



ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

H04N 19/96 (2006.01); H04N 19/186 (2006.01); H04N 1/6019 (2006.01)

(21)(22) Заявка: 2015147556, 17.03.2014

(24) Дата начала отсчета срока действия патента:  
17.03.2014

Дата регистрации:  
24.09.2018

Приоритет(ы):

(30) Конвенционный приоритет:

08.04.2013 EP 13305453.6;

15.07.2013 EP 13306010.3;

27.01.2014 EP 14305109.2

(43) Дата публикации заявки: 16.05.2017 Бюл. № 14

(45) Опубликовано: 24.09.2018 Бюл. № 27

(85) Дата начала рассмотрения заявки РСТ на  
национальной фазе: 09.11.2015

(86) Заявка РСТ:  
EP 2014/055333 (17.03.2014)

(87) Публикация заявки РСТ:  
WO 2014/166705 (16.10.2014)

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

(72) Автор(ы):

БОРД Филипп (FR),  
АНДРИВОН Пьер (FR),  
ЖОЛИ Эммануэль (FR)

(73) Патентообладатель(и):

ДОЛБИ ИНТЕРНЭШНЛ АБ (NL)

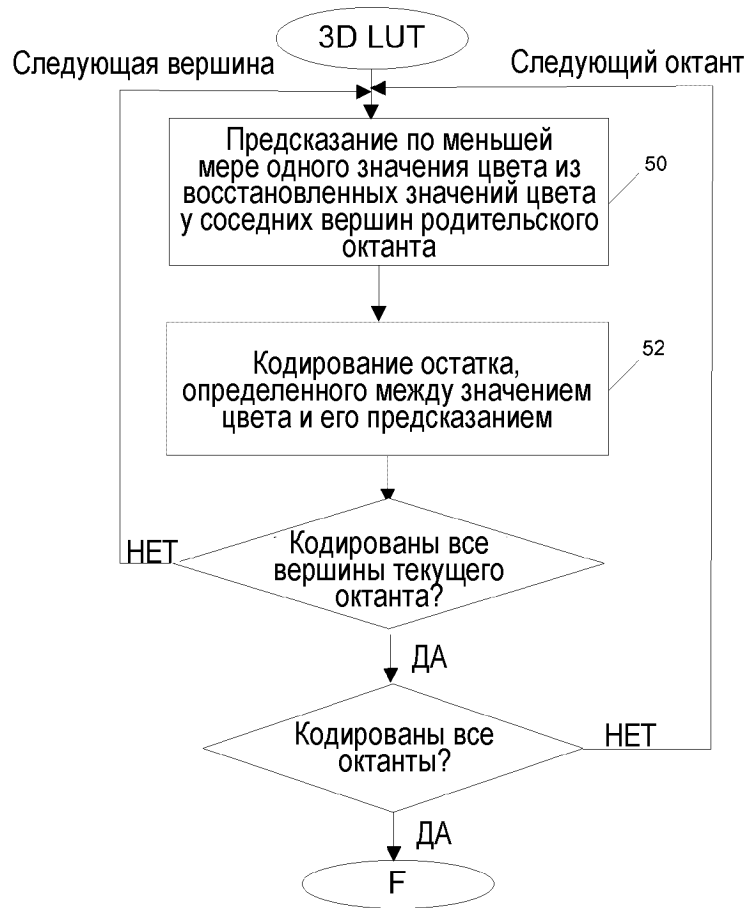
(56) Список документов, цитированных в отчете  
о поиске: WO 2012/044886 A1, 05.04.2012. US  
2009/0016598 A1, 15.01.2009. US 7265870 B2,  
04.09.2007. US 2006/0268297 A1, 30.11.2006.  
RU 2447607 C1, 10.04.2012.

(54) СПОСОБ КОДИРОВАНИЯ И СПОСОБ ДЕКОДИРОВАНИЯ LUT И СООТВЕТСТВУЮЩИЕ  
УСТРОЙСТВА

(57) Реферат:

Изобретение относится к вычислительной технике. Технический результат заключается в повышении эффективности кодирования за счет снижения объема кодируемых данных. Способ кодирования цветовой 3-мерной справочной таблицы, ассоциированной с октантами октодеревы, содержит этапы, на которых

кодируют флаг разделения для текущего октанта, указывающий, кодируется ли рекурсивно по меньшей мере один из его непосредственных дочерних октантов, и кодируют упомянутый текущий октант в ответ на упомянутый флаг разделения. 4 н. и 24 з.п. ф-лы, 18 ил., 9 табл.



ФИГ. 6



FEDERAL SERVICE  
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC

*H04N 19/96 (2006.01); H04N 19/186 (2006.01); H04N 1/6019 (2006.01)*(21)(22) Application: **2015147556, 17.03.2014**(24) Effective date for property rights:  
**17.03.2014**Registration date:  
**24.09.2018**

Priority:

(30) Convention priority:  
**08.04.2013 EP 13305453.6;**  
**15.07.2013 EP 13306010.3;**  
**27.01.2014 EP 14305109.2**(43) Application published: **16.05.2017** Bull. № 14(45) Date of publication: **24.09.2018** Bull. № 27(85) Commencement of national phase: **09.11.2015**(86) PCT application:  
**EP 2014/055333 (17.03.2014)**(87) PCT publication:  
**WO 2014/166705 (16.10.2014)**

Mail address:

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

(72) Inventor(s):

**BORD Filipp (FR),**  
**ANDRIVON Per (FR),**  
**ZHOLI Emmanuel (FR)**

(73) Proprietor(s):

**DOLBI INTERNESHNL AB (NL)**(54) **METHOD FOR ENCODING AND METHOD FOR DECODING A LUT AND CORRESPONDING DEVICES**

(57) Abstract:

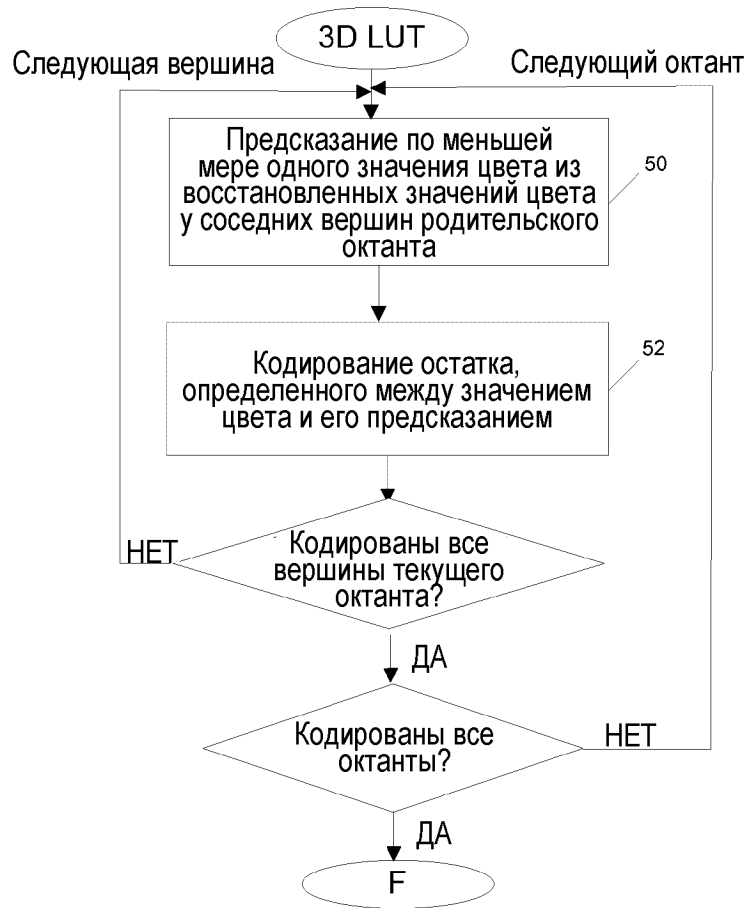
FIELD: electrical communication engineering.

SUBSTANCE: invention relates to the computer equipment. Method for encoding a color 3-dimensional reference table associated with octant octets, comprises the steps of encoding the separation flag for the current octant indicating whether at least one of its immediate

child octants is recursively recoded, and encoding said current octant in response to said division flag.

EFFECT: technical result is to increase the coding efficiency by reducing the amount of encoded data.

28 cl, 18 dwg, 9 tbl



ФИГ. 6

## 1. ОБЛАСТЬ ТЕХНИКИ, К КОТОРОЙ ОТНОСИТСЯ ИЗОБРЕТЕНИЕ

Раскрывается способ кодирования Справочной таблицы, заданной в виде сетки вершин, где по меньшей мере одно значение ассоциируется с каждой вершиной сетки. Для примера, значение является значением цвета. Раскрываются соответствующий способ декодирования, устройство кодирования и устройство декодирования.

## 2. УРОВЕНЬ ТЕХНИКИ ИЗОБРЕТЕНИЯ

Как изображено на фиг.1, масштабируемое декодирование видео заключается в декодировании (соответственно кодированию) потока битов базового слоя (BL) и по меньшей мере одного потока битов улучшающего слоя (EL). Как правило, изображения EL предсказываются из декодированных изображений BL (по возможности, подвергнутых повышающей дискретизации). Однако, когда изображения EL и изображения BL представляются с помощью разных цветовых пространств и/или по-разному откорректированы по цвету, предсказание менее эффективно. Чтобы улучшить предсказание, известно применение преобразования цветов к декодированным изображениям BL. Точнее говоря, преобразование цветов отображает цвета в цветовом пространстве BL (первом цветовом пространстве) в цвета в цветовом пространстве EL (втором цветовом пространстве), используя информацию о цвете.

Как изображено на фиг.2, при распространении видеоконтента преобразование цветов обычно применяется к декодированным изображениям, чтобы преобразованные декодированные изображения были приспособлены к возможности визуализации у конечного устройства.

Это преобразование цветов также известно как Функция отображения цветов (CMF). CMF приближенно выражается, например, матрицей  $3 \times 3$  коэффициентов усиления плюс смещение (Модель усиления-смещения). В этом случае CMF задается 12 параметрами. Однако такое приближение CMF не очень точное, потому что оно предполагает модель линейного преобразования. Следовательно, для описания такой CMF используется Справочная таблица 3D (также известная как 3D LUT), без какой-либо заранее заданной модели CMF. 3D LUT гораздо более точная, потому что ее размер можно увеличивать в зависимости от необходимой точности. Однако 3D LUT может, таким образом, представлять огромный набор данных. Передача 3D LUT в приемник потребует, соответственно, кодирования LUT.

LUT, приближенно выражающая CMF, ассоциирует по меньшей мере одно значение цвета в первом цветовом пространстве с другим значением цвета во втором цветовом пространстве. LUT допускает разбиение первого цветового пространства на набор областей, разделенных вершинами LUT. Для примера 3D LUT ассоциирует тройку значений цвета в первом цветовом пространстве с набором значений цвета. Набор значений цвета может быть тройкой значений цвета во втором цветовом пространстве или набором значений цвета, характеризующим преобразование цветов (например, локально заданные параметры CMF), используемое для преобразования значений цвета в первом цветовом пространстве в значения цвета во втором цветовом пространстве. На фиг.3 квадратная 3D LUT представляется в виде сетки с  $N \times N \times N$  вершинами. Для каждой вершины  $V(c_1, c_2, c_3)$  в 3D LUT нужно хранить соответствующую тройку значений цвета  $(V_{c_1}, V_{c_2}, V_{c_3})$ . Объем данных, ассоциированный с 3D LUT, составляет  $N \times N \times N \times K$ , где  $K$  – количество битов, используемых для хранения одного значения тройки LUT. Значением тройки является, например, тройка  $(R, G, B)$ , тройка  $(Y, U, V)$  или тройка  $(Y, C_b, C_r)$ , и т. п. Кодирование всех значений вершин не эффективно, поскольку представляет огромный объем данных.

