

(43) Application published: **11.03.2019 Bull. № 8**(45) Date of publication: **29.08.2019 Bull. № 25**

Mail address:

**129090, Moskva, ul. B. Spasskaya, 25, str. 3, OOO
"Yuridicheskaya firma Gorodisskij i Partnery"**

(72) Inventor(s):

PASCHALAKIS, Stavros (GB)

(73) Proprietor(s):

New Luck Global Limited (VG)**(54) IMAGE PROCESSING DEVICE AND A METHOD OF ENCODING AN IMAGE DESCRIPTOR BASED ON HISTOGRAMS OF GRADIENTS**

(57) Abstract:

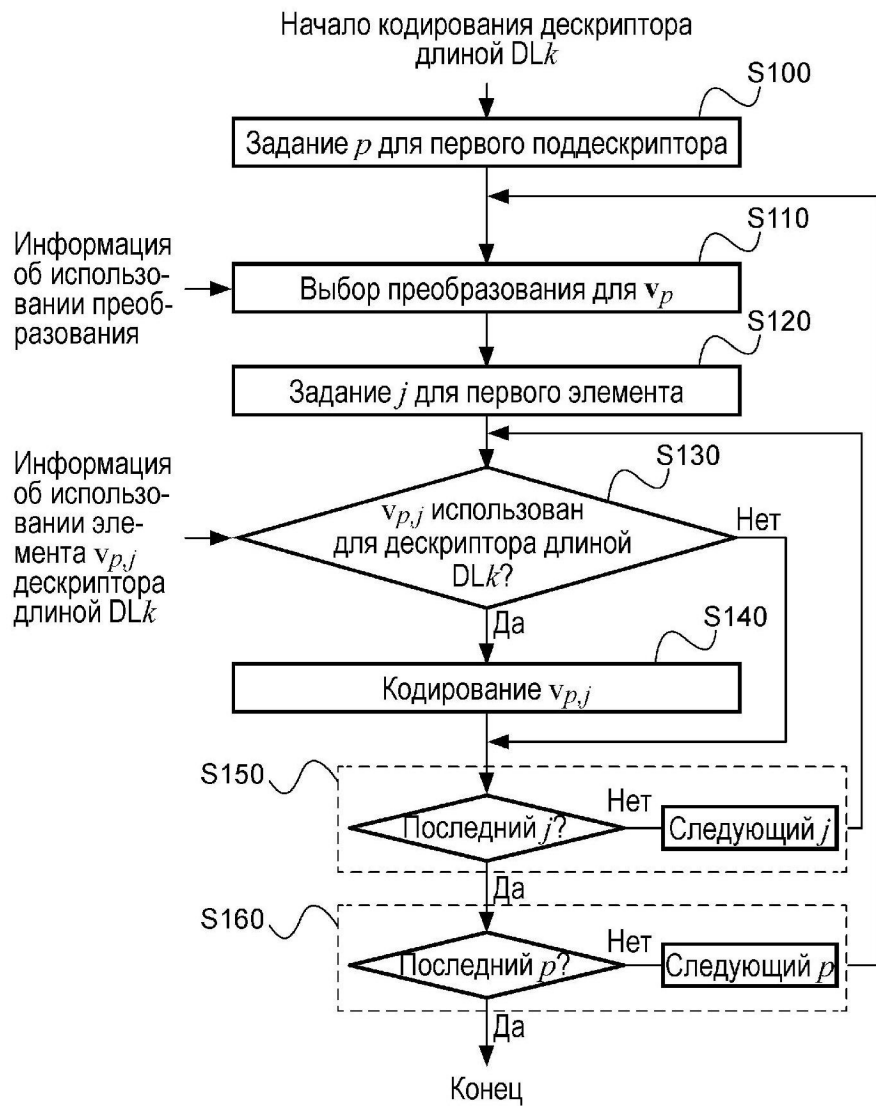
FIELD: data processing.

SUBSTANCE: invention relates to image processing. A method of encoding an image descriptor based on histograms of gradients into a converted descriptor containing transformed sub-descriptors, wherein each histogram of gradients contains a plurality of histogram bins, and each sub-descriptor has a set of values. A set of sub-descriptor values is obtained by converting each histogram bistable histogram of gradients and performing scalar quantization of the

obtained transformed values. Said sub-descriptors are generated in accordance with a list of element usage order, which determine index of sub-descriptors from said set of sub-descriptors and index of elements from said set of values.

EFFECT: technical result consists in improvement of efficiency, reduction of computational complexity and amount of information required to create scalable bit streams.

22 cl, 38 dwg



ФИГ. 5

Настоящее изобретение относится к способу эффективного кодирования, транскодирования, декодирования и обработки дескрипторов изображения, вычисляемых в локальных областях вокруг представляющих интерес ключевых точек изображения, и к устройству обработки изображений, содержащему средство для

кодирования, транскодирования, декодирования и обработки таких дескрипторов. Такие дескрипторы изображения находят широкое применение во многих прикладных областях машинного зрения, в том числе при распознавании объектов, поиске изображения по содержанию и совмещении изображений, не говоря о многом другом.

Существующие способы кодирования таких дескрипторов имеют определенные недостатки.

Например, существующие способы кодирования приводят к образованию таких дескрипторов, что требуется синтаксический анализ всех дескрипторов для выполнения транскодирования, в результате чего дескриптор данной длины преобразуется в дескриптор другой длины, или для выполнения декодирования и сравнения дескрипторов различной длины.

В качестве другого примера упомянем, что существующие способы кодирования неэффективны, если иметь в виду сложность кодирования, поскольку в них игнорируются общности и избыточности при выполнении операций, которые необходимы для образования дескрипторов изображения с переменной длиной.

В еще не опубликованной заявке № TO2012A000602 на патент Италии, поданной заявителем этой заявки, описано кодирование локальных дескрипторов изображения, в результате выполнения которого робастные (надежные), различаемые, масштабируемые и компактные дескрипторы изображения вычисляются по дескрипторам изображения при использовании гистограмм градиентов на основе преобразования указанных гистограмм градиентов, когда при указанном преобразовании собирается существенная и робастная информация, содержащаяся в них, в виде формы распределений и зависимости между значениями их бинов.

В указанной еще не опубликованной заявке на патент Италии раскрыты способы кодирования указанных дескрипторов, которые более эффективны, чем способы из предшествующего уровня техники, в части получения легко масштабируемых битовых потоков.

Такие дескрипторы раскрыты в упомянутой выше, еще не опубликованной заявке № TO2012A000602 на патент Италии, в которой раскрыто вычисление робастных, различаемых, масштабируемых и компактных дескрипторов изображения по дескрипторам изображения, в которых используются гистограммы градиентов, на основе преобразования указанных гистограмм градиентов, когда при указанном преобразовании собирается существенная и робастная информация, содержащаяся в них, в виде формы распределений и зависимости между значениями их бинов.

Ниже в этой заявке описываются значимые аспекты вычисления робастных, различаемых, масштабируемых и компактных дескрипторов изображения по дескрипторам изображения, в которых используются гистограммы градиентов, в частности SIFT дескриптора изображения согласно еще не опубликованной заявке № TO2012A000602 на патент Италии.

Коротко говоря, при использовании способа SIFT (масштабно-инвариантного преобразования признаков) локальные дескрипторы изображения формируются следующим образом: сначала выполняют поиск по всему представленному в многочисленных масштабах изображению и местам изображения, чтобы идентифицировать и локализовать стабильные ключевые точки изображения, которые

инвариантны к масштабу и ориентации; затем для каждой ключевой точки одну или несколько преобладающих ориентаций определяют на основании локальных градиентов изображения, что позволяет затем выполнить вычисление локального дескриптора относительно присвоенных ориентации, масштаба и местоположения каждой ключевой

5 точки, тем самым получают инвариантность к этим преобразованиям. После этого локальные дескрипторы изображения вокруг ключевых точек формируются следующим образом: сначала находят информацию о величине градиента и ориентации в точках выборки изображения в области вокруг ключевой точки; затем эти выборки собирают в гистограммы ориентаций с суммированием содержимого по $n \times n$ подобластям.

10 На фигурах 1а и 1б только для иллюстрации показан пример SIFT дескриптора ключевой точки, где на фигуре 1а показано подразделение локальной области R на 4×4 подобласти SR и на фигуре 1б показано подразделение 360-градусного диапазона ориентаций на восемь бинов (элементов, столбцов) для каждой гистограммы ориентаций, при этом длина каждой стрелки соответствует величине этого элемента гистограммы.

15 Таким образом, локальный дескриптор изображения, показанный на фигуре 1, имеет $4 \times 4 \times 8 = 128$ элементов. Дополнительные подробности относительно способа SIFT можно найти в David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

Согласно еще не опубликованной заявке № TO2012A000602 на патент Италии

20 робастный, различаемый, масштабируемый и компактный дескриптор изображения может быть вычислен по SIFT дескриптору следующим образом.

В последующем описании $\mathbf{H}$ представляет собой целый SIFT дескриптор, содержащий 16 гистограмм градиентов $\mathbf{h}$, где каждая имеет восемь бинов h , тогда как $\mathbf{V}$ представляет собой целый локальный дескриптор согласно настоящему изобретению, содержащий

25 16 поддескрипторов $\mathbf{v}$, где каждый имеет восемь элементов v .

Пусть $\mathbf{H}$ обозначает локальный SIFT дескриптор изображения, содержащий 16 гистограмм градиентов $\mathbf{h}_0$ - $\mathbf{h}_{15}$, показанных на фигуре 2а, где каждая гистограмма содержит восемь значений h_0 - h_7 бинов, показанных на фигуре 2б. Более робастный, различаемый, масштабируемый и компактный дескриптор изображения можно

30 вычислить путем преобразования каждого $\mathbf{h}_0$ - $\mathbf{h}_{15}$ из $\mathbf{H}$ и затем выполнения скалярного квантования полученных преобразованных значений. Более конкретно, каждый $\mathbf{h}_0$ - $\mathbf{h}_{15}$ преобразуют в соответствии с преобразованием $\mathbf{A}$ или преобразованием $\mathbf{B}$, показанными ниже, в соответствии с информацией об использовании преобразования из фигуры 3, то есть преобразование $\mathbf{A}$ применяют к $\mathbf{h}_0$, $\mathbf{h}_2$, $\mathbf{h}_5$, $\mathbf{h}_7$, $\mathbf{h}_8$, $\mathbf{h}_{10}$, $\mathbf{h}_{13}$, $\mathbf{h}_{15}$ и преобразование $\mathbf{B}$

35 применяют к $\mathbf{h}_1$, $\mathbf{h}_3$, $\mathbf{h}_4$, $\mathbf{h}_6$, $\mathbf{h}_9$, $\mathbf{h}_{11}$, $\mathbf{h}_{12}$, $\mathbf{h}_{14}$, что дает преобразованный дескриптор $\mathbf{V}$ вместе с поддескрипторами $\mathbf{v}_0$ - $\mathbf{v}_{15}$, соответствующими $\mathbf{h}_0$ - $\mathbf{h}_{15}$, соответственно, и где каждый содержит элементы v_0 - v_7 , что дает в сумме 128 элементов.