## 3. СУЩНОСТЬ ИЗОБРЕТЕНИЯ

Цель изобретения – устранить по меньшей мере один из недостатков известного уровня техники. Раскрывается способ кодирования LUT, заданной в виде сетки вершин. По меньшей мере одно значение ассоциируется с каждой вершиной сетки. Способ кодирования для текущей вершины содержит:

- 5 - предсказание по меньшей мере одного значения, ассоциированного с текущей вершиной, из другого значения; и
- кодирование в поток битов по меньшей мере одного остатка, вычисленного между упомянутым по меньшей мере одним значением текущей вершины и его предсказанием в потоке битов.

10 Преимущественно, что другое значение получается из восстановленных значений, ассоциированных с соседними вершинами.

В соответствии с характерной особенностью изобретения значение является значением цвета. В соответствии с конкретным вариантом осуществления значение цвета соответствует преобразованию цветов. В соответствии с характерной особенностью 15 изобретения значение цвета является параметром усиления или смещением.

В соответствии с конкретным вариантом осуществления LUT является 3D LUT, кодированной с использованием октодеревя, и с каждой вершиной сетки ассоциируются три значения. В этом случае соседние вершины, используемые для предсказания, принадлежат родительскому октанту текущего октанта, которому принадлежит текущая 20 вершина.

В соответствии с характерной особенностью изобретения предсказание по меньшей мере одного значения, ассоциированного с текущей вершиной, из восстановленных значений, ассоциированных с соседними вершинами, содержит интерполяцию по 25 меньшей мере одного значения из соответствующих восстановленных значений соседних вершин.

В соответствии с характерной особенностью способ дополнительно содержит кодирование индекса в поток битов, указывающего тип интерполяции.

В соответствии с конкретным вариантом осуществления кодирование по меньшей мере одного остатка содержит квантование по меньшей мере одного остатка с помощью 30 квантователя в квантованный остаток и энтропийное кодирование квантованного остатка в поток битов, и где способ дополнительно содержит кодирование квантователя в поток битов.

В соответствии с характерной особенностью изобретения для текущей вершины кодируется флаг, указывающий, кодируется ли по меньшей мере один остаток для 35 вершины.

В соответствии с разновидностью для каждого значения текущей вершины кодируется флаг, указывающий, кодируется ли остаток для того значения, или является ли остаток не кодируемым и подразумеваемым равным нулю.

40 Преимущественно, что для текущего октанта кодируется флаг разделения, указывающий, кодируются ли его непосредственные потомки рекурсивно, или подразумеваются ли равными нулю все остатки вершин всех его еще не кодированных потомков.

Также раскрывается способ декодирования LUT, заданной в виде сетки вершин. По меньшей мере одно значение ассоциируется с каждой вершиной сетки. Способ 45 декодирования для текущей вершины содержит:

- декодирование по меньшей мере одного остатка из потока битов;
- предсказание по меньшей мере одного значения, ассоциированного с текущей вершиной, из другого значения; и

- восстановление по меньшей мере значения текущей вершины из его предсказания и по меньшей мере одного декодированного остатка.

Преимущественно, что другое значение получается из восстановленных значений, ассоциированных с соседними вершинами.

5 В соответствии с характерной особенностью изобретения значение является значением цвета. В соответствии с конкретным вариантом осуществления значение цвета соответствует преобразованию цветов. В соответствии с характерной особенностью изобретения значение цвета является параметром усиления или смещением.

10 В соответствии с конкретным вариантом осуществления LUT является 3D LUT, декодированной с использованием октодерева, и с каждой вершиной сетки ассоциируются три значения, соседние вершины принадлежат родительскому октанту текущего октанта, которому принадлежит текущая вершина.

15 В соответствии с характерной особенностью изобретения предсказание по меньшей мере одного значения, ассоциированного с текущей вершиной, из восстановленных значений, ассоциированных с соседними вершинами, содержит интерполяцию по меньшей мере одного значения из соответствующих восстановленных значений соседних вершин.

В соответствии с характерной особенностью способ дополнительно содержит декодирование индекса из потока битов, указывающего тип интерполяции.

20 В соответствии с конкретным вариантом осуществления декодирование по меньшей мере одного остатка содержит энтропийное декодирование квантованного остатка из потока битов и обратное квантование квантованного остатка с помощью квантователя в декодированный остаток, и где способ дополнительно содержит декодирование квантователя из потока битов.

25 В соответствии с характерной особенностью изобретения для каждой вершины декодируется флаг, указывающий, кодируется ли по меньшей мере один остаток для вершины.

30 В соответствии с разновидностью для каждого значения текущей вершины декодируется флаг, указывающий, декодируется ли остаток для того значения, или подразумевается ли остаток равным нулю.

Преимущественно, что для текущего октанта декодируется флаг разделения, указывающий, декодируются ли его непосредственные потомки рекурсивно, или подразумеваются ли равными нулю все остатки вершин всех его еще не декодированных потомков.

35 Кодер для кодирования LUT, заданной в виде сетки вершин, где по меньшей мере одно значение ассоциируется с каждой вершиной сетки, содержащий, для текущей вершины:

- средство для предсказания по меньшей мере одного значения, ассоциированного с текущей вершиной, из другого значения; и

40 - средство для кодирования в поток битов по меньшей мере одного остатка, вычисленного между упомянутым по меньшей мере одним значением текущей вершины и его предсказанием в потоке битов.

Преимущественно, что другое значение получается из восстановленных значений, ассоциированных с соседними вершинами.

45 Кодер конфигурируется для исполнения этапов способа кодирования.

Декодер для декодирования LUT, заданной в виде сетки вершин, где по меньшей мере одно значение ассоциируется с каждой вершиной сетки, содержащий:

- средство для декодирования по меньшей мере одного остатка, ассоциированного

с текущей вершиной, из потока битов;

- средство для предсказания по меньшей мере одного значения, ассоциированного с текущей вершиной, из другого значения; и

- средство для восстановления по меньшей мере значения текущей вершины из его предсказания и по меньшей мере одного декодированного остатка.

Преимущественно, что другое значение получается из восстановленных значений, ассоциированных с соседними вершинами.

Кодер конфигурируется для исполнения этапов способа декодирования.

Раскрывается поток битов, кодирующий по меньшей мере LUT, заданную в виде сетки вершин. По меньшей мере одно значение ассоциируется с каждой вершиной сетки, и поток битов содержит по меньшей мере один кодированный в нем остаток, вычисленный между упомянутым по меньшей мере одним значением текущей вершины и его предсказанием.

#### 4. КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Другие флаги и преимущества изобретения проявятся вместе с нижеследующим описанием некоторых из его вариантов осуществления, причем это описание выполняется применительно к чертежам, на которых:

- фиг.1 иллюстрирует архитектуру масштабируемого видеодекодера, который использует масштабируемость гаммы в соответствии с известным уровнем техники;

- фиг.2 иллюстрирует архитектуру видеодекодера, который содержит преобразование цветов, чтобы приспособиться к характеристикам визуализации дисплея в соответствии с известным уровнем техники;

- фиг.3 схематически показывает квадратную 3D LUT, представленную в виде сетки с  $N \times N \times N$  вершинами, в соответствии с известным уровнем техники;

- фиг.4 изображает блок-схему алгоритма способа кодирования в соответствии с примерным вариантом осуществления изобретения;

- фиг.5 схематически показывает 2D LUT;

- фиг.6 представляет блок-схему алгоритма способа кодирования в соответствии с другим примерным вариантом осуществления изобретения;

фиг.7А и 7В схематически показывают рекурсивное подразделение куба на октанты (слева) и соответствующее октодереву (справа);

- фиг.8 иллюстрирует интерполяцию значений цвета вершины в соответствии с изобретением;

- фиг.9А схематически показывает 3D LUT, представленную в виде неравномерной сетки;

- фиг.9В иллюстрирует предсказание значения, ассоциированного с вершиной, в соответствии с конкретным и неограничивающим вариантом осуществления;

- фиг.9С представляет октант, в котором кодируются только 4 вершины из 8;

- фиг.9D представляет октанты, совместно использующие вершины;

- фиг.10 представляет блок-схему алгоритма способа декодирования в соответствии с примерным вариантом осуществления изобретения;

- фиг.11 представляет блок-схему алгоритма способа декодирования в соответствии с другим примерным вариантом осуществления изобретения;

- фиг.12 схематически показывает кодер для кодирования LUT в соответствии с изобретением;

- фиг.13 схематически показывает декодер для декодирования LUT в соответствии с изобретением; и

- фиг.14 изображает положение 8 вершин октанта.



## 5. ПОДРОБНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

Фиг.4 представляет блок-схему алгоритма способа кодирования в соответствии с примерным вариантом осуществления изобретения. Способ предназначен для кодирования LUT, заданной в виде сетки вершин, где по меньшей мере одно значение, например значение цвета, ассоциируется с каждой вершиной сетки. В отношении декодирования термины "восстановление" и "декодирование" очень часто используются как синонимы. В дальнейшем термин "значение цвета" содержит значения цвета в заданном цветовом пространстве, например значения RGB, YUV или Y,Cb,Cr, и дополнительно содержит значения, характеризующие преобразование цветов, например параметры CMF, то есть параметры матрицы и значения смещения.

На этапе 40 по меньшей мере одно значение цвета у текущей вершины предсказывается из восстановленных значений цвета, ассоциированных с соседними вершинами.

В качестве примера на фиг.5 изображается 2D LUT, ассоциирующая вершину  $V_0$  ( $c_{10}, c_{20}$ ) с соответствующей парой значений ( $V_{0c1}, V_{0c2}$ ), например значений цвета. Значения, ассоциированные с вершиной  $V_0$ , предсказываются например, из соседних в пространственном отношении вершин  $V_1(c_{11}, c_{21})$ ,  $V_2(c_{12}, c_{22})$ ,  $V_3(c_{13}, c_{23})$  и  $V_4(c_{14}, c_{24})$  с соответствующей парой значений  $(V_{ic1}, V_{ic2})_{i=1..4}$ . Предсказание  $P(Pc1, Pc2)$

вычисляется, например, с использованием интерполяции следующим образом:

$$Pc1 = 0,25 * (V_{1c1} + V_{2c1} + V_{3c1} + V_{4c1})$$

$$Pc2 = 0,25 * (V_{1c2} + V_{2c2} + V_{3c2} + V_{4c2})$$

В соответствии с разновидностью 2D LUT ассоциирует вершину  $V_0(c_{10}, c_{20})$  с набором параметров ( $m_{11}, m_{12}, m_{21}, m_{22}, o_1, o_2$ ) вместо пары значений ( $V_{0c1}, V_{0c2}$ ). Этот набор параметров может использоваться для восстановления значений ( $V_{0c1}, V_{0c2}$ ) из значений ( $c_{10}, c_{20}$ ) у  $V_0$  следующим образом:

$$V_{0c1} = m_{11} * c_{10} + m_{12} * c_{20} + o_1$$

$$V_{0c2} = m_{21} * c_{10} + m_{22} * c_{20} + o_2$$

Параметры, ассоциированные с вершиной  $V_0$ , предсказываются, например, из восстановленных параметров, ассоциированных с соседними в пространственном отношении вершинами  $V_1(c_{11}, c_{21})$ ,  $V_2(c_{12}, c_{22})$ ,  $V_3(c_{13}, c_{23})$  и  $V_4(c_{14}, c_{24})$ . Предсказание вычисляется, например, для параметра текущей вершины с использованием интерполяции.

На этапе 42 определяется по меньшей мере один остаток между, по меньшей мере, одним значением цвета у текущей вершины и его предсказанием и дополнительно кодируется в поток  $F$  битов. Остаток определяется путем вычитания предсказания по меньшей мере из одного значения цвета у текущей вершины. Кодирование обычно содержит энтропийное кодирование. В соответствии с разновидностью кодирование содержит квантование остатка с помощью квантователя  $q$  и энтропийное кодирование квантованного остатка.

В качестве примера первый и второй остатки вычисляются для вершины  $V_0$  в случае 2D LUT. Первый остаток равен  $(V_{0c1} - Pc1)$ , а второй остаток равен  $(V_{0c2} - Pc2)$ . Остатки, или квантованные остатки  $(V_{0c1} - Pc1)/q$  и  $(V_{0c2} - Pc2)/q$ , затем энтропийно кодируются в поток  $F$  битов. Энтропийное кодирование применяет традиционные методики двоичного кодирования, например экспоненциальное кодирование Голомба, метод Хаффмана, CABAC (английская аббревиатура для "контекстно-адаптивное двоичное

арифметическое кодирование").

Этапы 40 и 42 повторяются для кодирования дальнейшей вершины LUT до тех пор, пока не будут кодированы все вершины LUT.

При желании способ кодирования содержит кодирование значения  $q$  квантователя в поток  $F$  битов. По меньшей мере одно значение цвета у текущей вершины дополнительно восстанавливается для использования в предсказании других вершин. Каждое восстановленное значение цвета равно соответствующему исходному значению цвета, если не используется никакого квантователя, то есть  $q=1$ . В противном случае по меньшей мере одно восстановленное значение цвета определяется путем деквантования остатка и сложения деквантованного остатка с предсказанием.

В соответствии с разновидностью способ кодирования дополнительно содержит кодирование типа интерполяции в поток  $F$  битов, используемого для предсказания по меньшей мере одного значения цвета у вершины. Точнее говоря, из потока битов декодируется индекс, который идентифицирует тип интерполяции. В качестве примера индекс 0 идентифицирует билинейную интерполяцию, индекс 1 идентифицирует интерполяцию по ближайшим соседним элементам, индекс 2 идентифицирует линейную интерполяцию с 2 ближайшими соседними элементами.

В соответствии с разновидностью не все вершины в LUT кодируются в поток  $F$  битов. Например, если абсолютное значение всех остатков или всех квантованных остатков вершины меньше порогового значения  $TH$ , то для той вершины не кодируется никакого остатка, например  $TH=0$  или  $TH=1$ . Соответственно, в поток битов для каждой вершины кодируется двоичный флаг, указывающий, кодируется ли по меньшей мере один остаток для той вершины, или подразумеваются ли равными нулю остатки, которые не кодируются.

В соответствии с разновидностью для каждого значения цвета каждой вершины кодируется двоичный флаг, указывающий, кодируется ли остаток для того значения цвета, или является ли остаток не кодируемым и подразумеваемым равным нулю.

Размер LUT при желании также кодируется в поток битов.

Фиг.6 представляет блок-схему алгоритма способа кодирования в соответствии с другим примерным вариантом осуществления изобретения. Способ предназначен для кодирования 3D LUT, заданной в виде сетки вершин, с использованием октодерева, где набор из  $n$  значений цвета, например тройка значений цвета, ассоциируется с каждой вершиной сетки, где  $n$  – целое число  $\geq 1$ . Октодерево предназначено для разбиения цветового пространства 3D путем его рекурсивного подразделения на восемь октантов, как изображено на фиг.7А и 7В. На фиг.7А разбиение симметрично, тогда как на фиг.7В разбиение асимметрично. Октант уровня  $N$  имеет родительский октант на уровне  $N-1$ . Октодеревья являются трехмерным аналогом квадродеревьев. Способ кодирования раскрывается для кодирования текущей вершины  $V$  сетки, которая принадлежит текущему октанту. В дальнейшем  $n=3$ .

На этапе 50 каждое из трех значений цвета ( $V_r, V_g, V_b$ ), ассоциированное с текущей вершиной  $V$  с координатами  $(r, g, b)$ , предсказывается из восстановленных значений цвета, ассоциированных с соседними вершинами, то есть вершинами, которые принадлежат родительскому октанту у текущего октанта. Для упрощения записей используется  $(r, g, b)$  вместо  $(c_1, c_2, c_3)$ . Однако изобретение не ограничивается цветовым пространством  $(R, G, B)$ . Оно может применяться к представлениям цветовых пространств  $(Y, U, V)$ ,  $(Y, Cb, Cr)$  и т. п.

Соответственно, определяется предсказание для каждого значения цвета с использованием, например, трилинейной интерполяции, как проиллюстрировано с

помощью фиг.8:

$$V_r = K \times \sum_{i=0,1} \sum_{j=0,1} \sum_{k=0,1} s_i(r) \times s_j(g) \times s_k(b) \times LUT[r_i][g_j][b_k].r$$

где: (r<sub>i</sub>, g<sub>j</sub>, b<sub>k</sub>) при i=0, 1, j=0, 1 и k=0, 1 являются координатами вершин родительского октанта в цветовом пространстве 3D;

(r, g, b) являются координатами текущей вершины;

LUT[r<sub>i</sub>][g<sub>j</sub>][b<sub>k</sub>].r является первым значением цвета, ассоциированным с вершиной (r<sub>i</sub>, g<sub>j</sub>, b<sub>k</sub>);

$$K = \frac{1}{(r_1 - r_0) \times (g_1 - g_0) \times (b_1 - b_0)}$$

s<sub>0</sub>(t)=t<sub>1</sub>-t и s<sub>1</sub>(t)=t-t<sub>0</sub> при t=r, g или b.

Такие же уравнения используются для g и b. Можно использовать другой тип интерполяции, например призматическую, пирамидальную или тетраэдральную.

Сетка не обязательно является равномерной, как изображено на фиг.9А.

В разновидности этапа 50 каждое из трех значений цвета (V<sub>r</sub>, V<sub>g</sub>, V<sub>b</sub>), ассоциированное

с текущей вершиной V с координатами (r, g, b) в текущем октанте с индексом j, предсказывается по меньшей мере из одного восстановленного значения цвета,

ассоциированного с одной соседней вершиной V<sub>p</sub> с координатами (r<sub>p</sub>, g<sub>p</sub>, b<sub>p</sub>), то есть

вершиной, которая принадлежит соседнему октанту с индексом (j-1). Соседний октант является октантом, совместно использующим по меньшей мере одну вершину с текущим октантом и/или предшествующим текущему октанту в списке кодирования октантов.

Список кодирования октантов является списком, задающим порядок кодирования октанта. В конкретном варианте осуществления все октанты принадлежат одному и

тому же уровню. Соответственно, определяется предсказание для каждого значения цвета, как проиллюстрировано с помощью фиг.9В для компоненты r:  $\bar{V}_r = \text{PredA}_r + \text{PredB}_r$ ,

где PredA<sub>r</sub> является значением, которое зависит от положения вершины в 3D LUT, а PredB<sub>r</sub> является значением, которое зависит от восстановленного значения цвета,

ассоциированного с соседней вершиной V<sub>p</sub>. PredA<sub>r</sub>, например, равно r<sub>p</sub>. Для примера

$\text{PredB}_r = V'_r - \text{PredA}'_r$ , где V'<sub>r</sub> является восстановленным значением цвета вершины V<sub>p</sub> в соседнем октанте, а PredA'<sub>r</sub> является значением, которое зависит от положения вершины V<sub>p</sub> в 3D LUT. PredA'<sub>r</sub>, например, равно r<sub>p</sub>. Такие же уравнения используются для g и b.

На этапе 52 вычисляются три остатка для текущей вершины, по одному для каждой цветовой компоненты: res<sub>r</sub>=(V<sub>r</sub>- $\bar{V}_r$ ), res<sub>g</sub>=(V<sub>g</sub>- $\bar{V}_g$ ) и res<sub>b</sub>=(V<sub>b</sub>- $\bar{V}_b$ ).

Остатки затем энтропийно кодируются в поток битов или квантуются перед энтропийным кодированием. Энтропийное кодирование применяет традиционные методики двоичного кодирования, например экспоненциальное кодирование Голомба, метод Хаффмана, САВАС (английская аббревиатура для "контекстно-адаптивное двоичное арифметическое кодирование").

Октант уровня 0 не имеет родительского октанта. Точно так же первый октант в списке кодирования октантов не имеет предшествующего октанта. Следовательно, каждое из трех значений цвета (V<sub>r</sub>, V<sub>g</sub>, V<sub>b</sub>), ассоциированное с текущей вершиной V этого октанта, предсказывается из известного значения цвета, например значения 128. В соответствии с разновидностью используются разные известные значения для разных вершин октанта уровня 0. В соответствии с другой разновидностью предсказание из известных значений цвета также выполняется для других октантов помимо октанта

уровня 0. В соответствии с разновидностью каждое из трех значений цвета ( $V_r$ ,  $V_g$ ,  $V_b$ ), ассоциированное с текущей вершиной  $V$  этого октанта, предсказывается из уже восстановленных значений цвета, ассоциированных с соседними вершинами, которые принадлежат тому же октанту.

При желании способ кодирования содержит кодирование значения  $q$  квантователя в поток битов. Три значения цвета текущей вершины далее восстанавливаются и могут использоваться для предсказания других вершин. Каждое из восстановленных значений цвета равно исходному значению цвета, если не используется никакого квантователя ( $q=1$ ). В противном случае каждое восстановленное значение цвета определяется путем деквантования соответствующего остатка и сложения деквантованного остатка с соответствующим предсказанием.

В соответствии с разновидностью способ кодирования дополнительно содержит кодирование типа интерполяции в поток битов, используемого для предсказания по меньшей мере одного значения цвета у вершины. Точнее говоря, в поток битов кодируется индекс, который идентифицирует тип интерполяции. В качестве примера индекс 0 идентифицирует трилинейную интерполяцию, индекс 1 идентифицирует призматическую интерполяцию, индекс 2 идентифицирует пирамидальную интерполяцию, а индекс 3 идентифицирует тетраэдральную интерполяцию.

В соответствии с разновидностью не все вершины в LUT кодируются в поток битов. Например, если абсолютные значения всех остатков или всех квантованных остатков вершины меньше порогового значения  $T_H$ , то для той вершины не кодируется никакого остатка, например  $T_H=0$  или  $T_H=1$ . Соответственно, в поток битов для каждой вершины кодируется двоичный флаг, указывающий, кодируется ли по меньшей мере один остаток для той вершины. В соответствии с разновидностью для каждого значения цвета каждой вершины кодируется двоичный флаг, указывающий, кодируется ли остаток для того значения цвета, или является ли остаток не кодируемым и подразумеваемым равным нулю. В другой разновидности кодируются только определенные вершины, как изображено на фиг.9C. На этой фигуре кодируются только 4 вершины из 8 на октант.

LUT обычно кодируется в поток битов для использования в преобразовании изображений из видео в таких приложениях, как цветовая адаптация или масштабирование цветовой гаммы у визуализации дисплея. LUT можно кодировать и передавать вместе с видео. Если способу кодирования известно, что некоторые части 3D LUT не будут использоваться приложением, то не кодируются вершины, которые принадлежат этой части 3D LUT. Точно так же, если способу кодирования известно, что некоторые части 3D LUT обладают небольшим влиянием на окончательную визуализацию видео, то не кодируются вершины, которые принадлежат этой части 3D LUT.

Способ применяется рекурсивно для кодирования всей 3D LUT. В этом случае кодируются все октанты октодеревя. Как только кодированы все вершины текущего октанта, кодируются вершины дочернего октанта у текущего октанта.

В одной разновидности 3D LUT предварительно обрабатывается перед кодированием. В этом случае с каждым октантом в октодереве ассоциируется флаг разделения, и сначала устанавливается в ложь. Во время этапа предварительной обработки определяются значения флага разделения для каждого октанта. Если по меньшей мере одна кодируемая вершина текущего октанта содержит по меньшей мере один, по возможности квантованный, остаток больше  $T_H$ , то флаг разделения у его родительского октанта устанавливается в "истину". Таким образом, флаг разделения у текущего октанта уровня  $N$  указывает, кодируются ли рекурсивно его непосредственные потомки

(то есть потомки уровня N+1), или подразумеваются ли равными нулю все остатки вершин всех его еще не кодированных потомков (то есть потомков уровня N+k при k>0).