40 Преобразование A (1)

$$v_0 = h_2 - h_6$$

$$v_1 = h_3 - h_7$$

$$v_2 = h_0 - h_1$$

$$v_3 = h_2 - h_3$$

$$v_4 = h_4 - h_5$$

$$v_5 = h_6 - h_7$$

$$v_6 = (h_0 + h_4) - (h_2 + h_6)$$

$$v_7 = (h_0 + h_2 + h_4 + h_6) - (h_1 + h_3 + h_5 + h_7)$$

Преобразование В (2)

$$v_0 = h_0 - h_4$$

$$v_1 = h_1 - h_5$$

$$v_2 = h_2 - h_6$$

$$v_3 = h_3 - h_7$$

$$v_4 = h_4 - h_0$$

$$v_5 = h_5 - h_1$$

$$v_6 = (h_1 + h_5) - (h_3 + h_7)$$

$$v_7 = (h_0 + h_1 + h_2 + h_3) - (h_4 + h_5 + h_6 + h_7)$$

Затем каждый элемент подвергают грубому скалярному квантованию, например троичному (3-уровневому) квантованию, при этом пороги квантования выбирают так, чтобы получать определенное распределение вероятностей появления между бинами квантования для каждого элемента. Это скалярное квантование дает квантованный дескриптор $\mathbf{V}, \sim$ с поддескрипторами $v_0, \sim v_1, \sim v_2, \sim v_3, \sim v_4, \sim v_5, \sim v_6, \sim v_7$, и опять в сумме имеются 128 элементов. В этом компактном дескрипторе собрана наиболее различаемая и робастная информация, содержащаяся в исходных гистограммах градиентов, в виде формы распределений и зависимости между значениями их бинов.

Главное преимущество дескриптора $\mathbf{V}, \sim$ а также его квантованной версии $\mathbf{V}, \sim$, заключается в том, что он является сильно масштабируемым, и при необходимости его размерность может быть легко снижена в соответствии с требованиями к объему памяти или характеристикам передачи канала путем простого исключения одного или нескольких его элементов. Ради простоты в описании, которое следует ниже, будут описаны значимые аспекты изобретения в части кодирования предварительно квантованного дескриптора $\mathbf{V}$ с поддескрипторами $v_0 - v_{15}$, где каждый содержит элементы $v_0 - v_7$, и должно быть понятно, что если не оговорено иное, кодирование квантованного дескриптора $\mathbf{V}, \sim$ происходит аналогичным образом.

На фигурах 4а-4е показаны для примера наборы элементов, которые положены в основу для получения высокой степени различения и робастности дескрипторов пяти предварительно определенных длин, от дескриптора длиной 0 (DL0), дескриптора наименьшей длины с использованием только 20 элементов дескриптора, до дескриптора длиной 4 (DL4), дескриптора наибольшей длины с использованием всех 128 элементов. Более конкретно, на фигуре 4а показан пример набора элементов для дескриптора длиной DL0, содержащего 20 элементов, на фигуре 4б показан пример набора элементов для дескриптора длиной DL1, содержащего 40 элементов, на фигуре 4с показан пример набора элементов для дескриптора длиной DL2, содержащего 64 элемента, на фигуре 4д показан пример набора элементов для дескриптора длиной DL3, содержащего 80 элементов, и на фигуре 4е показан пример набора элементов для дескриптора длиной DL4, содержащего все 128 элементов. Таким образом, для дескриптора каждой длины каждый элемент каждого поддескриптора будет или не будет декодироваться в соответствии с наборами использования элементов (из которых удалены ненужные элементы) из фигур 4а-4е.

Объяснение этого свойства масштабируемости заключается в том, что набор элементов, используемых для дескриптора каждой длины, должен быть таким же, как поднабор, или самим поднабором из набора элементов, используемых для дескрипторов

всех больших длин, показанных на фигурах 4а-4е. Это позволяет осуществлять транскодирование и сравнение дескрипторов различной длины путем простого исключения излишних элементов дескриптора с большой длиной, чтобы выполнять приведение к такому же набору элементов, как в дескрипторе с небольшой длиной.

Способ непосредственного кодирования этого дескриптора содержит вычисление и кодирование элементов в порядке, определяемом поддескрипторами, то есть в общем случае как $v_{0,0}, v_{0,1}, \dots, v_{0,7}, v_{1,0}, v_{1,1}, \dots, v_{1,7}, \dots, v_{15,0}, v_{15,1}, \dots, v_{15,7}$, где $v_{i,j}$ обозначает элемент v_j поддескриптора v_i . Это означает кодирование элементов $v_0, v_1, \dots, v_7$ для преобразованного поддескриптора v_0 , затем кодирование элементов $v_0, v_1, \dots, v_7$ для преобразованного поддескриптора v_1 и т.д., использование соответствующих преобразований, например, показанных на фигуре 3, и также использование соответствующих наборов использования элементов для дескриптора предварительно определенной длины, например, показанных на фигуре 4, для принятия решения, какие элементы следует кодировать.

Это кодирование приводит, например, для дескриптора длиной DL_0 к дескриптору $v_{0,0}, v_{1,0}, v_{2,0}, v_{3,0}, v_{4,0}, v_{5,0}, v_{5,6}, v_{6,0}, v_{6,6}, v_{7,0}, v_{8,0}, v_{9,0}, v_{9,6}, v_{10,0}, v_{10,6}, v_{11,0}, v_{12,0}, v_{13,0}, v_{14,0}, v_{15,0}$ и для дескриптора длиной DL_1 к дескриптору $v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1}, v_{2,0}, v_{2,1}, v_{3,0}, v_{3,1}, v_{4,0}, v_{4,1}, v_{5,0}, v_{5,1}, v_{5,2}, v_{5,6}, v_{6,0}, v_{6,1}, v_{6,2}, v_{6,6}, v_{7,0}, v_{7,1}, v_{8,0}, v_{8,1}, v_{9,0}, v_{9,1}, v_{9,2}, v_{9,6}, v_{10,0}, v_{10,1}, v_{10,2}, v_{10,6}, v_{11,0}, v_{11,1}, v_{12,0}, v_{12,1}, v_{13,0}, v_{13,1}, v_{14,0}, v_{14,1}, v_{15,0}, v_{15,1}$.

На фигуре 5 работа такого непосредственного кодера показана в виде последовательности этапов. При последующем описании, а также при последующих описаниях работы кодера, если не оговорено иное, такая последовательность этапов будет соответствовать этапам, которые являются концептуальными, и не будет соответствовать конкретным аппаратным или программным реализациям, компонентам и командам, но будет представлять в целом работу кодера. Более конкретно, на фигуре 5 показана работа кодера применительно к дескриптору длиной DL_k , например, соответствующего дескриптору одной из длин, показанных на фигуре 4. На этапе S100 из фигуры 5 кодирование дескриптора начинается с первого поддескриптора, то есть v_0 . На этапе S110 соответствующее преобразование выбирается для обрабатываемого поддескриптора, например, в соответствии с использованием преобразования из фигуры 3. Следует отметить, что вычисление дескриптора **V** по дескриптору **H** в соответствии с двумя различными преобразованиями, описанными в этой заявке, является только примером. Кроме того, вычисление дескриптора **V** по дескриптору **H** можно выполнять в соответствии с одним преобразованием, например только преобразованием А или только преобразованием В, при этом выполнение этапа S110 не является необходимым, или в соответствии с более чем двумя преобразованиями. На этапе S120 кодирование обрабатываемого поддескриптора начинается с первого элемента поддескриптора, то есть v_0 . Затем на этапе S130 проверяется, использовался или нет конкретный элемент конкретного поддескриптора, то есть $v_{0,0}$, по информации об использовании элементов для дескриптора длиной DL_k , например, при использовании одного из наборов использования из фигуры 4. Если элемент не использовался, то при обработке осуществляется переход к этапу S150. Если элемент использовался для дескриптора длиной DL_k , то кодирование его выполняется на этапе S140. В этом месте, а также при последующих описаниях операции кодирования, если не оговорено иное, слово «кодирование» означает одно или несколько действий или сочетаний их, что делает элемент $v_{0,0}$ частью локального дескриптора изображения, при этом указанные действия

включают в себя, в качестве примера и без ограничения, вычисление в соответствии с подходящей функцией (1) или (2) преобразования, показанного ранее, выбор элемента для включения в локальный дескриптор изображения в случае, когда для всех элементов выполнены предварительные вычисления без знания, какие элементы будут в конечном

5 счете использоваться в дескрипторе, квантование значения элемента, сохранение элемента в энергонезависимом или энергонезависимом запоминающем устройстве и передачу элемента по каналу передачи. После этапа S140 или, если на этапе S130 было решено, что элемент не использовался для дескриптора длиной DL_k , при обработке осуществляется переход к этапу S150. На этапе S150, если текущий элемент не является

10 последним элементом поддескриптора, при обработке осуществляется переход к следующему элементу поддескриптора, в противном случае при обработке осуществляется переход к этапу S160. На этапе S160, если текущий поддескриптор не является последним поддескриптором локального дескриптора изображения, при обработке осуществляется переход к следующему поддескриптору локального

15 дескриптора изображения, в противном случае обработка заканчивается. Таким образом, ясно, что этапы S100, S120, S150 и S160 относятся к порядку, в котором обработка выполняется, тогда как этапы S110, S130 и S140 относятся к фактическому кодированию локального дескриптора изображения.

Другой способ непосредственного кодирования этого дескриптора содержит

20 вычисление и кодирование элементов в порядке, определяемом элементами, то есть в общем случае как $v_{0,0}, v_{1,0}, \dots, v_{15,0}, v_{0,1}, v_{1,1}, \dots, v_{15,1}, \dots, v_{0,7}, v_{1,7}, \dots, v_{15,7}$, то есть кодирование элемента v_0 для поддескрипторов $v_0, v_1, \dots, v_{15}$, затем кодирование элемента v_1 для поддескрипторов $v_0, v_1, \dots, v_{15}$, и т.д., и опять с использованием надлежащих

25 преобразований, например, показанных на фигуре 3, и также с использованием соответствующих наборов использования элементов для дескриптора предварительно определенной длины, например, показанных на фигуре 4, для принятия решения, какие элементы следует кодировать. Такой кодер может работать аналогично кодеру из

30 фигуры 5 при соответствующем изменении порядка этапов. В общем случае ни один из двух упомянутых выше способов не обеспечивает преимущества перед другим способом. Для задач транскодирования, декодирования и обработки декодер также должен снабжаться информацией о процессе кодирования, и упорядочении элементов, и наборах использования, чтобы он мог обрабатывать и сравнивать дескрипторы, возможно различной длины, для задач, имеющих отношение к прикладным областям

35 машинного зрения. Таким образом, наборы использования элементов совместно с дескрипторами необходимо постоянно фиксировать или сохранять/передавать. В связи с этим процесс непосредственного кодирования является невыгодным.

Более конкретно, при таком кодировании игнорируется относительная значимость различных элементов в порядке кодирования. Следовательно, при транскодировании,

40 в соответствии с которым дескриптор предварительно определенной длины преобразуется в дескриптор другой длины, или при декодировании и сравнении дескрипторов различной длины путем сравнения соответствующих элементов двух дескрипторов, такое кодирование неизбежно влечет за собой синтаксический анализ дескрипторов для достижения предварительно определенного результата.

Кроме того, при таком кодировании игнорируются относительно значимые картины

45 избыточности различных элементов и существует чрезмерная сложность, касающаяся принятия решения относительно того, следует или не следует кодировать отдельные элементы.

Поэтому задача настоящего изобретения заключается в нахождении способа кодирования дескриптора изображения на основе гистограммы градиентов, который является более эффективным по сравнению со способами из предшествующего уровня техники, и связанного со способом устройства для обработки изображений.