Во время этапа кодирования флаги разделения и остатки кодируются в поток битов.  
 5 Как только кодированы все вершины текущего октанта, кодируются вершины дочернего октанта у текущего октанта, если флагом разделения текущего октанта является истина. Вершины, которые принадлежат двум октантам, предпочтительно кодируются только один раз. В одной разновидности вершины, совместно используемые между несколькими октантами, кодируются более одного раза. В частности, вершины кодируются несколько  
 10 раз разными значениями, по одному значению для каждого октанта, которому она принадлежит. Со ссылкой на фиг.9D вершина V1 совместно используется между октантом (j) и октантом (j-1). Поэтому V1 можно кодировать дважды: одним значением для октанта (j), а другим значением – для октанта (j-1). Первое значение и другое значение могут отличаться. Точно так же вершину V2, которая совместно используется  
 15 между 4 октантами, можно кодировать 4 раза.

3D LUT кодируется, например, в VPS ("Набор параметров видео"), SPS ("Набор параметров последовательности"), PPS ("Набор параметров изображения") или в одном сообщении SEI ("Дополнительная информация улучшения"), которые заданы в стандартах кодирования видео AVC, HEVC, SVC или SHVC. 3D LUT кодируется,  
 20 например, в сообщении SEI, заданном ниже. Размер S у LUT при желании также кодируется в поток битов. S является количеством вершин в одном направлении.

В соответствии с разновидностью n 3D LUT, состоящих из вершин с 1 значением цвета, кодируются вместо кодирования одной 3D LUT, состоящей из вершин с n ассоциированными значениями цвета, например n=3, как раскрыто выше.

25 В соответствии с разновидностью, когда размер 3D LUT равен S=2, кодируется флаг, указывающий декодеру вычислить матрицу 3×3 коэффициентов усиления плюс смещение из декодированной 3D LUT и использовать ее в качестве CMF вместо 3D LUT.

В соответствии с другой разновидностью одна 3D LUT, состоящая из вершин с n ассоциированными значениями цвета, используется для кодирования локально заданных  
 30 параметров преобразования цветов. Для примера каждая вершина 3D LUT ассоциируется с 12 значениями цвета, характеризующими преобразование цветов. Вместо ассоциации 3 значений цвета ( $V_r, V_g, V_b$ ) с текущей вершиной V 12 значений цвета, характеризующих преобразование цветов, ассоциируются с вершиной V(r,g,b), где 12 значений цвета [ $a_i, b_i, c_i, o_i$ ] при i=0, 1 или 2 задаются так, чтобы подтверждалось следующее уравнение:

$$35 \quad \begin{pmatrix} Vr \\ Vg \\ Vb \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} + \begin{pmatrix} o_0 \\ o_1 \\ o_2 \end{pmatrix}$$

В соответствии с разновидностью только параметры [ $a_i, b_i, c_i$ ] при i=0, 1 или 2 ассоциируются с вершинами.

40 Конечно, в случае 3D LUT с размером 2 (один единственный октант с 8 вершинами) можно выбрать три значения цвета у вершин так, чтобы 3D LUT была эквивалента Модели усиления-смещения, заданной следующим уравнением:

$$45 \quad \begin{pmatrix} y' \\ u' \\ v' \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} y \\ u \\ v \end{pmatrix} + \begin{pmatrix} o_0 \\ o_1 \\ o_2 \end{pmatrix}$$

Для этого нужно установить 3 значения цвета ( $u_x, u_y, v_x$ ) $_{x=A,...,N}$  у 8 вершин с A по N в следующие значения:

$$\begin{array}{l}
 5 \quad \left\{ \begin{array}{l} y_A = 255.a_0 \\ y_B = 255.a_0 + o_0 \\ y_C = 255.b_0 + o_0 \\ y_D = 255.(a_0 + b_0) + o_0 \\ y_E = 255.c_0 + o_0 \\ y_F = 255.(a_0 + c_0) + o_0 \\ y_G = 255.(b_0 + c_0) + o_0 \\ y_H = 255.(a_0 + b_0 + c_0) + o_0 \end{array} \right. \quad \left\{ \begin{array}{l} u_A = 255.a_1 \\ u_B = 255.a_1 + o_1 \\ u_C = 255.b_1 + o_1 \\ u_D = 255.(a_1 + b_1) + o_1 \\ u_E = 255.c_1 + o_1 \\ u_F = 255.(a_1 + c_1) + o_1 \\ u_G = 255.(b_1 + c_1) + o_1 \\ u_H = 255.(a_1 + b_1 + c_1) + o_1 \end{array} \right.
 \end{array}$$

$$\begin{array}{l}
 10 \quad \left\{ \begin{array}{l} v_A = 255.a_2 \\ v_B = 255.a_2 + o_2 \\ v_C = 255.b_2 + o_2 \\ v_D = 255.(a_2 + b_2) + o_2 \\ v_E = 255.c_2 + o_2 \\ v_F = 255.(a_2 + c_2) + o_2 \\ v_G = 255.(b_2 + c_2) + o_2 \\ v_H = 255.(a_2 + b_2 + c_2) + o_2 \end{array} \right.
 \end{array}$$

15 К является постоянной, которая зависит от количества битов, используемых для представления каждого значения цвета. К=255, если значения цвета представляются на 8 битах, К=1023, если значения цвета представляются на 10 битах, и т.п. Тогда трилинейная интерполяция эквивалентна Модели усиления-смещения.

20 Фиг.10 представляет блок-схему алгоритма способа декодирования в соответствии с примерным вариантом осуществления изобретения.

На этапе 140 из потока F битов декодируется по меньшей мере один остаток. Декодирование обычно содержит энтропийное декодирование. В соответствии с разновидностью декодирование содержит энтропийное декодирование квантованного остатка и обратное квантование квантованного остатка с помощью квантователя q.

25 Энтропийное декодирование применяет традиционные методики двоичного декодирования, например экспоненциальное кодирование Голомба, метод Хаффмана, САВАС (английская аббревиатура для "контекстно-адаптивное двоичное арифметическое кодирование").

30 При желании способ декодирования содержит декодирование значения q квантователя из потока F битов.

В соответствии с разновидностью не все вершины в LUT кодируются в поток F битов. При желании из потока битов для каждой вершины декодируется двоичный флаг, указывающий, кодируется ли по меньшей мере один остаток для той вершины. Если не кодируется никакого остатка, то для этой вершины остаток (остатки) предполагается

35 (предполагаются) равным нулю.

На этапе 142 по меньшей мере одно значение цвета у текущей вершины предсказывается из восстановленных значений цвета, ассоциированных с соседними вершинами. В качестве примера на фиг.5 изображается 2D LUT, ассоциирующая вершину V0(c1,c2) с соответствующей парой значений цвета (V0c1, V0c2). Значения цвета, ассоциированные с вершиной V0, предсказываются из соседних в пространственном отношении вершин V1, V2, V3 и V4. В качестве примера предсказание P(Pc1, Pc2) вычисляется с использованием интерполяции следующим образом:

$$Pc1 = 0,25 * (V1_{c1} + V2_{c1} + V3_{c1} + V4_{c1})$$

$$Pc2 = 0,25 * (V1_{c2} + V2_{c2} + V3_{c2} + V4_{c2})$$

45 В соответствии с разновидностью 2D LUT ассоциирует вершину V0(c10,c20) с набором параметров (m11, m12, m21, m22, o1, o2) вместо пары значений (V0c1, V0c2). Этот набор параметров может использоваться для восстановления значений (V0c1, V0c2) из значений

( $c_{10}, c_{20}$ ) у  $V_0$  следующим образом:

$$V_{0c1} = m_{11} * c_{10} + m_{12} * c_{20} + o_1$$

$$V_{0c2} = m_{21} * c_{10} + m_{22} * c_{20} + o_2$$

5 Параметры, ассоциированные с вершиной  $V_0$ , предсказываются, например, из восстановленных параметров, ассоциированных с соседними в пространственном отношении вершинами  $V_1(c_{11}, c_{21})$ ,  $V_2(c_{12}, c_{22})$ ,  $V_3(c_{13}, c_{23})$  и  $V_4(c_{14}, c_{24})$ . Предсказание вычисляется, например, для параметра текущей вершины с использованием интерполяции.

10 На этапе 144 восстанавливается вершина. Точнее говоря, восстанавливается по меньшей мере одно значение цвета у текущей вершины из предсказания и по меньшей мере одного декодированного остатка.

В качестве примера в случае 2D LUT декодируются два остатка  $R_{c1}$  и  $R_{c2}$  для текущей вершины  $V_0$ . Таким образом, текущая вершина восстанавливается путем вычисления  
15 двух значений цвета следующим образом: ( $R_{c1} + Pc_1$ ) и ( $R_{c2} + Pc_2$ ).

В соответствии с разновидностью способ декодирования дополнительно содержит декодирование из потока  $F$  битов типа интерполяции, используемого для предсказания по меньшей мере одного значения цвета у вершины. Точнее говоря, из потока битов декодируется индекс, который идентифицирует тип интерполяции. В качестве примера  
20 индекс 0 идентифицирует билинейную интерполяцию, а индекс 1 идентифицирует интерполяцию по ближайшим вершинам.

Размер LUT при желании также декодируется из потока битов.

Фиг.11 представляет блок-схему алгоритма способа декодирования в соответствии с другим примерным вариантом осуществления изобретения. Способ предназначен  
25 для декодирования 3D LUT, заданной в виде сетки вершин, с использованием октодеревя, где с каждой вершиной сетки ассоциируется тройка значений цвета. Способ декодирования раскрывается для декодирования текущей вершины  $V$  сетки, которая принадлежит текущему октанту.

На этапе 150 из потока  $F$  битов декодируются три остатка  $res_r$ ,  $res_g$ ,  $res_b$ .  
30 Декодирование обычно содержит энтропийное декодирование. В соответствии с разновидностью декодирование содержит энтропийное декодирование квантованного остатка и обратное квантование квантованного остатка с помощью квантователя  $q$ . Энтропийное декодирование применяет традиционные методики двоичного декодирования, например экспоненциальное кодирование Голомба, метод Хаффмана,  
35 САВАС (английская аббревиатура для "контекстно-адаптивное двоичное арифметическое кодирование").

При желании способ декодирования содержит декодирование значения  $q$  квантователя из потока  $F$  битов.

40 В соответствии с разновидностью не все вершины в LUT кодируются в поток  $F$  битов. При желании из потока битов для каждой вершины декодируется двоичный флаг, указывающий, кодируется ли по меньшей мере один остаток для той вершины. Если не кодируется никакого остатка, то для этой вершины остаток (остатки) предполагается (предполагаются) равным нулю. В другой разновидности декодируются только определенные вершины, как изображено на фиг.9C. На этой фигуре декодируются  
45 только 4 вершины из 8 на октант.

На этапе 152 каждое из трех значений цвета ( $V_r$ ,  $V_g$ ,  $V_b$ ), ассоциированное с текущей вершиной  $V$  с координатами ( $r$ ,  $g$ ,  $b$ ), предсказывается из восстановленных значений

цвета, ассоциированных с соседними вершинами, то есть вершинами, которые принадлежат родительскому октанту у текущего октанта. Для упрощения записей используется (r, g, b) вместо (c1, c2, c3). Однако изобретение не ограничивается цветовым пространством (R, G, B). Оно может применяться к представлениям цветовых пространств (Y,U,V), (Y,Cb,Cr) и т. п.

Соответственно, предсказание определяется для каждого значения цвета.

Соответственно, определяется предсказание для каждого значения цвета с использованием, например, трилинейной интерполяции, как проиллюстрировано с помощью фиг.8:

$$V_r = K \times \sum_{i=0,1} \sum_{j=0,1} \sum_{k=0,1} s_i(r) \times s_j(g) \times s_k(b) \times LUT[r_i][g_j][b_k].r$$

где: (ri, gj, bk) при i=0, 1, j=0, 1 и k=0, 1 являются координатами вершин родительского октанта в цветовом пространстве 3D;

(r, g, b) являются координатами текущей вершины;

$LUT[r_i][g_j][b_k].r$  является первым значением цвета, ассоциированным с вершиной (ri, gj, bk);

$$K = \frac{1}{(r_1 - r_0) \times (g_1 - g_0) \times (b_1 - b_0)}$$

$s_0(t)=t_1-t$  и  $s_1(t)=t-t_0$  при  $t=r, g$  или  $b$ .