5 Дальнейшая задача настоящего изобретения заключается в нахождении способа кодирования дескриптора изображения на основе гистограммы градиентов, который является более гибким, и связанного с ним устройства для обработки изображений.

10 Дальнейшая задача настоящего изобретения заключается в нахождении способа кодирования дескриптора изображения на основе гистограммы градиентов, который позволяет осуществлять оптимизированное кодирование, и связанного с ним устройства для обработки изображений.

15 Дальнейшая задача настоящего изобретения заключается в нахождении способа кодирования дескриптора изображения на основе гистограммы градиентов, который позволяет получать дескрипторы изображения любой длины, и связанного с ним устройства для обработки изображений.

Эти и другие задачи изобретения решаются способом кодирования дескриптора изображения на основе гистограммы градиентов, и связанным с ним устройством для обработки изображений, заявленными в прилагаемой формуле изобретения, которая является неотъемлемой частью настоящего описания.

20 Коротко говоря, найден способ эффективного кодирования дескрипторов изображения, таких как дескрипторы, описанные выше, путем кодирования их в соответствии с порядком использования элементов, в результате которого образуются масштабируемые дескрипторы, которые могут быть преобразованы в дескрипторы меньшей длины путем простого усечения дескриптора вместо выполнения
25 синтаксического анализа.

Кодирование выполняют в соответствии с группами поддескрипторов, формируемыми в соответствии с картинками избыточности с учетом относительной значимости соответствующих элементов указанных групп поддескрипторов.

30 Более конкретно, группирование выполняют путем группирования поддескрипторов, соответствующие элементы которых имеют аналогичную значимость в упорядочении всех элементов дескриптора, в соответствии с их относительной значимостью в части достижения хорошей характеристики распознавания, более конкретно, путем группирования поддескрипторов в соответствии с их расстоянием от центра дескриптора и кроме того, путем группирования поддескрипторов в соответствии с расстояниями
35 между ними, и/или кроме того, упорядочения поддескрипторов из группы согласно соответствующим характеристикам кодирования, и/или кроме того, путем упорядочения поддескрипторов из группы в соответствии с расстояниями между ними.

Способ кодирования согласно изобретению представляет интерес как более эффективный по сравнению со способами из предшествующего уровня техники в части
40 производительности, вычислительной сложности и/или количества информации, необходимой для генерирования масштабируемых битовых потоков.

Дальнейшие признаки изобретения изложены в прилагаемой формуле изобретения, которая предполагается неотъемлемой частью настоящего описания.

45 Изложенные выше задачи станут более ясными из нижеследующего подробного описания способа кодирования дескриптора изображения на основе гистограммы градиентов, и связанного с ним устройства для обработки изображений с конкретной привязкой к прилагаемым чертежам, на которых:

фигуры 1a и 1b - пример дескриптора ключевой точки из предшествующего уровня

техники;

фигуры 2a и 2b - гистограммы градиентов дескриптора ключевой точки из фигуры 1 и значения бинов, имеющих отношение к одной из указанных гистограмм градиентов, соответственно;

5 фигура 3 - примеры преобразований, применяемых к гистограммам градиентов из фигуры 2;

фигуры 4a-4e - примеры наборов элементов для пяти дескрипторов соответствующих предварительно определенных длин;

10 фигура 5 - блок-схема последовательности действий, иллюстрирующая работу кодера при использовании наборов элементов из фигуры 4;

фигура 6 - иллюстрация порядка использования элементов, применяемого в способе согласно настоящему изобретению;

фигура 7 - иллюстрация работы кодера при использования порядка использования элементов из фигуры 6;

15 фигура 8 - иллюстрация центров области и подобласти локального дескриптора изображения;

фигура 9 - первая группировка поддескрипторов локального дескриптора изображения в соответствии с первым вариантом осуществления или четвертым вариантом осуществления способа согласно изобретению;

20 фигуры 12 и 14 - примеры второй и третьей группировок, соответственно, поддескрипторов локального дескриптора изображения в соответствии с первым вариантом осуществления способа согласно изобретению;

фигуры 10, 13 и 15 - иллюстрации первого, второго и третьего порядков использования групповых элементов, относящихся к группировкам из фигур 9, 12, 14

25 и 17, соответственно;

фигура 11 - иллюстрация работы кодера в соответствии с первым, вторым или четвертым вариантами осуществления способа согласно изобретению;

фигура 16 - пример четвертой группировки в соответствии с первым вариантом осуществления способа согласно изобретению;

30 фигура 17 - пример пятой группировки поддескрипторов локального дескриптора изображения в соответствии со вторым и третьим вариантами осуществления способа согласно изобретению;

35 фигура 18 - иллюстрация работы кодера при использовании порядка использования элементов из фигуры 17, в соответствии с третьим вариантом осуществления способа согласно изобретению;

фигура 19 - иллюстрация работы кодера в соответствии с четвертым вариантом осуществления способа согласно изобретению;

40 фигура 20 - пример группировки поддескрипторов локального дескриптора изображения в соответствии с пятым вариантом осуществления способа согласно изобретению;

фигуры 21a-21e - примеры наборов элементов в соответствии с группировкой из фигуры 20;

фигуры 22a-22e - наборы элементов из фигур 21a-21e, соответственно, преобразованные в наборы использования групповых элементов;

45 фигура 23 - иллюстрация работы кодера при кодировании дескриптора для получения преобразованных наборов элементов из фигур 22a-22e; и

фигура 24 - структурная схема устройства для обработки изображений, пригодного для осуществления способа согласно настоящему изобретению.

Согласно изобретению вместо кодирования в соответствии с наборами использования элементов для дескрипторов различной длины, более эффективный кодер может работать в соответствии с порядком использования элементов, чтобы создавать дескриптор, элементы которого упорядочены в соответствии с порядком использования элементов, и который может быть преобразован в дескриптор меньшей длины простым усечением дескриптора. Такой порядок использования элементов может иметь вид 128-элементного упорядоченного списка, который можно кодировать 112 байтами, при этом, как показано на фигуре 6, каждой позицией списка определяются индекс поддескриптора и индекс элемента. Так например, на фигуре 6 показан список приоритетов элементов, в соответствии с которым элементу $v_{5,0}$ придан наивысший приоритет, элементу $v_{9,0}$ придан второй наивысший приоритет и т.д. Поэтому при таком порядке использования элементов кодер может создавать дескриптор длиной I путем кодирования верхних элементов I в списке.

На фигуре 7 показана работа такого кодера с использованием порядка использования элементов из фигуры 6. Что касается кодера из фигуры 7, то кодирование дескриптора начинается с элемента с верхним приоритетом (приоритетом 1) в порядке использования элементов, то есть с элемента v_0 поддескриптора v_5 . На этапе S200 подходящее преобразование выбирается в соответствии с поддескриптором, к которому этот элемент принадлежит, например, в соответствии с использованием преобразования из фигуры 3. Следует отметить, что вычисление дескриптора V по дескриптору H в соответствии с двумя различными преобразованиями, описанными в этой заявке, является только примером. Кроме того, в других вариантах осуществления вычисление дескриптора V по дескриптору H может выполняться в соответствии с одним преобразованием, например только преобразованием A или только преобразованием B , при этом выполнение этапа S200 не является необходимым, или в соответствии с более чем двумя преобразованиями. Затем на этапе S210 выполняется кодирование элемента, то есть $v_{5,0}$. После этого на этапе S220, если еще не выполнено кодирование предварительно определенного количества элементов I дескриптора, при обработке осуществляется переход к элементу со следующим наивысшим приоритетом в порядке использования элементов, в противном случае обработка заканчивается. Таким образом, этап S220 относится к контролю количества кодированных элементов, тогда как этапы S200 и S210 относятся к фактическому кодированию локального дескриптора изображения.

Таким образом, кодером из фигуры 7 при использовании такого же порядка использования элементов, как на фигуре 6, образуются дескрипторы, элементы которых упорядочены в соответствии с порядком использования элементов, и которые могут быть преобразованы в дескрипторы меньшей длины путем простого усечения дескрипторов, то есть исключения последних элементов дескриптора, и при этом кодер является более гибким, чем кодер из фигуры 5, в котором используются такие же наборы использования элементов, как на фигуре 4.

Некоторое количество элементов I в дескрипторе можно сохранять/передавать совместно с дескриптором, возможно, на уровне изображения. Кроме того, для задач транскодирования, декодирования и обработки декодер также должен снабжаться информацией об упорядочении элементов, чтобы он мог выполнять обработку для задач, имеющих отношение к прикладным областям машинного зрения. Поэтому порядок использования элементов необходимо постоянно фиксировать или сохранять/передавать совместно с дескрипторами.

Однако на практике эффективность кодера из фигуры 7 можно повысить. Дело в

том, что порядок использования элементов, показанный на фигуре 6, содержит значительную степень избыточности и в известной мере его непрактично генерировать и использовать.

Причина этого заключается в том, что полный набор из 128 элементов **V** соответствует не одному дескриптору, элементы которого могут быть упорядочены в соответствии с одним списком приоритетов, а 16 различным 8-элементным поддескрипторам, где каждый поддескриптор извлечен из другой гистограммы градиентов в соответствии с конкретным преобразованием и, таким образом, в соответствующих элементах всех поддескрипторов захватывается зависимость между бинами с одинаковым угловым разделением.

При этом было обнаружено, что для получения хорошей характеристики распознавания при ограниченном наборе элементов, для дескриптора необходимо находить баланс между равномерным распределением элементов, то есть выбором элемента (элементов) из как можно большего числа поддескрипторов, и расстоянием от центра дескриптора, то есть приданием более высокого приоритета поддескрипторам, которые находятся ближе к центру дескриптора. Вместе с этим также было обнаружено, что значимость соответствующих элементов из различных поддескрипторов приблизительно одинаковая, когда расстояния поддескрипторов до центра дескриптора являются одними и теми же, тогда как значимость соответствующих элементов из различных поддескрипторов повышается по мере уменьшения расстояний поддескрипторов до центра дескриптора.

Как показано на фигуре 8, в данном случае расстояние от поддескриптора v до центра дескриптора отражает расстояние между центром подобласти, соответствующей гистограмме градиентов h , которая обуславливает поддескриптор v , и центром области, которая содержит подобласти. Хотя можно обращаться к размерам области и подобластей изображения для вычисления указанных расстояний, в этом нет необходимости, поскольку интерес представляет только сравнение указанных расстояний. Ввиду этого указанные расстояния можно вычислять в предположении, что каждая сторона каждой подобласти имеет, например, единичную длину. Кроме того, в этом описании указанные расстояния представляют собой евклидовы расстояния, хотя также могут быть использованы другие подходящие меры расстояния.

Следовательно, показано, что для задач эффективного кодирования дескриптора изображения его поддескрипторы можно группировать с тем, чтобы соответствующим элементам дескриптора для поддескрипторов каждой группы присваивать общую значимость в порядке использования и кодировать совместно.

Первый вариант осуществления

В первом варианте осуществления изобретения поддескрипторы группируют в соответствии с их расстоянием от центра дескриптора.

Например, одна такая группировка показана на фигуре 9, в которой имеются три группы, а именно $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$, содержащая поддескрипторы с наибольшими расстояниями до центра дескриптора, $g_1 = \{v_1, v_2, v_4, v_7, v_8, v_{11}, v_{13}, v_{14}\}$, содержащая поддескрипторы со вторыми наибольшими расстояниями до центра дескриптора, и $g_2 = \{v_5, v_6, v_9, v_{10}\}$, содержащая поддескрипторы с наименьшими расстояниями до центра дескриптора. В каждой группе поддескрипторы упорядочены в возрастающем порядке индекса поддескриптора, хотя это не является ограничивающим и можно использовать другие порядки, такие как по направлению движения часовой стрелки, начиная с самого верхнего, самого левого поддескриптора в группе. В каждой группе всем

соответствующим элементам поддескрипторов группы присваивают один и тот же приоритет кодирования.

На основании этой группировки можно сгенерировать порядок использования групповых элементов, который может иметь вид 24-элементного упорядоченного списка, который можно кодировать 15 байтами, при этом каждой позицией списка определяется группа поддескриптора и индекс элемента, показанные на фигуре 10. Следует отметить, что порядок использования групповых элементов из фигуры 10 является только примером, и другие порядки использования групповых элементов можно генерировать путем изменения приоритетов позиций списка. При этом порядок использования групповых элементов из фигуры 10 намного более экономичный, чем порядок использования элементов из фигуры 6, в части протяженности и объема кодирования. Так например, на фигуре 10 показан список приоритета групповых элементов, в котором элементу v_0 из группы g_2 придан наивысший приоритет, указывающий кодеру, что первыми четырьмя элементами для кодирования являются $v_{5,0}$, $v_{6,0}$, $v_{9,0}$ и $v_{10,0}$, элементу v_0 из группы g_1 придан второй наивысший приоритет, указывающий кодеру, что следующими восемью элементами для кодирования являются $v_{1,0}$, $v_{2,0}$, $v_{4,0}$, $v_{7,0}$, $v_{8,0}$, $v_{11,0}$, $v_{13,0}$ и $v_{14,0}$ и т.д.

На фигуре 11 показана работа такого кодера с использованием порядка использования групповых элементов из фигуры 10 и выполненного с возможностью кодирования верхних m групп элементов в указанном порядке использования групповых элементов. В кодере из фигуры 11 кодирование дескриптора начинается с группы элементов с высшим приоритетом (приоритетом 1) в порядке использования групповых элементов, то есть с элемента v_0 из группы g_2 , содержащей поддескрипторы v_5 , v_6 , v_9 и v_{10} . На этапе S300 кодирование дескриптора начинается с первого поддескриптора из группы, то есть v_5 . На этапе S310 надлежащее преобразование выбирается для поддескриптора, например, в соответствии с использованием преобразования из фигуры 3. Следует отметить, что вычисление дескриптора V по дескриптору H в соответствии с двумя различными преобразованиями, описанными в этой заявке, является только примером. В других вариантах осуществления вычисление дескриптора V по дескриптору H также может выполняться в соответствии с одним преобразованием, например только преобразованием А или только преобразованием В, при этом выполнение этапа S310 не является необходимым, или в соответствии с более чем двумя преобразованиями. Затем на этапе S320 выполняется кодирование элемента, то есть $v_{5,0}$. На этапе S330, если текущий поддескриптор не является последним поддескриптором в группе, при обработке осуществляется переход к следующему поддескриптору, в противном случае при обработке осуществляется переход к этапу S340. После этого на этапе S340, если предварительно определенное количество m групп элементов еще не кодировано, при обработке осуществляется переход к группе элементов со следующим наивысшим приоритетом в порядке использования групповых элементов, в противном случае обработка заканчивается. Таким образом, этапы S300, S330 и S340 относятся к порядку, в котором выполняется обработка, и к контролю количества кодированных групп элементов, тогда как этапы S310 и S320 относятся к фактическому кодированию локального дескриптора изображения.

Некоторое количество групп m или соответствующее некоторое количество элементов /в дескрипторе можно сохранять/передавать совместно с дескриптором, возможно, на уровне изображения.

То, что порядок использования групповых элементов из фигуры 10 является более

экономичным, чем порядок использования элементов из фигуры 6, приводит к более эффективному и экономичному кодеру. Кроме того, как было показано ранее, для задач транскодирования, декодирования и обработки декодер также должен снабжаться информацией о процессе кодирования и порядке использования элементов, чтобы он мог выполнять обработку и сравнение дескрипторов для задач, имеющих отношение к прикладным областям машинного зрения, и это означает, что порядок использования элементов необходимо фиксировать или передавать совместно с дескрипторами.

Различные прикладные программы могут требоваться для изменения порядка использования элементов, возможно, на уровне изображения или фрагмента изображения, например, путем придания наивысшего приоритета поддескрипторам, ближайшим к центру дескриптора, или путем придания наивысшего приоритета отдельному классу элементов, например v_7 по сравнению с v_2 , чтобы получать хорошую характеристику распознавания при ограниченном наборе элементов. В этом случае порядок использования элементов должен сохраняться или передаваться совместно с дескрипторами. С учетом того, что низкоскоростные дескрипторы обычно имеют размер в несколько сотен байтов, порядок использования групповых элементов из фигуры 10 означает намного меньшие непроизводительные затраты по сравнению с порядком использования элементов из фигуры 6. Кроме того, группировку поддескрипторов можно фиксировать и представлять к кодеру и декодеру или можно передавать совместно с дескрипторами. Например, для группировки, рассматривавшейся до сих пор, некоторое количество групп и размер компоновки каждой группы можно кодировать меньше чем в 10 байтах.

В качестве другого примера согласно первому варианту осуществления изобретения на фигуре 12 показана другая группировка, в которой имеются две группы, а именно $g_0 = \{v_0, v_1, v_2, v_3, v_4, v_7, v_8, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}\}$, содержащая все периферийные поддескрипторы, и $g_1 = \{v_5, v_6, v_9, v_{10}\}$, содержащая поддескрипторы с наименьшими расстояниями до центра дескриптора, то есть все центральные поддескрипторы. Таким образом, в этом примере группа g_1 содержит поддескрипторы с переменными расстояниями до центра дескриптора, но всегда находящиеся дальше от центра, чем поддескрипторы группы g_0 . В каждой группе всем соответствующим элементам поддескрипторов из группы присваивают одинаковый приоритет кодирования. На основании этой группировки можно сгенерировать порядок использования групповых элементов, который может иметь вид 16-элементного упорядоченного списка, который можно кодировать 8 байтами, при этом, как показано на фигуре 13, каждой позицией списка определяются группа поддескриптора и индекс элемента. В таком случае кодер из фигуры 11 можно опять использовать для кодирования дескриптора в соответствии с порядком использования групповых элементов из фигуры 13. Следует отметить, что порядок использования групповых элементов из фигуры 13 является только примером, и другие порядки использования групповых элементов можно генерировать путем изменения приоритетов позиций в списке.

В качестве еще одного примера согласно первому варианту осуществления на фигуре 14 показана другая группировка, в которой имеются четыре группы, а именно $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$, содержащая поддескрипторы с наибольшими расстояниями до центра дескриптора, $g_1 = \{v_1, v_2, v_4, v_7\}$, содержащая набор из четырех поддескрипторов со вторыми наибольшими расстояниями до центра дескриптора, $g_2 = \{v_8, v_{11}, v_{13}, v_{14}\}$, содержащая набор из четырех различных поддескрипторов, и опять со вторыми

наибольшими расстояниями до центра дескриптора, и $g_3 = \{v_5, v_6, v_9, v_{10}\}$, содержащая поддескрипторы с наименьшими расстояниями до центра дескриптора. Таким образом, в этом примере поддескрипторы групп g_1 и g_2 находятся на одинаковом расстоянии от центра дескриптора. Эта группировка получена на основании группировки из фигуры 9 путем подразделения исходной группы g_1 на новые группы g_1 и g_2 . Преимущество заключается в том, что оно приводит к группам с тем же самым количеством поддескрипторов, что желательно при реализациях оптимизированного кодера. В каждой группе всем соответствующим элементам поддескрипторов присвоен один и тот же приоритет кодирования. На основании этой группировки можно генерировать порядок использования групповых элементов, который может иметь вид 32-элементного упорядоченного списка, который может быть закодирован 20 байтами, при этом, как показано на фигуре 15, каждой позицией списка определяются группа поддескриптора и индекс элемента. В таком случае кодер из фигуры 11 можно опять использовать для кодирования дескриптора в соответствии с порядком использования групповых элементов из фигуры 15. Следует отметить, что порядок использования групповых элементов из фигуры 15 является только примером, и другие порядки использования групповых элементов можно генерировать путем изменения приоритетов позиций в списке.

Ясно, что группировка из фигуры 14 в виде четырех групп не является однозначной. Например, на фигуре 16 показана альтернативная группировка, содержащая группы $g_0 = \{v_0, v_3, v_{12}, v_{15}\}$, $g_1 = \{v_1, v_2, v_{13}, v_{14}\}$, $g_2 = \{v_4, v_7, v_8, v_{11}\}$ и $g_3 = \{v_5, v_6, v_9, v_{10}\}$. Таким образом, на фигуре 16 группы g_0 и g_3 идентичны таким же группам из фигуры 14, но поддескрипторы g_1 и g_2 меняются местами по сравнению с фигурой 14, так что каждая из этих двух групп содержит один поддескриптор из верхней левой, верхней правой, нижней левой и нижней правой части сетки поддескрипторов.

Одно различие между кодером из фигуры 7 и кодером из фигуры 11 заключается в том, что в отличие от первого, который образует дескрипторы любой длины, последний позволяет кодировать дескрипторы, длина которых имеет степень детализации, определяемую группами поддескрипторов. На практике последний можно конфигурировать для образования дескрипторов любой длины, что будет показано ниже.

Второй вариант осуществления

Во втором варианте осуществления изобретения поддескрипторы группируют, во-первых, в соответствии с их расстоянием от центра (первое условие) и, во-вторых, в соответствии с их взаимными расстояниями (второе условие).

В данном случае расстояние между поддескрипторами опять может иметь форму евклидова расстояния или других подходящих расстояний, таких как расстояние Манхэттена.

Вторым условием может быть, например, то, что группа не должна содержать поддескрипторов, расстояния которых друг от друга ниже предварительно определенного порога. Поэтому предварительно определенный порог можно устанавливать, чтобы предотвращать, например, группирование соседних поддескрипторов.

Второе условие направлено на обеспечение гарантии того, что по возможности поддескрипторы различных групп будут извлекаться с относительно отдаленных мест сетки поддескрипторов, в результате чего будет повышаться информационное наполнение дескрипторов с очень небольшим количеством признаков. Следует отметить,

что второе условие не всегда может удовлетворяться, например, оно не может удовлетворяться для группы, которая содержит все центральные поддескрипторы v_5 , v_6 , v_9 и v_{10} .

Например, одна такая группировка показана на фигуре 17, где имеются четыре группы, а именно $g_0=\{v_0, v_3, v_{12}, v_{15}\}$, содержащая поддескрипторы с наибольшими расстояниями до центра дескриптора, $g_1=\{v_1, v_7, v_8, v_{14}\}$, содержащая набор из четырех поддескрипторов со вторыми наибольшими расстояниями до центра дескриптора, $g_2=\{v_2, v_4, v_{11}, v_{13}\}$, содержащая набор из четырех различных поддескрипторов, и опять со вторыми наибольшими расстояниями до центра дескриптора, и $g_3=\{v_5, v_6, v_9, v_{10}\}$, содержащая поддескрипторы с наименьшими расстояниями до центра дескриптора. Таким образом, в этом примере поддескрипторы из групп g_0 , g_1 и g_2 удовлетворяют условию, поскольку они не имеют соседних поддескрипторов. В каждой группе всем соответствующим элементам поддескрипторов из группы присвоен один и тот же приоритет кодирования. На основании этой группировки можно сгенерировать порядок использования групповых элементов, такой как порядок, показанный на фигуре 15, и в таком случае кодер, такой как кодер, показанный на фигуре 11, опять можно использовать для кодирования дескриптора в соответствии с порядком использования групповых элементов из фигуры 15.