Такие же уравнения используются для g и b. Можно использовать другой тип интерполяции, например призматическую, пирамидальную или тетраэдральную. Сетка не обязательно является равномерной, как изображено на фиг.9А.

В разновидности этапа 152 каждое из трех значений цвета ( $V_r, V_g, V_b$ ),

ассоциированное с текущей вершиной V с координатами (r, g, b) в текущем октанте с индексом j, предсказывается по меньшей мере из одного восстановленного значения цвета, ассоциированного с одной соседней вершиной  $V_p$  с координатами ( $r_p, g_p, b_p$ ), то есть вершиной, которая принадлежит соседнему октанту с индексом (j-1). Соседний октант является октантом, совместно использующим по меньшей мере одну вершину с текущим октантом и/или предшествующим текущему октанту в списке кодирования октантов. Список кодирования октантов является списком, задающим порядок кодирования октанта. В конкретном варианте осуществления все октанты принадлежат одному и тому же уровню. Соответственно, определяется предсказание для каждого значения цвета, как проиллюстрировано с помощью фиг.9В для компоненты r:

$V_r = \text{PredA}_r + \text{PredB}_r$ , где  $\text{PredA}_r$  является значением, которое зависит от положения вершины в 3D LUT, а  $\text{PredB}_r$  является значением, которое зависит от восстановленного значения цвета, ассоциированного с соседней вершиной  $V_p$ .  $\text{PredA}_r$ , например, равно  $r_p$ . Для примера  $\text{PredB}_r = V'_r - \text{PredA}_r$ , где  $V'_r$  является восстановленным значением цвета вершины  $V_p$  в соседнем октанте, а  $\text{PredA}_r$  является значением, которое зависит от положения вершины  $V_p$  в 3D LUT.  $\text{PredA}_r$ , например, равно  $r_p$ . Такие же уравнения используются для g и b.

На этапе 154 вычисляются (соответственно, восстанавливаются) три значения цвета для текущей вершины из их предсказания и соответствующих декодированных остатков ( $\text{res}_r, \text{res}_g, \text{res}_b$ ). Таким образом, текущая вершина восстанавливается путем вычисления трех значений цвета следующим образом:  $(\text{res}_r + \bar{V}_r)$ ,  $(\text{res}_g + \bar{V}_g)$  и  $(\text{res}_b + \bar{V}_b)$ .

Октант уровня 0 не имеет родительского октанта. Точно так же первый октант в



списке кодирования октантов не имеет предшествующего октанта. Следовательно, каждое из трех значений цвета ( $V_r$ ,  $V_g$ ,  $V_b$ ), ассоциированное с текущей вершиной  $V$  этого октанта, предсказывается из известного значения цвета, например значения 128. В соответствии с разновидностью каждое из трех значений цвета ( $V_r$ ,  $V_g$ ,  $V_b$ ),

ассоциированное с текущей вершиной  $V$  этого октанта, предсказывается из уже восстановленных значений цвета, ассоциированных с соседними вершинами, которые принадлежат тому же октанту.

Способ применяется рекурсивно для декодирования всей 3D LUT. Вершины, которые принадлежат двум октантам, предпочтительно декодируются только один раз. В одной разновидности вершины, совместно используемые между несколькими окантами, декодируются более одного раза. В частности, вершины декодируются несколько раз разными значениями, по одному значению для каждого октанта, которому она принадлежит. Со ссылкой на фиг.9D вершина  $V1$  совместно используется между октантом ( $j$ ) и октантом ( $j-1$ ). Поэтому  $V1$  можно декодировать дважды: одним значением для октанта ( $j$ ), а другим значением – для октанта ( $j-1$ ). Первое значение и другое значение могут отличаться. Точно так же вершину  $V2$ , которая совместно используется между 4 окантами, можно декодировать 4 раза.

Как только декодированы все вершины текущего октанта (уровень  $N$ ), декодируются вершины дочерних (уровень  $N+1$ ) октантов у текущего октанта.

В соответствии с разновидностью декодируется флаг разделения для текущего октанта уровня  $N$ , который указывает, декодируются ли рекурсивно его непосредственные потомки (то есть потомки уровня  $N+1$ ), или подразумеваются ли равными нулю все остатки вершин всех еще не декодированных потомков (то есть потомков уровня  $N+k$  при  $k>0$ ).

В соответствии с разновидностью способ декодирования дополнительно содержит декодирование из потока  $F$  битов типа интерполяции, используемого для предсказания по меньшей мере одного значения цвета у вершины. Точнее говоря, из потока битов декодируется индекс, который идентифицирует тип интерполяции. В качестве примера индекс 0 идентифицирует трилинейную интерполяцию, индекс 1 идентифицирует призматическую интерполяцию, индекс 2 идентифицирует пирамидальную интерполяцию, а индекс 3 идентифицирует тетраэдральную интерполяцию.

3D LUT декодируется, например, из VPS, SPS, PPS или из одного сообщения SEI, которые заданы в стандартах кодирования видео AVC, HEVC, SVC или SHVC. Размер LUT при желании также декодируется из потока битов. 3D LUT декодируется, например, из сообщения SEI, заданного ниже.

В соответствии с разновидностью  $n$  3D LUT, состоящих из вершин с 1 значением цвета, декодируются вместо декодирования одной 3D LUT, состоящей из вершин с  $n$  значениями цвета, например  $n=3$ .

В соответствии с разновидностью, когда размер 3D LUT равен  $S=2$ , декодируется флаг, указывающий декодеру вычислить матрицу  $3 \times 3$  коэффициентов усиления плюс смещение из декодированной 3D LUT и использовать ее в качестве CMF вместо 3D LUT.

Такие же варианты, раскрытые по отношению к способу кодирования в соответствии с фиг.4 и 6, могут применяться на стороне декодера.

На фиг.4, 6, 10 и 11 представленные прямоугольники являются исключительно функциональными объектами, которые не обязательно соответствуют физическим отдельным объектам. Как поймет специалист в данной области техники, аспекты настоящих принципов можно воплотить в виде системы, способа либо машиночитаемого носителя. Соответственно, аспекты настоящих принципов могут принимать вид

полностью аппаратного варианта осуществления, полностью программного варианта осуществления (включая микропрограммное обеспечение, резидентное программное обеспечение, микрокод и так далее) или варианта осуществления, объединяющего программные и аппаратные аспекты, которые в этом документе можно в целом называть "схемой", "модулем" или "системой". Кроме того, аспекты настоящих принципов могут принимать вид машиночитаемого носителя информации. Может использоваться любое сочетание одного или нескольких машиночитаемых носителей информации.

Блок-схема алгоритма и/или блок-схемы на фигурах иллюстрируют конфигурацию, работу и функциональные возможности у возможных реализаций систем, способов и компьютерных программных продуктов в соответствии с различными вариантами осуществления настоящего изобретения. В этой связи каждый блок на блок-схеме алгоритма или блок-схемах может представлять собой модуль, сегмент или часть кода, который (которая) содержит одну или несколько исполняемых команд для реализации заданной логической функции (функций). Также следует отметить, что в некоторых альтернативных реализациях функции, указанные в блоке, могут совершаться не в том порядке, указанном на фигурах. Например, показанные последовательно два блока фактически могут исполняться практически одновременно, или блоки иногда могут исполняться в обратном порядке, или блоки могут исполняться в альтернативном порядке в зависимости от заложенных функциональных возможностей. Также отметим, что каждый блок из блок-схем и/или блок-схемы алгоритма и сочетания блоков на блок-схемах и/или блок-схеме алгоритма можно реализовать с помощью специализированных аппаратных систем, которые выполняют заданные функции и/или действия, либо сочетаний специализированных аппаратных средств и компьютерных команд. Хотя и не описано явно, настоящие варианты осуществления могут применяться в любой комбинации или подкомбинации.

Также раскрывается поток битов, который кодирует LUT, например 3D LUT. Поток битов, созданный по способу кодирования из фигуры, кодирующий по меньшей мере LUT, заданную в виде сетки вершин, где по меньшей мере одно значение цвета ассоциируется с каждой вершиной сетки, где упомянутый поток битов содержит кодированный в нем по меньшей мере один остаток, вычисленный между упомянутым по меньшей мере одним значением цвета текущей вершины и его предсказанием.

Примерный вариант осуществления предлагается в инфраструктуре стандарта кодирования HEVC, заданного в документе JCTVC-L1003 от Объединенной команды по кодированию видео (JCT-VC) из SG16 WP3 ITU-T и JTC1/SC29/WG11 ISO/IEC, или стандарта кодирования SHVC, который является масштабируемым расширением стандарта кодирования HEVC, заданным в документе JCTVC-L1008 от Объединенной команды по кодированию видео (JCT-VC) из SG16 WP3 ITU-T и JTC1/SC29/WG11 ISO/IEC. Стандарт задает синтаксис, которому обязан соответствовать любой поток кодированных данных, чтобы быть совместимым с этим стандартом. Синтаксис задает, в частности, то, как кодируются различные элементы информации (например, данные, относящиеся к изображениям, включенным в последовательность, векторам движения и т. п.). Применительно к стандарту кодирования SHVC LUT можно кодировать в PPS или в VPS. Синтаксический элемент `use_color_prediction` используется для указания использования предсказания цвета в текущем изображении, как показано в Таблице 1.

<code>use_color_prediction</code>
<code>if(use_color_prediction)</code>
<code>3D_LUT_color_data ( )</code>
<code>rbps_trailing_bits( )</code>

Таблица 1  
сигнализация параметров предсказания

Если флаг `use_color_prediction` равен "1", то вызывается функция `3D_LUT_color_data`, чтобы сигнализировать данные 3D LUT, которые показаны в Таблице 2.

3D_LUT_color_data ( ) {	Дескриптор
nbpCode	u(3)
coding_octant(0, 0,0,0)	
}	
Таблица 2 кодирование данных о цвете 3D LUT	

`nbpCode` указывает размер 3D LUT, который перечислен в Таблице 4 для заданного значения `nbpCode`. Значение квантователя может кодироваться с помощью функции `3D_LUT_color_data()`.

В соответствии с разновидностью `3D_LUT_color_data ( )` задается в Таблице 3 следующим образом.

3D_LUT_color_data ( ) {	Дескриптор
nbpCode	u(3)
NbitsPerSample	u(5)
coding_octant(0, 0,0,0)	
}	
Таблица 3 кодирование данных о цвете 3D LUT	

`nbpCode` указывает размер 3D LUT, который перечислен в Таблице 4 для заданного значения `nbpCode`. Значение квантователя может кодироваться с помощью функции `3D_LUT_color_data()`.