Ясно, что альтернативные условия, основанные на расстояниях поддескрипторов в группе, также можно использовать, чтобы максимизировать полные расстояния между поддескрипторами, и т.д.

Третий вариант осуществления

В третьем варианте осуществления изобретения поддескрипторы группируют в соответствии с их расстоянием от центра и поддескрипторы из каждой группы кодируют в последовательности, определяемой в соответствии с их характеристиками кодирования, такими как их соответствующие преобразования.

Например, при рассмотрении группировки из фигуры 17 согласно второму варианту осуществления изобретения в сочетании с картиной выполнения преобразования из фигуры 3 видно, что в каждой группе два поддескриптора преобразуются в соответствии с преобразованием А и два поддескриптора преобразуются в соответствии с преобразованием В. Поэтому можно задать общее условие последовательности кодирования, в соответствии с которым для каждой группы используемая последовательность преобразования, то есть последовательность кодирования, должна быть "А А В В", в результате чего для каждой группы поддескрипторов преобразование первого поддескриптора в группе является преобразованием А, преобразование второго поддескриптора в группе также является преобразованием А, преобразование третьего поддескриптора в группе является преобразованием В и преобразование четвертого поддескриптора в группе также является преобразованием В. Таким образом, имеются группы $g_0=\{v_0, v_{15}, v_3, v_{12}\}$, содержащая поддескрипторы с наибольшими расстояниями до центра дескриптора, $g_1=\{v_7, v_8, v_1, v_{14}\}$, содержащая набор из четырех поддескрипторов со вторыми наибольшими расстояниями до центра дескриптора, $g_2=\{v_2, v_{13}, v_4, v_{11}\}$, содержащая набор из четырех других поддескрипторов, и опять со вторыми наибольшими расстояниями до центра дескриптора, и $g_3=\{v_5, v_{10}, v_6, v_9\}$, содержащая поддескрипторы с наименьшими расстояниями до центра дескриптора. В каждой группе всем соответствующим элементам поддескрипторов из группы присвоен одинаковый приоритет кодирования. Кроме того, в каждой группе информация об

использовании преобразования для четырех поддескрипторов в группе всегда имеет вид "А А В В", и это означает, что реализация эффективного кодера не нужна для идентификации преобразования, применяемого к каждому поддескриптору.

На фигуре 18 показана работа такого кодера с использованием порядка использования групповых элементов из фигуры 15, выполненного с возможностью кодирования верхних m групп элементов в указанном порядке использования групповых элементов. Кодером из фигуры 18 кодирование дескриптора начинается с группы элементов с высшим приоритетом (приоритетом 1) в порядке использования групповых элементов, то есть с элемента v_0 из группы g_3 , содержащей поддескрипторы v_5 , v_{10} , v_6 и v_9 . На этапе S400 кодирование дескриптора начинается с первого поддескриптора из группы, то есть v_5 . При условии, что все группы имеют общий и фиксированный порядок использования преобразования, при обработке осуществляется переход к этапу S410, на котором выполняется кодирование элемента, то есть $v_{5,0}$. На этапе S420, если текущий поддескриптор не является последним поддескриптором в группе, при обработке осуществляется переход к следующему поддескриптору, в противном случае при обработке осуществляется переход к этапу S430. Затем на этапе S430, если еще не выполнено кодирование предварительно определенного количества групп из m групп элементов, при обработке осуществляется переход к группе элементов со следующим наивысшим приоритетом в порядке использования групповых элементов, в противном случае обработка заканчивается. Таким образом, этапы S400, S420 и S430 относятся к порядку, в котором выполняется обработка и к контролю количества кодированных групп элементов, тогда как этап S410 относится только к фактическому кодированию локального дескриптора изображения.

В приведенном описании общее условие последовательности кодирования определено на основании преобразований, используемых в каждой группе, но это условие также может быть определено на основании других характеристик кодирования, таких как вид и уровень квантования, или сочетаний их.

Ясно, что если требуется, другие условия последовательности кодирования можно применить к другим группам поддескрипторов. Например, в первом варианте осуществления группирование в соответствии с фигурой 9 приводит к генерированию трех групп, а именно g_0 , содержащей четыре поддескриптора с наибольшими расстояниями до центра дескриптора, g_1 , содержащей восемь поддескрипторов со вторыми наибольшими расстояниями до центра дескриптора, и g_2 , содержащей четыре поддескриптора с наименьшими расстояниями до центра дескриптора. В этом случае вследствие различных размеров групп условие последовательности кодирования, в соответствии с которым последовательность использования преобразования должна быть "А А В В", можно применить к группам g_0 и g_2 , а другое условие последовательности кодирования, в соответствии с которым последовательность использования преобразования должна быть "А А А А В В В В", можно применить к группе g_1 .

Четвертый вариант осуществления

В четвертом варианте осуществления изобретения поддескрипторы группируют в соответствии с их расстоянием от центра, и поддескрипторы каждой группы упорядочивают в соответствии с их расстояниями между ними.

Условие упорядочения может заключаться, например, в том, что расстояния между последовательными поддескрипторами в группе не должны быть ниже предварительно

определенного порога. Поэтому предварительно определенный порог можно устанавливать для исключения, например, последовательных соседних поддескрипторов в группе. Другое условие упорядочения может заключаться, например, в том, что расстояния между последовательными поддескрипторами в группе должны быть

увеличенными до максимума.

Как поясняется ниже, это упорядочение является особенно полезным при использовании больших групп поддескрипторов для снижения объема порядка использования групповых элементов и частичного кодирования групп. Следует отметить, что упорядочение этого вида не всегда возможно, например оно невозможно, когда необходимо не иметь последовательных соседних дескрипторов для группы, которая содержит все центральные поддескрипторы v_5, v_6, v_9 и v_{10} .

Например, при рассмотрении группировки, показанной на фигуре 9, можно видеть, что поддескрипторы в каждой группе можно упорядочить для максимизации расстояний между последовательными дескрипторами, начиная от поддескриптора с наименьшим индексом в группе, что приведет к образованию групп $g_0=\{v_0, v_{15}, v_3, v_{12}\}$, $g_1=\{v_1, v_{14}, v_2, v_{13}, v_7, v_8, v_{11}, v_4\}$ и $g_2=\{v_5, v_{10}, v_6, v_9\}$. В каждой группе всем соответствующим элементам поддескрипторов из группы присвоен один и тот же приоритет кодирования.

На основе этой группировки и упорядочения поддескрипторов в каждой группе можно сгенерировать порядок использования групповых элементов, такой как порядок, показанный на фигуре 10, и в таком случае кодер, такой как кодер, показанный на фигуре 11, можно опять использовать для кодирования дескриптора в соответствии с порядком использования групповых элементов из фигуры 10. В качестве варианта можно использовать такой кодер, какой показан на фигуре 19.

Более конкретно, одно различие между кодером из фигуры 7 и кодерами из фигуры 11 и фигуры 18 заключается в том, что в отличие от первого, который образует дескрипторы любой длины, последний позволяет кодировать дескрипторы, длина которых имеет степень детализации, определяемую группами поддескрипторов. Однако на практике каждый из кодеров из фигур 11 и 18 можно легко выполнять с возможностью кодирования дескрипторов любой длины путем частичного кодирования последней группы элементов по достижении дескриптора предварительно определенной длины. В этой связи на фигуре 19 показана такая модификация кодера из фигуры 11. По существу, кодер из фигуры 19 получен путем простой перестановки этапов S330 и S340 для кодера из фигуры 11, что позволяет кодеру из фигуры 19 завершать кодирование группы после того, как будет кодировано предварительно определенное количество элементов. Ясно, что аналогичная модификация также применима к кодеру из фигуры 18.

В этой связи очень предпочтительно упорядочение поддескрипторов в группе в соответствии с расстояниями между ними, поскольку оно дает в результате последовательные элементы, извлекаемые с относительно отдаленных мест сетки поддескрипторов, что в случае частичного кодирования группы повышает информационное наполнение дескрипторов с небольшим количеством признаков.

Пятый вариант осуществления

В предшествующих вариантах осуществления изобретения было показано эффективное кодирование дескрипторов изображения в соответствии с порядком использования групповых элементов, полученным в результате группирования поддескрипторов в группы поддескрипторов на основании их расстояний от центра дескриптора и/или их расстояний друг от друга.

В альтернативном варианте осуществления поддескрипторы можно группировать так, чтобы для каждого поддескриптора в группе имелся такой же набор использования элементов, как для всех других поддескрипторов в группе.

Например, рассмотрим группировку из фигуры 20, в которой имеются четыре группы, а именно $g_0=\{v_5, v_6, v_9, v_{10}\}$, $g_1=\{v_1, v_7, v_8, v_{14}\}$, $g_2=\{v_2, v_4, v_{11}, v_{13}\}$ и $g_3=\{v_0, v_3, v_{12}, v_{15}\}$. Она является такой же группировкой, какая показана на фиг. 17, но индексы групп переприсвоены (то есть g_0 и g_3 переставлены), так что группы с низкими индексами содержат поддескрипторы, которые находятся ближе к центру дескриптора. Сочетание этой группировки с наборами использования элементов из фигуры 4а-4е приводит к наборам использования групповых элементов из фигуры 21а-21е.

Поскольку набор используемых элементов для дескриптора каждой длины должен быть таким же, как поднабор, или самим поднабором из набора используемых элементов для дескрипторов всех больших длин, наборы использования групповых элементов из фигуры 21а-21е можно преобразовать в инкрементные наборы использования групповых элементов, показанные на фигуре 22а-22е, где для дескриптора каждой длины (например DL2 из фигуры 22с) в соответствующем перестроенном наборе элементов показаны только дополнительные элементы, которые образуют дескриптор указанной длины в сопоставлении с дескриптором ближайшей меньшей длины (например DL1 из фигуры 22b).

На основании инкрементных наборов использования групповых элементов из фигуры 22а-22е кодер может генерировать дескрипторы, которые могут быть преобразованы в дескрипторы меньших длин путем простого усечения дескриптора.

На фигуре 23 показана работа такого кодера при кодировании дескриптора длиной DLk. Более конкретно, на этапе S600 обработка начинается с кодирования дескриптора наименьшей длины, то есть DL0. На этапе S610 кодирование дескриптора длиной DL0 начинается с кодирования первой группы поддескрипторов, то есть g_0 , и на этапе S620 кодирование первой группы поддескрипторов начинается с кодирования первого элемента, то есть v_0 . На этапе S630, если элемент v_0 не использовался для g_0 в

дескрипторе длиной DL0 в соответствии с инкрементными наборами использования групповых элементов из фигуры 22, при обработке осуществляется переход к этапу S680, в противном случае при обработке осуществляется переход к этапу S640. На этапе S640 выбирается первый поддескриптор из группы g_0 , то есть v_5 , и на этапе S650

выбирается подходящая функция преобразования, например, в соответствии с фигурой 3. Следует отметить, что вычисление дескриптора V по дескриптору H в соответствии с двумя различными преобразованиями, описанными в этой заявке, является только примером. В других вариантах осуществления вычисление дескриптора V по дескриптору H может также выполняться в соответствии с одним преобразованием, например только преобразованием A или только преобразованием B, при этом выполнение этапа S650 не является необходимым, или в соответствии с более чем двумя преобразованиями.

Затем на этапе S660 выполняется кодирование элемента $v_{5,0}$. На этапе S670, если текущий поддескриптор не является последним дескриптором в группе, при обработке осуществляется переход к следующему поддескриптору в группе, в противном случае при обработке осуществляется переход к этапу S680. На этапе S680, если текущий элемент не является последним элементом, то есть v_7 , при обработке осуществляется переход к следующему элементу, в противном случае при обработке осуществляется переход к этапу S690. На этапе S690, если текущая группа поддескрипторов не является последней группой поддескрипторов, при обработке осуществляется переход к

следующей группе поддескрипторов, в противном случае при обработке осуществляется переход к этапу S695. На этапе S695, если поддескриптор текущей длины не является поддескриптором предварительно определенной длины, при обработке осуществляется переход к поддескриптору следующей длины, выполняется кодирование дополнительных элементов, определяемых в соответствии с инкрементными наборами использования групповых элементов. В противном случае обработка заканчивается.

Хотя аспекты и варианты осуществления настоящего изобретения подробно изложены применительно к вычислению робастных, различаемых, масштабируемых и компактных дескрипторов изображения по SIFT дескриптору изображения, изобретение применимо к другим дескрипторам изображения на основе гистограмм градиентов, представленным в еще не опубликованной заявке № TO2012A000602 на патент Италии.

Только для примера на фигуре 24 показано концептуальное устройство для обработки изображений, предназначенное для осуществления способа согласно настоящему изобретению. Более конкретно, устройство 1100 для обработки принимает входные данные, которые могут содержать визуальные данные, такие как данные изображений или видеоданные, предварительно вычисленные дескрипторы на основе гистограмм градиентов, предварительно вычисленные компактные дескрипторы согласно способу настоящего изобретения, команды на программирование или входные данные пользователя с устройства 1000 ввода, которое может иметь вид пользовательского устройства ввода, считывателя с носителя или приемника передаваемых сигналов. Устройство 1100 для обработки содержит основные блоки обработки в центральном процессоре 1100, который управляет работой других блоков обработки, энергонезависимое запоминающее устройство (ЗУ) 1120, энергозависимое запоминающее устройство (ЗУ) 1130, необязательный блок 1140 извлечения дескрипторов, выполненный с возможностью генерирования дескрипторов на основании гистограмм градиентов, кодер 1150 компактных дескрипторов, выполненный с возможностью осуществления способа согласно настоящему изобретению, и необязательный процессор 1160 компактных дескрипторов, выполненный с возможностью обработки указанных компактных дескрипторов, например, для установления или проверки визуальных соответствий. Устройство 1100 для обработки соединено с устройством 1900 вывода, которое может иметь вид визуального устройства отображения, устройства записи на носители или передатчик сигналов, который предоставляет выходные данные, которые могут содержать аннотированные визуальные данные, такие как данные изображений или видеоданные, текущую информацию, такую как установленные или проверенные визуальные соответствия, или компактные дескрипторы согласно способу настоящего изобретения. Следует понимать, что блоки обработки и архитектура, показанные на фигуре 24, являются только концептуальными и могут не точно соответствовать каждому устройству, реализующему способ согласно настоящему изобретению.

Способ кодирования дескриптора изображения на основе гистограммы градиентов, и связанное с ним устройство для обработки изображений, описанные в этой заявке для примера, могут подвергаться многочисленным возможным изменениям без отступления от новой сущности идеи изобретения; кроме того, ясно, что при практическом осуществлении изобретения показанные подробности могут иметь различные формы или могут быть заменены другими, технически эквивалентными элементами.

Поэтому можно легко понять, что настоящее изобретение не ограничено способом кодирования дескриптора изображения на основе гистограммы градиентов, и связанным с ним устройством для обработки изображений, а может подвергаться многочисленным

модификациям, усовершенствованиям или заменам эквивалентных частей и элементов без отступления от идеи изобретения, ясно определенной в нижеследующей формуле изобретения.

(57) Формула изобретения

1. Способ кодирования дескриптора изображения на основе гистограмм градиентов в преобразованный дескриптор, содержащий преобразованные поддескрипторы, при этом каждая гистограмма градиентов содержит множество бинов гистограммы, и каждый поддескриптор содержит набор значений,

при этом для каждого поддескриптора упомянутый набор значений поддескриптора получают посредством преобразования каждого из бинов гистограммы соответствующей гистограммы градиентов и затем выполнения скалярного квантования полученных преобразованных значений, и

при этом упомянутые поддескрипторы генерируют в соответствии со списком порядка использования элементов, определяющим индекс поддескрипторов из упомянутого набора поддескрипторов и индекс элементов из упомянутого набора значений.

2. Способ по п.1, в котором упомянутые поддескрипторы группируют в группы поддескрипторов в соответствии с их расстоянием от центра дескриптора изображения, и при этом в каждой группе поддескрипторов всем соответствующим значениям из группы поддескрипторов присваивают один и тот же приоритет кодирования в упомянутом списке порядка использования элементов.

3. Способ по п.2, в котором упомянутые поддескрипторы группируют в первую группу, содержащую все поддескрипторы с наименьшим расстоянием до центра дескриптора изображения, и вторую группу, содержащую все остальные поддескрипторы.

4. Способ по п.2, в котором упомянутые поддескрипторы группируют в первую группу, содержащую поддескрипторы с наибольшим расстоянием от центра дескриптора, вторую группу, содержащую первый набор поддескрипторов со вторым наибольшим расстоянием до центра дескриптора, третью группу, содержащую второй набор поддескрипторов со вторым наибольшим расстоянием до центра дескриптора, и четвертую группу, содержащую поддескрипторы с наименьшим расстоянием до центра дескриптора.

5. Способ по п.4, в котором упомянутые первая, вторая, третья и четвертая группы содержат одно и то же количество поддескрипторов.

6. Способ по п.1, в котором упомянутые поддескрипторы группируют в группы поддескрипторов в соответствии с их расстоянием от центра дескриптора изображения и в соответствии с их взаимными расстояниями, и в котором в каждой группе поддескрипторов всем соответствующим значениям из группы поддескрипторов присваивают один и тот же приоритет кодирования в упомянутом списке порядка использования элементов.

7. Способ по п.6, в котором упомянутые поддескрипторы группируют в первую группу, содержащую поддескрипторы с наибольшим расстоянием от центра дескриптора, вторую группу, содержащую первый набор поддескрипторов со вторым наибольшим расстоянием до центра дескриптора, третью группу, содержащую второй набор поддескрипторов со вторым наибольшим расстоянием до центра дескриптора, и четвертую группу, содержащую поддескрипторы с наименьшим расстоянием до центра дескриптора.

8. Способ по п.6, в котором группа поддескрипторов не содержит поддескрипторов,

расстояния которых друг от друга ниже предварительно определенного порога.

9. Способ по п.6, в котором упомянутые взаимные расстояния заключают в себе максимизацию полных расстояний между поддескрипторами.

10. Способ по п.1, в котором упомянутые поддескрипторы группируют в группы поддескрипторов в соответствии с их расстоянием от центра дескриптора изображения и упорядочивают в соответствии с их взаимными расстояниями, и в котором в каждой группе поддескрипторов всем соответствующим значениям из группы поддескрипторов присваивают один и тот же приоритет кодирования в упомянутом списке порядка использования элементов.

11. Способ по п.10, в котором группа поддескрипторов не содержит последовательных поддескрипторов, взаимные расстояния которых ниже предварительно определенного порога.

12. Способ по п.10, в котором упомянутые взаимные расстояния заключают в себе максимизацию расстояний между последовательными поддескрипторами.

13. Способ по п.1, в котором упомянутые поддескрипторы группируют в группы поддескрипторов в соответствии с их расстоянием от центра дескриптора изображения, и поддескрипторы каждой группы кодируют в последовательности, определяемой в соответствии с их соответствующими характеристиками кодирования, и в котором в каждой группе поддескрипторов всем соответствующим значениям из группы поддескрипторов присваивают один и тот же приоритет кодирования в упомянутом списке порядка использования элементов.

14. Способ по п.13, в котором упомянутые характеристики кодирования представляют собой соответствующие преобразования.

15. Способ по п.14, в котором для каждой группы поддескрипторов преобразование первого поддескриптора в группе представляет собой преобразование первого вида, преобразование второго поддескриптора в группе представляет собой преобразование упомянутого первого вида, преобразование третьего поддескриптора в группе представляет собой преобразование второго вида, и преобразование четвертого поддескриптора в группе представляет собой преобразование упомянутого второго вида.

16. Способ по п.13, в котором упомянутые характеристики кодирования представляют собой вид и уровень квантования или их сочетания.

17. Способ по п.2, в котором более высокий приоритет придают заданному классу значений.

18. Способ по п.2, в котором более высокий приоритет придают группам поддескрипторов, ближайшим к центру дескриптора.

19. Способ по любому из пп.2-18, в котором упомянутый список порядка использования элементов сохраняют или передают вместе с дескриптором изображения.

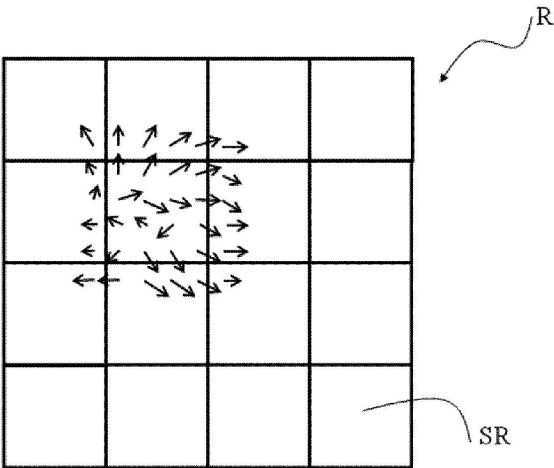
20. Способ кодирования дескриптора изображения по любому из пп.1-19, в котором упомянутый дескриптор изображения преобразуют в дескриптор, имеющий уменьшенную длину за счет отсечения одного или нескольких последних элементов упомянутого дескриптора.

21. Способ кодирования дескриптора изображения по п.1, в котором упомянутый список порядка использования элементов имеет вид 16-, или 24-, или 32-, или 128-элементного упорядоченного списка.

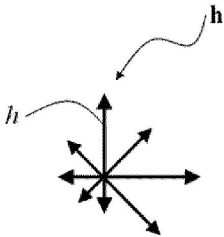
22. Устройство обработки изображений, содержащее средство для осуществления способа по любому из пп.1-21.