`NbitsPerSample` указывает количество битов, используемое для представления значений цвета.

nbpCode	размер 3D LUT
0	2
1	3
2	5
3	9
4	17
5	33
Таблица 4 интерпретация nbpCode	

Декодирование октанта (слой, `y,u,v`) является рекурсивной функцией, как показано в Таблице 4. Каждый октант состоит из 8 вершин, ассоциированных с флагом (`encoded_flag[i]`), указывающим, кодируются ли остаточные значения цвета, или все они подразумеваются равными нулю. Значения цвета восстанавливаются путем сложения остатков с предсказанием значений цвета. Предсказание значений цвета вычисляется с использованием, например, трилинейной интерполяции 8 соседних вершин с `layer_id-1`.

coding_octant ( layer_id, y,u,v) {	Дескриптор
for( i=0; i < 8 ; i++ ) {	
encoded_flag[i]	u(1)
if ( encoded_flag[i] ) {	
resY[j]	ue(v)
resU[j]	ue(v)

	resV[j]	ue(v)
	}	
	}	
	split_flag	u(1)
5	if ( split_flag ) {	
	for( i=0; i < 8 ; i++ ) {	
	coding_octant ( layer_id+1, y+dy[i],u+du[i],v+dv[i])	
	}	
	}	
	}	
10	Таблица 5 синтаксические элементы для coding_octant()	

В соответствии с другим преимущественным вариантом осуществления LUT кодируется в сообщении SEI (SEI означает "дополнительную информацию улучшения"). Стандарт HEVC в Приложении D задает способ, которым кодируется дополнительная информация, называемая SEI. На эту дополнительную информацию ссылаются с помощью поля payloadType в синтаксисе. Сообщения SEI участвуют, например, в связанных с отображением процессах. Отметим, что если устройство декодирования не обладает функциональными возможностями, необходимыми для ее использования, то эта информация игнорируется. В соответствии с конкретным вариантом осуществления изобретения задается новый тип сообщения SEI, чтобы кодировать дополнительную информацию, относящуюся к 3D LUT. С этой целью новое значение для поля payloadType задается из числа еще не используемых значений (например, payloadType равно 24).

Синтаксис данных SEI (то есть sei\_payload) расширяется следующим образом:

25	color_mapping_info( payloadSize ) {	Дескриптор
	color_map_id	ue(v)
	color_map_cancel_flag	u(1)
	if( !color_map_cancel_flag ) {	
	color_map_repetition_period	ue(v)
	color_description_present_flag	u(1)
30	If (colour_description_present_flag) {	
	color_primaries_input_id	u(8)
	color_primaries_output_id	u(8)
	}	
	color_output_rgb	u(1)
35	lut_bit_depth_minus8	u(4)
	3D_LUT_color_data ( )	
	}	
	}	
	Таблица 6 сообщение SEI отображения цветов	

В соответствии с разновидностью сообщение SEI дополнительно содержит индикатор color\_interpolator\_id, например, после color\_description\_present\_flag, чье значение указывает тип интерполяции, который задан в Таблице 7.

	color_interpolator_id	размер 3D LUT
	0	Трилинейный
45	1	Тетраэдральный
	Таблица 7 тип интерполяции	

Это сообщение SEI предоставляет информацию для разрешения повторного

отображения выборок цвета у выходных декодированных изображений для адаптации к конкретным средам отображения. Процесс повторного отображения отображает кодированные выборочные значения в цветовом пространстве RGB в целевые выборочные значения. Отображения выражаются либо в области яркости, либо области

5 цветового пространства RGB и должны применяться соответственно к компоненте яркости либо к каждой компоненте RGB, порожденной преобразованием цветового пространства у декодированного изображения.

3D\_LUT\_color\_data ( ) задается в Таблице 2 или 3.

Декодированная 3D LUT применяется к декодированным изображениям,

10 принадлежащим слою, идентифицированному, например, индексом nuh\_layer\_id в Заголовке блока NAL (см. раздел 7.3.1.2 стандарта кодирования HEVC, заданного в документе JCTVC-L1003 от Объединенной команды по кодированию видео (JCT-VC) из SG16 WP3 ITU-T и JTC1/SC29/WG11 ISO/IEC) в сообщении SEI.

color\_map\_id содержит идентифицирующий номер, который может использоваться

15 для идентификации предназначения модели отображения цветов. Значения color\_map\_id могут использоваться так, как определено приложением. color\_map\_id может использоваться для поддержки операций отображения цветов, которые подходят для разных сценариев отображения. Например, разные значения color\_map\_id могут соответствовать разным битовым глубинам отображения.

20 color\_map\_cancel\_flag, равный 1, указывает, что сообщение SEI с информацией отображения цветов отменяет наличие любого предыдущего сообщения SEI с информацией отображения цветов в порядке вывода. color\_map\_cancel\_flag, равный 0, указывает, что информация отображения цветов идет далее.

color\_map\_repetition\_period задает наличие сообщения SEI с информацией отображения

25 цветов и может задавать интервал счетчика последовательности изображений, в пределах которого в потоке битов должно присутствовать другое сообщение SEI с информацией отображения цветов с таким же значением color\_map\_id либо конец кодированной видеопоследовательности. color\_map\_repetition\_period, равный 0, задает, что информация отображения цветов применяется только к текущему декодированному изображению.

30 color\_map\_repetition\_period, равный 1, задает, что информация отображения цветов присутствует в порядке вывода до тех пор, пока выполняется любое из следующих условий:

- Начинается новая кодированная видеопоследовательность.

- Выводится изображение в блоке доступа, содержащем сообщение SEI с информацией

35 отображения цветов с таким же значением color\_map\_id, имеющее счетчик последовательности изображений (известный как POC) больше, чем POC текущего декодированного изображения, обозначенный PicOrderCnt( CurrPic ).

color\_map\_repetition\_period, равный 0 или равный 1, указывает, что может присутствовать или не присутствовать другое сообщение SEI с информацией

40 отображения цветов с таким же значением color\_map\_id.

color\_map\_repetition\_period больше 1 задает, что информация отображения цветов присутствует до тех пор, пока выполняется любое из следующих условий:

- Начинается новая кодированная видеопоследовательность.

- Выводится изображение в блоке доступа, содержащем сообщение SEI с информацией

45 отображения цветов с таким же значением color\_map\_id, имеющее POC больше PicOrderCnt( CurrPic ) и меньше либо равный PicOrderCnt( CurrPic )+ color\_map\_repetition\_period.

color\_map\_repetition\_period больше 1 указывает, что должно присутствовать другое

сообщение SEI с информацией отображения цветов с таким же значением `color_map_id` для изображения в блоке доступа, который выводится, имеющее РОС больше `PicOrderCnt( CurrPic )` и меньше либо равный `PicOrderCnt( CurrPic )`+

`color_map_repetition_period`; пока не закончится поток битов или не начнется новая кодированная видеопоследовательность, без вывода такого изображения.

`color_description_present_flag`, равный 1, задает, что присутствуют `colour_primaries_input_id` и `colour_primaries_output_id`. `colour_description_present_flag`, равный 0, задает, что `colour_primaries_input_id` и `colour_primaries_output_id` отсутствуют.

`color_primaries_input_id` указывает координаты цветности у основных цветов источника, как задано в Таблице 8 на основе определения CIE 1931хи у, которое задано ISO 11664-1.

`color_primaries_output_id` указывает координаты цветности у отображенных основных цветов, как задано в Таблице 8 на основе определения CIE 1931хи у, которое задано ISO 11664-1, как только применяется цветовая 3D lut.

`color_output_rgb`, равный 1, задает, что выходные выборки цвета являются сигналами яркости и цветности. `color_output_rgb`, равный 0, задает, что выходные выборки цвета являются значениями зеленого, красного, синего.

`lut_bit_depth_minus8` задает битовую глубину выборок 3D LUT.

`nbr_code` указывает размер `nbr` 3D LUT, который перечислен в Таблице 4 для заданного значения `nbr_code`.

Результатом декодирования 3D LUT является 3-мерный массив LUT размера `nbr×nbr×nbr`. Каждый элемент массива LUT называется вершиной и ассоциируется с 3 восстановленными выборочными значениями (`recSamplesY`, `recSamplesU`, `recSamplesV`) с битовой глубиной, равной (`lut_bit_depth_minus8+8`). Говорят, что вершина `lut[i][j][k]` принадлежит слою `layer_id`, если значения `i%(nbr>>layer_id)`, `j%(nbr>>layer_id)`, `k%(nbr>>layer_id)` равны нулю. Одна вершина может принадлежать нескольким слоям. Октант слоя `layer_id` состоит из 8 соседних вершин, принадлежащих `layer_id` (фиг.14).

Декодирование октанта (`layer_id`, `u`, `u`, `v`) является рекурсивной функцией. Каждый октант состоит из 8 вершин (`i=0, ... 7`), ассоциированных с флагом (`encoded_flag[i]`), указывающим, кодируются ли значения остаточных компонент (`resY[i]`, `resU[i]`, `resV[i]`), или все они подразумеваются равными нулю. Значения компонент восстанавливаются путем сложения остатков с предсказанием значений компонент. Предсказание значений компонент вычисляется с использованием трилинейной интерполяции 8 соседних вершин с `layer_id-1`. Как только вершина восстановлена, она помечается как восстановленная.

Где (`y+dy[i]`), (`u+du[i]`) и (`v+dv[i]`) являются координатами 8 дочерних октантов (координатами первой вершины цвета 3D) для текущего октанта (имеющего (`u`, `u`, `v`) в качестве координат первой вершины (`i=0`)). Значения `dy[i]`, `du[i]` и `dv[i]` для данного слоя изображаются в Таблице 9.

Значение	Основные цвета			Полезное замечание
0	Зарезервировано			Для будущего использования ITU-T/ISO/IEC
1	основной цвет	x	y	ITU-R Rec. BT.709-5
	зеленый	0.300	0.600	ITU-R Rec. BT.1361 традиционная система
	синий	0.150	0.060	цветовой гаммы и расширенная система
	красный	0.640	0.330	цветовой гаммы
	белый D65	0.3127	0.3290	IEC 61966-2-1 (sRGB либо sYCC)
				IEC 61966-2-4
				RP 177 (1993) Общества инженеров кино
				и телевидения, Приложение В
2	Не задано			Характеристики изображения неизвестны
				либо определяются приложением
3	Зарезервировано			Для будущего использования ITU-T/ISO/IEC
4	основной цвет	x	y	Система M ITU R Rec. BT.470 6 (исторически установленный)
	зеленый	0.21	0.71	Рекомендация 1953 г. Национального комитета по
	синий	0.14	0.08	телевизионным стандартам США для стандартов передачи
	красный	0.67	0.33	для цветного телевидения
	белый C	0.310	0.316	Федеральная комиссия по связи США,
				Раздел 47 Свода федеральных
				нормативных актов (2003) 73.682 (a) (20)
5	основной цвет	x	y	Система B, G ITU R Rec. BT.470 6
	green	0.29	0.60	(исторически установленный)
	синий	0.15	0.06	ITU-R Rec. BT.601-6 625
	красный	0.64	0.33	ITU-R Rec. BT.1358 625
	белый D65	0.3127	0.3290	ITU-R Rec. BT.1700 625 PAL и 625
				SECAM
6	основной цвет	x	y	ITU-R Rec. BT.601-6 525
	зеленый	0.310	0.595	ITU-R Rec. BT.1358 525
	синий	0.155	0.070	ITU-R Rec. BT.1700 NTSC
	красный	0.630	0.340	Общество инженеров кино и телевидения 170M (2004)
	белый D65	0.3127	0.3290	(функционально такое же, как значение 7)
7	основной цвет	x	y	Общество инженеров кино и телевидения
	зеленый	0.310	0.595	240M (1999)
	синий	0.155	0.070	(функционально такое же, как значение 6)
	красный	0.630	0.340	
	белый D65	0.3127	0.3290	
8	основной цвет	x	y	Типовой фильм (светофильтры, использующие
	зеленый	0.243	0.692	источник C света)
	(Wratten 58)			
	синий	0.145	0.049	
	(Wratten 47)			
	red	0.681	0.319	
	(Wratten 25)			
	белый C	0.310	0.316	
9	основной цвет	x	y	Rec. ITU-R BT.2020
	зеленый	0.170	0.797	
	синий	0.131	0.046	
	красный	0.708	0.292	
	белый D65	0.3127	0.3290	
10..255	Зарезервировано			Для будущего использования ITU-T/ISO/IEC

Таблица 8

Основные цвета (color\_primaries\_input\_id и color\_primaries\_output\_id)

i	dy[i]	du[i]	dv[i]
0	0	0	0
1	0	0	nbp >> layer_id
2	0	nbp >> layer_id	0
3	0	nbp >> layer_id	nbp >> layer_id

4	$\text{nbp} \gg \text{layer\_id}$	0	0
5	$\text{nbp} \gg \text{layer\_id}$	0	$\text{nbp} \gg \text{layer\_id}$
6	$\text{nbp} \gg \text{layer\_id}$	$\text{nbp} \gg \text{layer\_id}$	0
7	$\text{nbp} \gg \text{layer\_id}$	$\text{nbp} \gg \text{layer\_id}$	$\text{nbp} \gg \text{layer\_id}$
Таблица 9 значения $\text{dy}[i]$ , $\text{du}[i]$ и $\text{dv}[i]$ в функции индекса $i$ для вершин, принадлежащих слою= $\text{layer\_id}$ .			