1

1/22



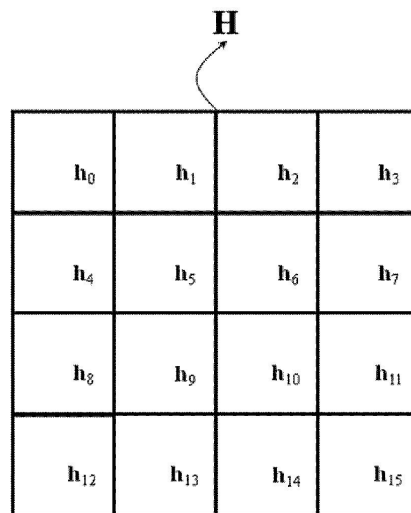
ФИГ. 1А



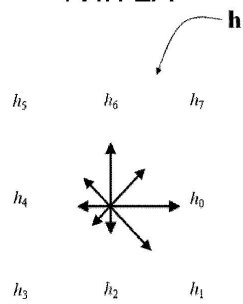
ФИГ. 1В

2

2/22

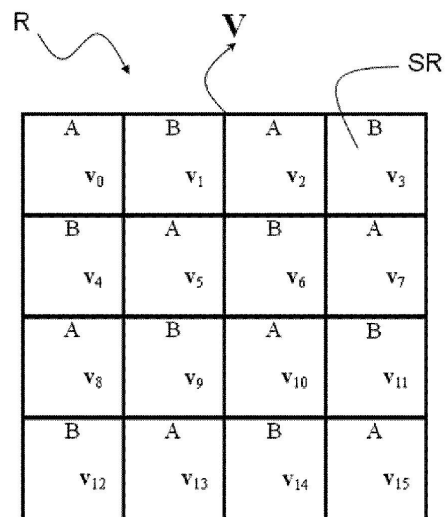


ФИГ. 2А



ФИГ. 2В

3/22



ФИГ. 3

4/22

V ₀					V ₁					V ₂					V ₃				
V ₀					V ₀					V ₀					V ₀				
V ₄					V ₅					V ₆					V ₇				
V ₀					V ₀				V ₆	V ₀				V ₆	V ₀				
V ₈					V ₉					V ₁₀					V ₁₁				
V ₀					V ₀				V ₀	V ₀				V ₀	V ₀				
V ₁₂					V ₁₃					V ₁₄					V ₁₅				
V ₀					V ₀					V ₀					V ₀				

ФИГ. 4А

V ₀					V ₁					V ₂					V ₃				
V ₀	V ₁				V ₀	V ₁				V ₀	V ₁				V ₀	V ₁			
V ₄					V ₅					V ₆					V ₇				
V ₀	V ₁				V ₀	V ₁	V ₀		V ₆	V ₀	V ₁	V ₀		V ₆	V ₀	V ₁			
V ₈					V ₉					V ₁₀					V ₁₁				
V ₀	V ₁				V ₀	V ₁	V ₂		V ₆	V ₀	V ₁	V ₂		V ₆	V ₀	V ₁			
V ₁₂					V ₁₃					V ₁₄					V ₁₅				
V ₀	V ₁				V ₀	V ₁				V ₀	V ₁				V ₀	V ₁			

ФИГ. 4В

V ₀					V ₁					V ₂					V ₃				
V ₀	V ₁	V ₂			V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		
V ₄					V ₅					V ₆					V ₇				
V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀
V ₈					V ₉					V ₁₀					V ₁₁				
V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀
V ₁₂					V ₁₃					V ₁₄					V ₁₅				
V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀

ФИГ. 4С

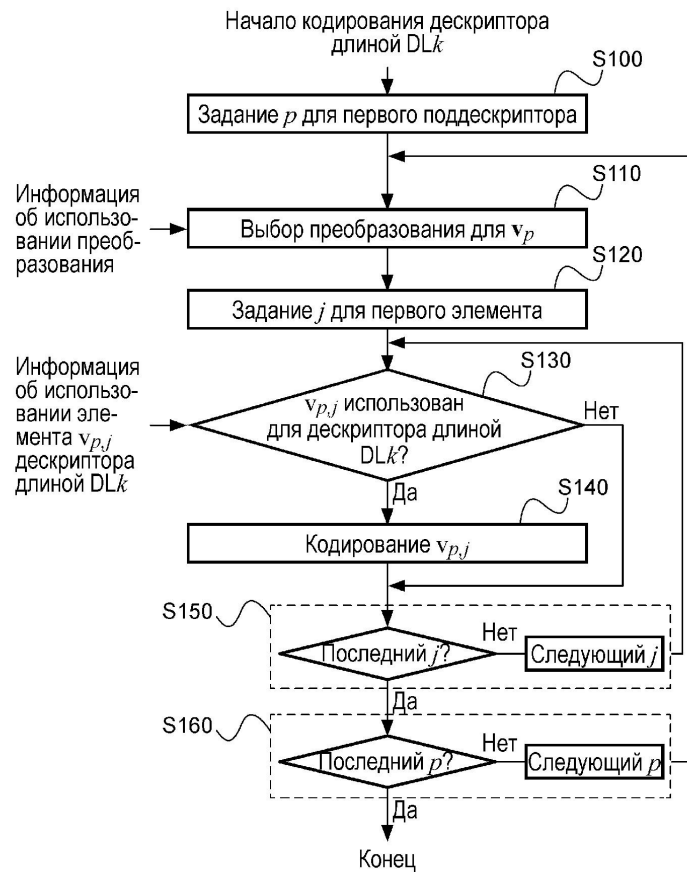
V ₀					V ₁					V ₂					V ₃				
V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀
V ₄					V ₅					V ₆					V ₇				
V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀
V ₈					V ₉					V ₁₀					V ₁₁				
V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀
V ₁₂					V ₁₃					V ₁₄					V ₁₅				
V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀	V ₀	V ₁	V ₂		V ₀

ФИГ. 4D

V ₀					V ₁					V ₂					V ₃				
V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇			V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇		
V ₄					V ₅					V ₆					V ₇				
V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇		V ₀	V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇		V ₀
V ₈					V ₉					V ₁₀					V ₁₁				
V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇		V ₀	V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇		V ₀
V ₁₂					V ₁₃					V ₁₄					V ₁₅				
V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇		V ₀	V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇		V ₀

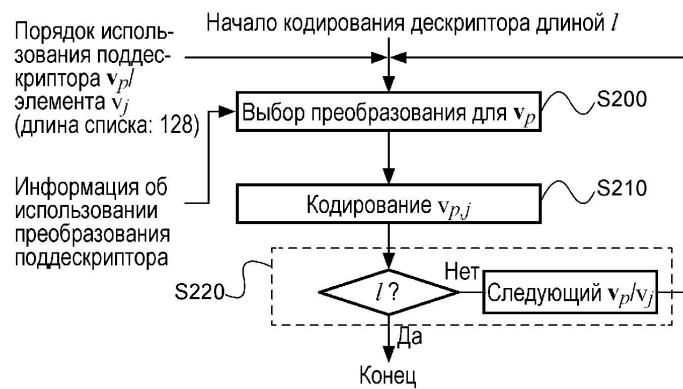
ФИГ. 4Е

5/22

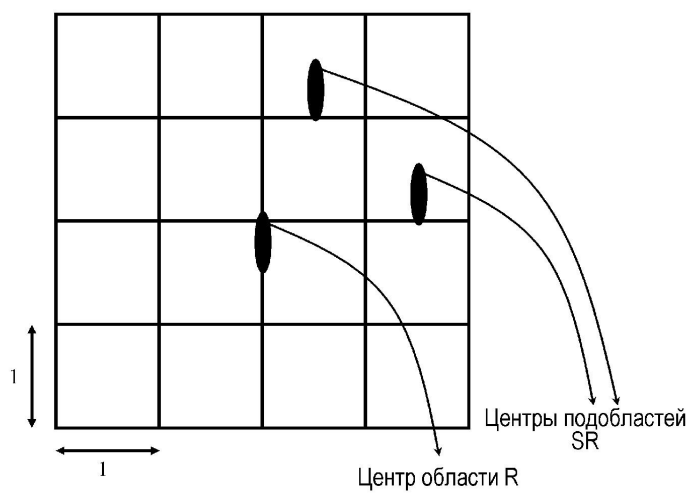


ФИГ. 5

7/22

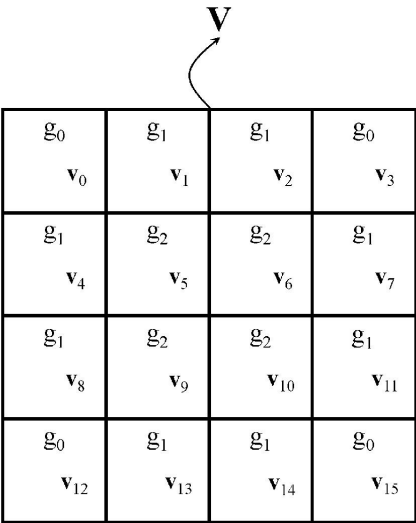


ФИГ. 7



ФИГ. 8

8/22



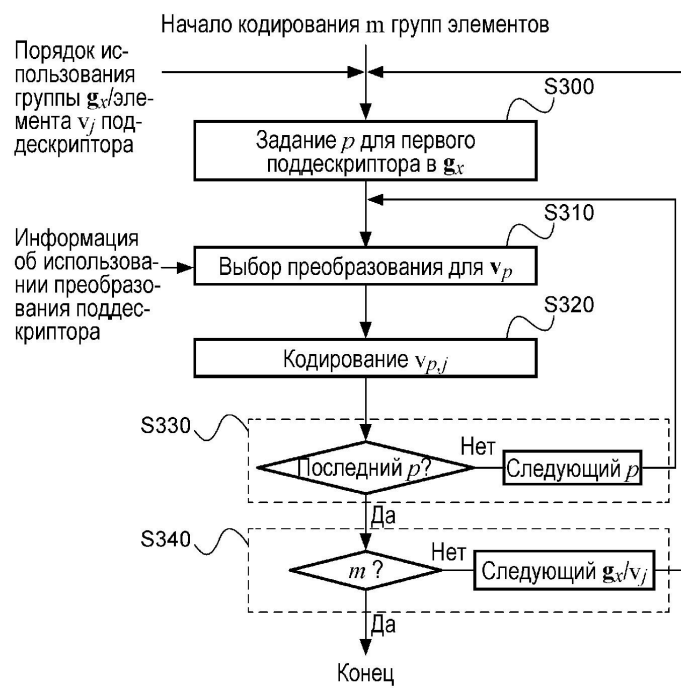
ФИГ. 9

9/22

Приоритет	Группа g_x	Элемент v_j
1	2	0
2	1	0
3	0	0
4	2	6
5	2	1
6	1	1
7	0	1
8	2	2
9	1	2
10	0	2
11	1	6
12	0	6
13	2	7
14	1	7
15	0	7
16	2	3
17	1	3
18	0	3
19	2	4
20	1	4
21	0	4
22	2	5
23	1	5
24	0	5

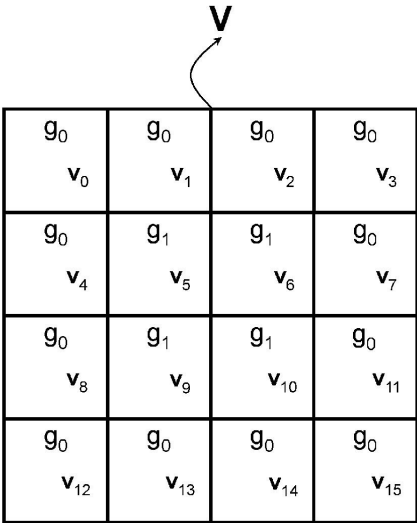
ФИГ. 10

10/22



ФИГ. 11

11/22



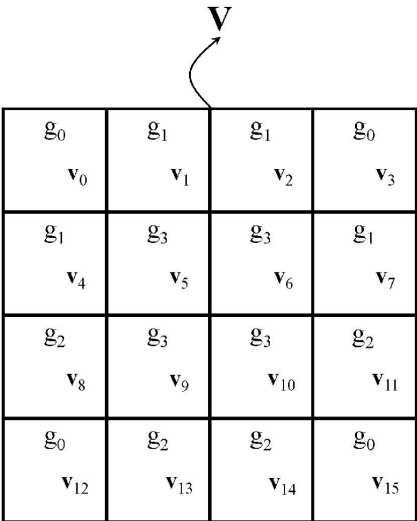
ФИГ. 12

12/22

Приоритет	Группа g_x	Элемент v_j
1	1	0
2	0	0
3	1	6
4	1	1
5	0	1
6	1	2
7	0	2
8	0	6
9	1	7
10	0	7
11	1	3
12	0	3
13	1	4
14	0	4
15	1	5
16	0	5