Восстановленные выборки цветовой 3D LUT ( $\text{recSamplesY}[i]$ ,  $\text{recSamplesU}[i]$ ,  $\text{recSamplesV}[i]$ ) для вершины (  $(y+\text{dy}[i])$ ,  $(u+\text{du}[i])$ ,  $(v+\text{dv}[i])$  ), принадлежащей октанту слоя= $\text{layer\_id}$ , имеют вид:

$$\text{recSamplesY}[i] = \text{resY}[i] + \text{predSamplesY}[i]$$

где значение  $\text{predSampleY}[i]$  выводится с использованием трилинейной интерполяции с вершинами в октанте слоя= $\text{layer\_id}-1$ , который содержит текущий октант.

Фиг.12 представляет примерную архитектуру кодера 1. Кодер конфигурируется для исполнения этапов способа кодирования. Кодер 1 содержит следующие элементы, которые соединяются шиной 64 данных и адреса:

- микропроцессор 61 (или CPU), который является, например, DSP (или цифровым процессором сигналов);
- ROM 62 (или постоянное запоминающее устройство);
- RAM 63 (или оперативное запоминающее устройство);
- одно или несколько устройств 65 I/O (ввод/вывод), например клавиатуру, мышь, веб-камеру; и
- источник 66 питания.

В соответствии с разновидностью источник 66 питания является внешним по отношению к кодеру. Каждый из этих элементов на фиг.12 известен специалистам в данной области техники и дополнительно раскрываться не будет. В каждом упомянутом запоминающем устройстве слово "регистр", используемое в спецификации, обозначает в каждом из упомянутых запоминающих устройств зону памяти небольшой емкости (некоторые двоичный данные), а также зону памяти большой емкости (позволяющую хранить всю программу либо все или часть данных, представляющих вычисленные данные или данные для отображения). ROM 62 содержит программу и параметры кодирования (например, пороговую величину TH). Алгоритм способа кодирования в соответствии с изобретением хранится в ROM 62. При включении CPU 61 загружает программу 620 в RAM и исполняет соответствующие команды.

RAM 63 содержит в регистре программу, исполняемую CPU 61 и загружаемую после включения кодера 1, входные данные в регистр, кодированные данные в разном состоянии способа кодирования в регистре и другие переменные, используемые в регистре для кодирования.

Фиг.13 представляет примерную архитектуру декодера 2. Декодер конфигурируется для исполнения этапов способа декодирования. Декодер 2 содержит следующие элементы, которые соединяются шиной 74 данных и адреса:

- микропроцессор 71 (или CPU), который является, например, DSP (или цифровым процессором сигналов);
- ROM 72 (или постоянное запоминающее устройство);
- RAM 73 (или оперативное запоминающее устройство);
- интерфейс 75 I/O для приема от приложения данных для передачи; и
- батарею 76.

В соответствии с разновидностью батарея 76 является внешней по отношению к декодеру. Каждый из этих элементов на фиг.13 известен специалистам в данной области техники и дополнительно раскрываться не будет. В каждом упомянутом запоминающем



устройстве слово "регистр", используемое в спецификации, может соответствовать области небольшой емкости (несколько бит) или очень большой области (например, вся программа либо большой объем принятых или декодированных данных). ROM 72 содержит по меньшей мере программу и параметры декодера. Алгоритм способа декодирования в соответствии с изобретением хранится в ROM 72. При включении CPU 71 загружает программу 720 в RAM и исполняет соответствующие команды.

RAM 73 содержит в регистре программу, исполняемую CPU 71 и загружаемую после включения декодера 2, входные данные в регистр, декодированные данные в разном состоянии способа декодирования в регистре и другие переменные, используемые в регистре для декодирования.

Описанные в этом документе реализации можно осуществить, например, в способе или процессе, устройстве, компьютерной программе, потоке данных или сигнале. Даже если реализация обсуждается применительно лишь к одному виду реализации (например, обсуждается только как способ или устройство), реализацию обсуждаемых флагов также можно осуществить в других видах (например, в программе). Устройство можно реализовать, например, в подходящих аппаратных средствах, программном обеспечении и микропрограммном обеспечении. Способы можно реализовать, например, в устройстве, например в процессоре, который относится в общем к устройствам обработки, включая, например, компьютер, микропроцессор, интегральную схему или программируемое логическое устройство. Процессоры также включают в себя устройства связи, например компьютеры, сотовые телефоны, портативные/персональные цифровые помощники ("PDA") и другие устройства, которые упрощают передачу информации между конечными пользователями.

Описанные в этом документе реализации различных процессов и флагов можно осуществить в разном оборудовании или приложениях. Примеры такого оборудования включают в себя кодер, декодер, постпроцессор, обрабатывающий вывод из декодера, препроцессор, предоставляющий ввод в кодер, видеокодер, видеodeкодер, видеокодек, веб-сервер, телевизионную приставку, переносной компьютер, персональный компьютер, сотовый телефон, PDA и другие устройства связи. Как должно быть понятно, оборудование может быть мобильным и даже устанавливаться в мобильном транспортном средстве.

Более того, способы можно реализовать с помощью команд, выполняемых процессором, и такие команды (и/или значения данных, созданные реализацией) можно хранить на читаемом процессором носителе, например интегральной схеме, носителе программного обеспечения или другом запоминающем устройстве, например на жестком диске, компакт-диске ("CD"), оптическом диске (например, DVD, часто называемом цифровым универсальным диском или цифровым видеодиском), оперативном запоминающем устройстве ("RAM") или постоянном запоминающем устройстве ("ROM"). Команды могут образовывать прикладную программу, материально воплощенную на читаемом процессором носителе. Команды могут входить в состав, например, аппаратных средств, микропрограммного обеспечения, программного обеспечения или их сочетания. Команды могут входить в состав, например, операционной системы, отдельного приложения или сочетания этих двух вариантов. Поэтому процессор можно описать, например, как устройство, сконфигурированное для осуществления процесса, и как устройство, которое включает в себя читаемый процессором носитель (например, запоминающее устройство), содержащий команды для осуществления процесса. Кроме того, читаемый процессором носитель может хранить созданные реализацией значения данных в дополнение к командам или вместо них.

Как станет очевидно специалисту в данной области техники, реализации могут порождать ряд сигналов, оформленных для переноса информации, которую можно, например, хранить или передавать. Информация может включать в себя, например, команды для выполнения способа или данные, созданные одной из описанных реализаций. Например, сигнал можно оформить для переноса, в качестве данных, правил для записи или считывания синтаксиса из описанного варианта осуществления или для переноса, в качестве данных, фактических значений синтаксиса, записанных с помощью описанного варианта осуществления. Такой сигнал можно оформить, например, в виде электромагнитной волны (например, с использованием радиочастотной части спектра) или в виде основополосного сигнала. Оформление может включать в себя, например, кодирование потока данных и модулирование несущей с кодированным потоком данных. Информация, которую переносит сигнал, может быть, например, аналоговой или цифровой информацией. Сигнал, как известно, можно передавать по разным проводным или беспроводным линиям связи. Сигнал можно сохранить на читаемом процессором носителе.

Описано некоторое количество реализаций. Тем не менее, станет понятно, что можно вносить различные модификации. Например, элементы разных реализаций можно объединять, дополнять, изменять или удалять для создания других реализаций. Более того, средний специалист поймет, что раскрытые структуры и процессы можно заменить другими, и результирующие реализации будут выполнять в основном такую же функцию (функции) в основном таким же способом (способами), чтобы достичь в основном такого же результата (результатов), как и раскрытые реализации. Соответственно, данной заявкой рассматриваются эти и другие реализации. В частности, способ кодирования LUT в соответствии с изобретением можно использовать для кодирования любого вида LUT независимо от типа значений, ассоциированных с каждой вершиной.

#### (57) Формула изобретения

1. Способ кодирования цветовой 3-мерной справочной таблицы, ассоциированной с октантами октодерева, содержащий этапы, на которых:
  - 30 кодируют флаг разделения для текущего октанта, указывающий, кодируется ли рекурсивно по меньшей мере один из его непосредственных дочерних октантов; и
  - кодируют упомянутый текущий октант в ответ на упомянутый флаг разделения.
2. Способ по п.1, в котором этап, на котором кодируют упомянутый текущий октант, содержит этапы, на которых предсказывают по меньшей мере одно значение вершины
  - 35 в упомянутом текущем октанте из другого значения, чтобы получить предсказание; и
  - кодируют остаток, вычисленный между упомянутым по меньшей мере одним значением вершины и упомянутым предсказанием.
3. Способ по п.2, дополнительно содержащий этап, на котором кодируют флаг для упомянутой вершины, указывающий, кодируется ли или нет по меньшей мере один
  - 40 остаток для упомянутой вершины.
4. Способ по п.2, дополнительно содержащий этап, на котором кодируют флаг для каждого значения упомянутой вершины, указывающий, кодируется ли остаток для упомянутого значения или является ли остаток некодированным и подразумеваемым равным нулю.
- 45 5. Способ по п.2, в котором упомянутое другое значение получается по меньшей мере из одного восстановленного значения соседней вершины.
6. Способ по п.5, в котором упомянутое по меньшей мере одно значение упомянутой вершины, упомянутое другое значение и упомянутое по меньшей мере одно

восстановленное значение упомянутой соседней вершины являются значениями цвета.

7. Способ по п.6, в котором упомянутые значения цвета соответствуют преобразованию цветов.

8. Способ по любому из пп.5–7, в котором упомянутое по меньшей мере одно  
5 восстановленное значение соседней вершины принадлежит родительскому октанту текущего октанта.

9. Способ декодирования цветовой 3-мерной справочной таблицы октантов октодеревя, содержащий этапы, на которых:

10 декодируют флаг разделения для текущего октанта, указывающий, декодируется ли  
рекурсивно по меньшей мере один из его непосредственных дочерних октантов; и  
восстанавливают упомянутый текущий октант в ответ на упомянутый флаг  
разделения.

10. Способ по п.9, в котором этап, на котором восстанавливают упомянутый текущий  
15 октант, содержит этапы, на которых предсказывают по меньшей мере одно значение вершины в упомянутом текущем октанте из другого значения, чтобы получить предсказание; и

восстанавливают упомянутое по меньшей мере одно значение вершины из упомянутого предсказания и декодированного остатка.

11. Способ по п.10, дополнительно содержащий этап, на котором декодируют флаг  
20 для упомянутой вершины, указывающий, декодируется ли или нет по меньшей мере один остаток для упомянутой вершины.

12. Способ по п.10, дополнительно содержащий этап, на котором декодируют флаг  
для каждого значения упомянутой вершины, указывающий, декодируется ли остаток  
для упомянутого значения или является ли остаток недекодируемым и подразумеваемым  
25 равным нулю.

13. Способ по п.10, в котором упомянутое другое значение получается по меньшей мере из одного восстановленного значения соседней вершины.

14. Способ по п.13, в котором упомянутое по меньшей мере одно значение  
упомянутой вершины, упомянутое другое значение и упомянутое по меньшей мере  
30 одно восстановленное значение упомянутой соседней вершины являются значениями цвета.

15. Способ по п.14, в котором упомянутые значения цвета соответствуют преобразованию цветов.

16. Способ по любому из пп.13–15, в котором упомянутое по меньшей мере одно  
35 восстановленное значение упомянутой соседней вершины принадлежит родительскому октанту текущего октанта.