ФИГ. 13

13/22



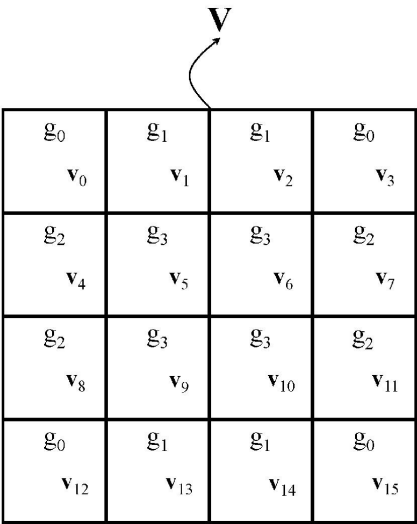
ФИГ. 14

14/22

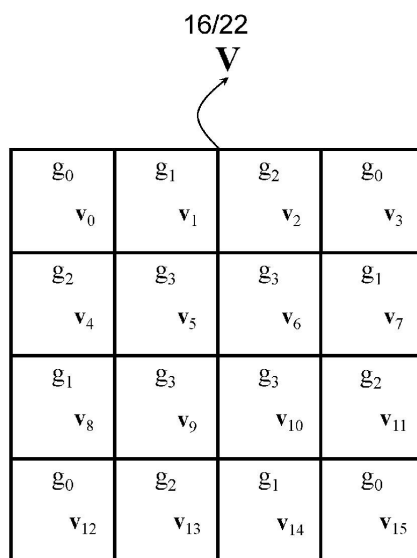
Приоритет	Группа g_x	Элемент v_j
1	3	0
2	1	0
3	2	0
4	0	0
5	3	6
6	3	1
7	1	1
8	2	1
9	0	1
10	3	2
11	1	2
12	2	2
13	0	2
14	1	6
15	2	6
16	0	6
17	3	7
18	1	7
19	2	7
20	0	7
21	3	3
22	1	3
23	2	3
24	0	3
25	3	4
26	1	4
27	2	4
28	0	4
29	3	5
30	1	5
31	2	5
32	0	5

ФИГ. 15

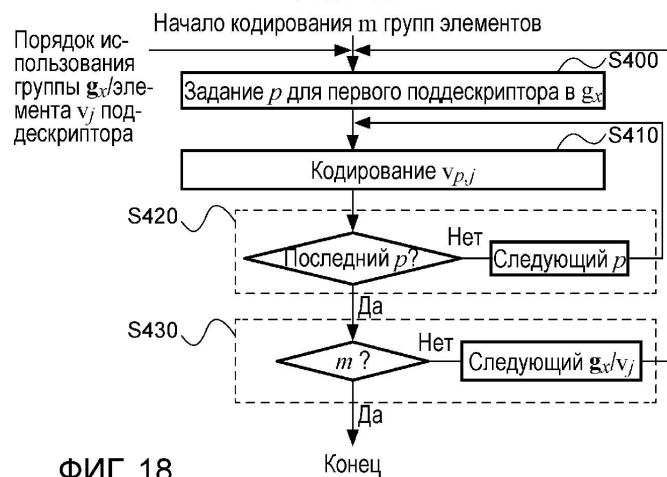
15/22



ФИГ. 16

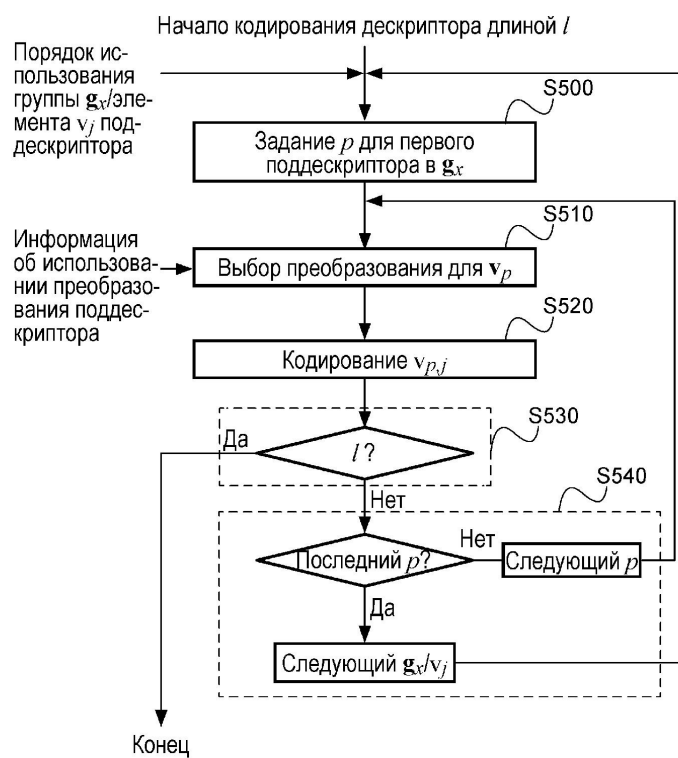


ФИГ. 17



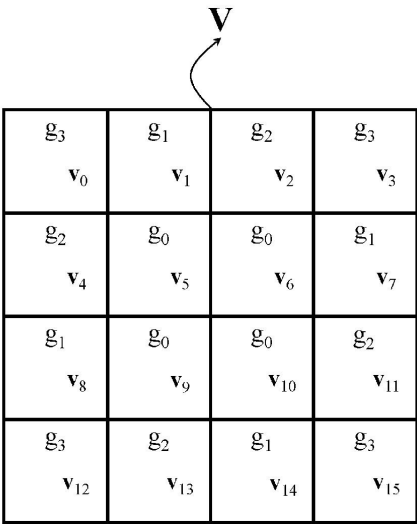
ФИГ. 18

17/22



ФИГ. 19

18/22



ФИГ. 20

g_0				g_1				g_2				g_3			
v_0				v_6			v_0				v_0				v_0

ФИГ. 21А

g_0				g_1				g_2				g_3			
v_0	v_1	v_2		v_6			v_0	v_1			v_0	v_1			v_0

ФИГ. 21В

g_0				g_1				g_2				g_3			
v_0	v_1	v_2		v_6			v_0	v_1	v_2		v_6			v_0	v_6

ФИГ. 21С

g_0				g_1				g_2				g_3			
v_0	v_1	v_2		v_6	v_7	v_0	v_1	v_2	v_6	v_7	v_0	v_1	v_2		v_6

ФИГ. 21D

g_0				g_1				g_2				g_3			
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_1	v_2	v_3	v_4	v_5	v_6	v_1	v_2	v_3

ФИГ. 21Е

[illegible]

ФИГ. 22А

$\mathbf{g}_0$	$\mathbf{g}_1$				$\mathbf{g}_2$				$\mathbf{g}_3$
v_1	v_2	v_3	v_4	v_1	v_2	v_3	v_4	v_1	v_2

ФИГ. 22В

$\mathfrak{g}_0$					$\mathfrak{g}_1$					$\mathfrak{g}_2$					$\mathfrak{g}_3$				

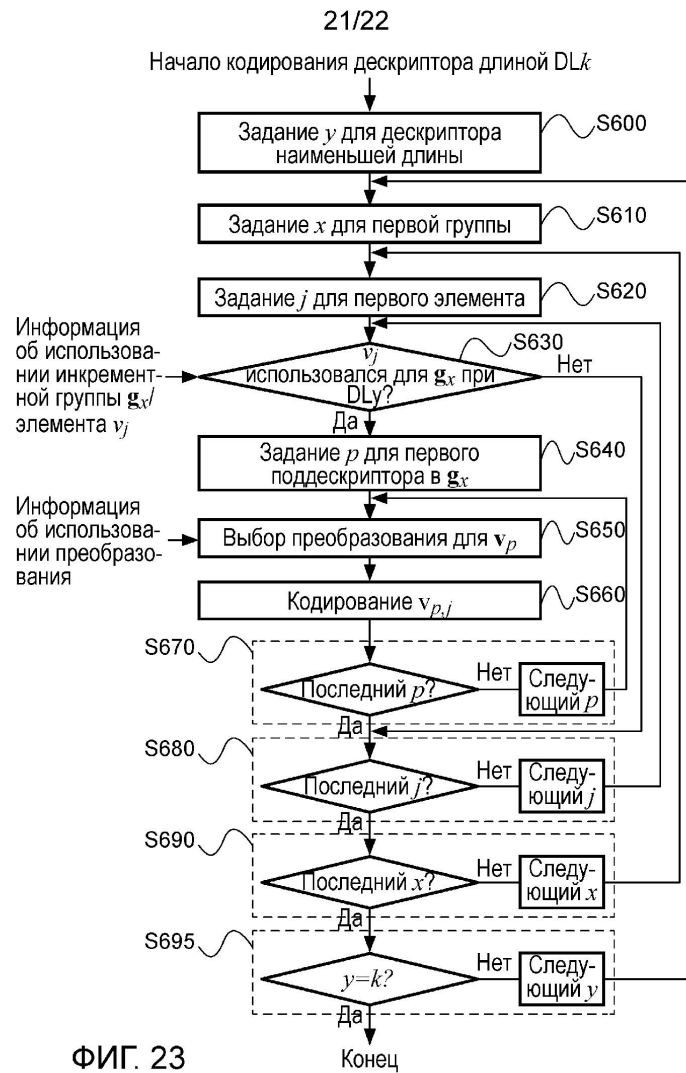
ФИГ. 22С

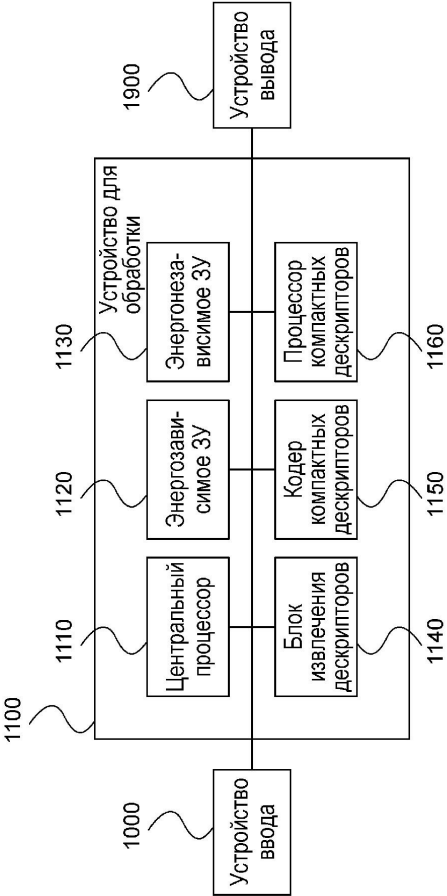
$\mathbf{q}_0$	$\mathbf{q}_1$	$\mathbf{q}_2$	$\mathbf{q}_3$
v_7	v_7	v_7	v_7

Фиг. 22D

[illegible]

ФИГ. 22Е





ФИГ. 24