17. Кодер для кодирования цветовой 3-мерной справочной таблицы октантов, содержащий интерфейс для обращения к упомянутой цветовой 3-мерной справочной  
таблице октантов и по меньшей мере процессор, сконфигурированный для:

40 кодирования флага разделения для текущего октанта, указывающего, кодируется ли рекурсивно по меньшей мере один из его непосредственных дочерних октантов; и кодирования упомянутого текущего октанта в ответ на упомянутый флаг разделения.

18. Кодер по п.17, в котором кодирование упомянутого текущего октанта содержит предсказание по меньшей мере одного значения вершины в упомянутом текущем  
45 октанте из другого значения, чтобы получить предсказание; и

кодирование остатка, вычисленного между упомянутым по меньшей мере одним значением вершины и упомянутым предсказанием.

19. Кодер по п.18, в котором упомянутый по меньшей мере один процессор

дополнительно сконфигурирован для кодирования флага для упомянутой вершины, указывающего, кодируется ли или нет по меньшей мере один остаток для упомянутой вершины.

20. Кодер по п.18 или 19, в котором упомянутое другое значение получается по меньшей мере из одного восстановленного значения соседней вершины.

21. Кодер по п.20, в котором упомянутое по меньшей мере одно значение упомянутой вершины, упомянутое другое значение и упомянутое по меньшей мере одно восстановленное значение упомянутой соседней вершины являются значениями цвета.

22. Кодер по п.21, в котором упомянутые значения цвета соответствуют преобразованию цветов.

23. Декодер для декодирования цветовой 3-мерной справочной таблицы октантов, содержащий интерфейс для обращения к потоку битов и по меньшей мере один процессор, сконфигурированный для:

декодирования флага разделения для текущего октанта, указывающего, декодируется ли рекурсивно по меньшей мере один из его непосредственных дочерних октантов; и восстановления упомянутого текущего октанта в ответ на упомянутый флаг разделения.

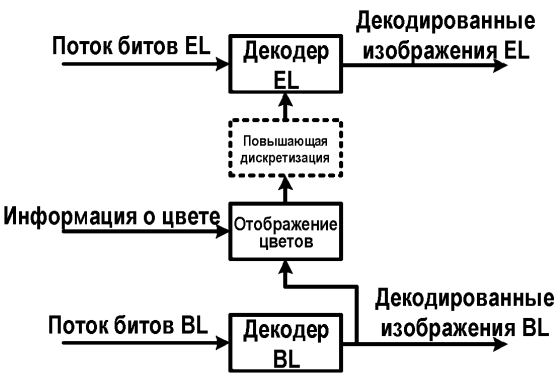
24. Декодер по п.23, в котором восстановление упомянутого текущего октанта содержит предсказание по меньшей мере одного значения вершины в упомянутом текущем октанте из другого значения, чтобы получить предсказание; и восстановление упомянутого по меньшей мере одного значения вершины из упомянутого предсказания и декодированного остатка.

25. Декодер по п.24, в котором упомянутый по меньшей мере один процессор дополнительно сконфигурирован для декодирования флага для упомянутой вершины, указывающего, декодируется ли или нет по меньшей мере один остаток для упомянутой вершины.

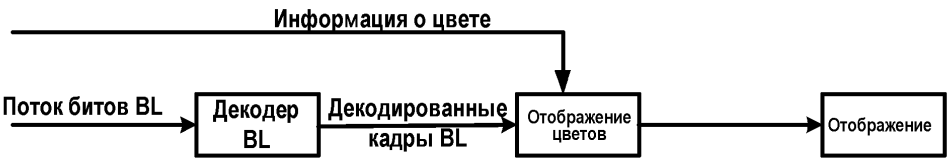
26. Декодер по п.24 или 25, в котором упомянутое другое значение получается по меньшей мере из одного восстановленного значения соседней вершины.

27. Декодер по п.26, в котором упомянутое по меньшей мере одно значение упомянутой вершины, упомянутое другое значение и упомянутое по меньшей мере одно восстановленное значение упомянутой соседней вершины являются значениями цвета.

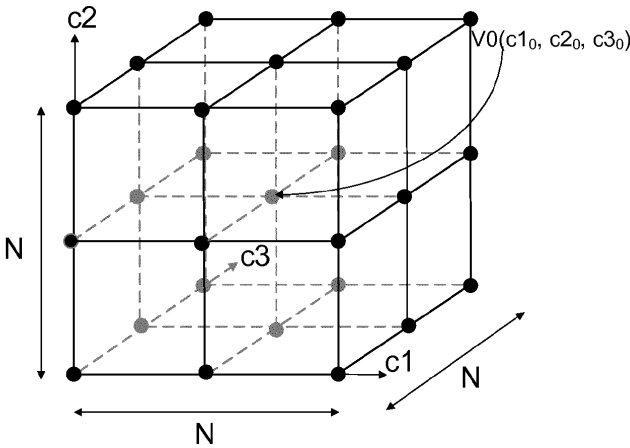
28. Декодер по п.27, в котором упомянутые значения цвета соответствуют преобразованию цветов.



ФИГ. 1 Известный уровень техники

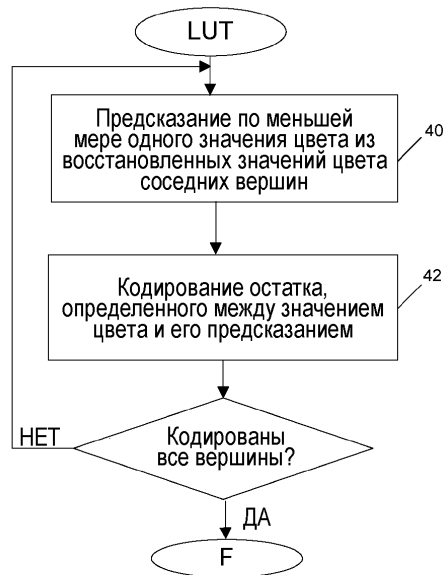


ФИГ. 2 Известный уровень техники

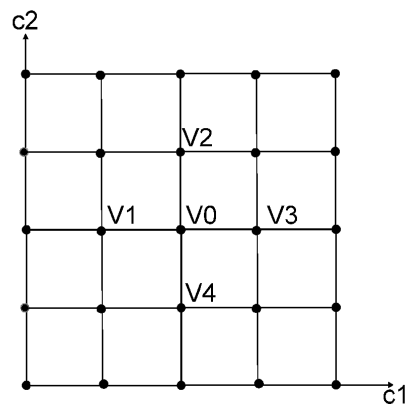


ФИГ. 3 Известный уровень техники

2/9

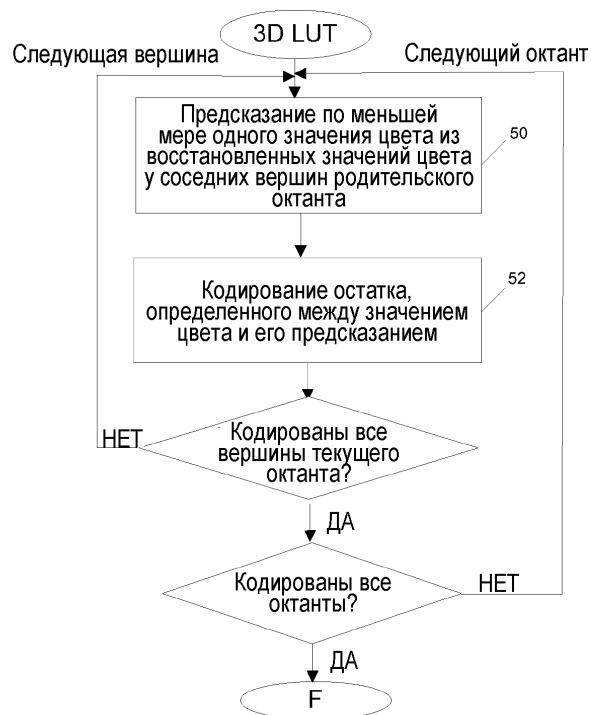


ФИГ. 4

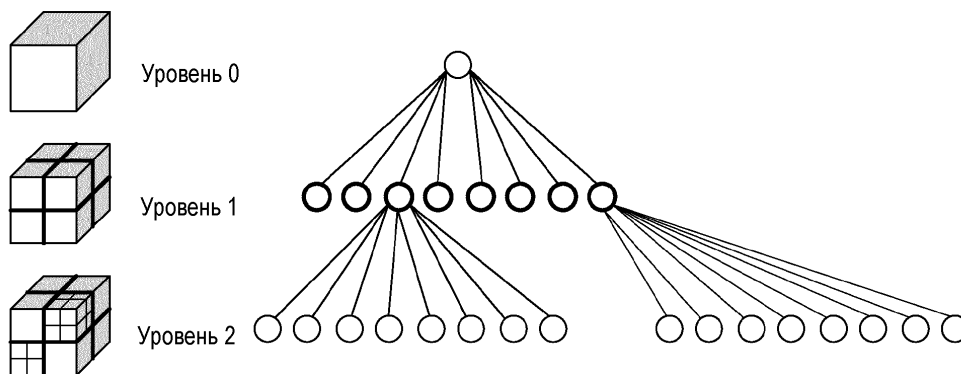


ФИГ. 5

3/9

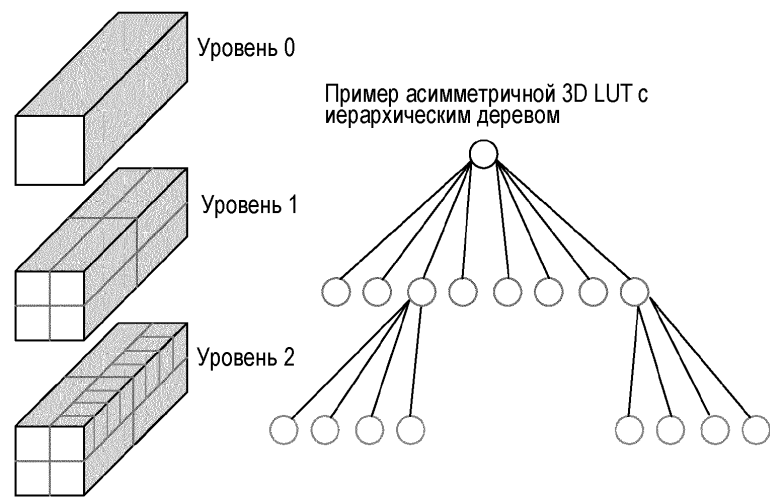


ФИГ. 6

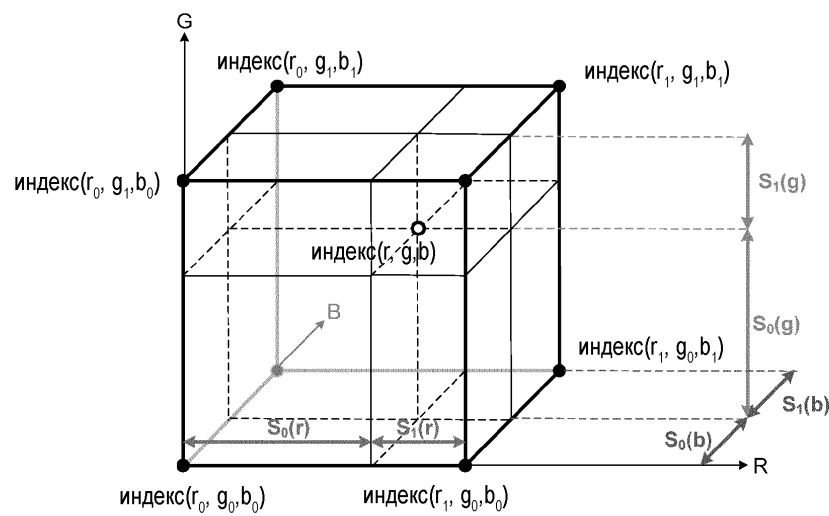


ФИГ. 7А

4/9



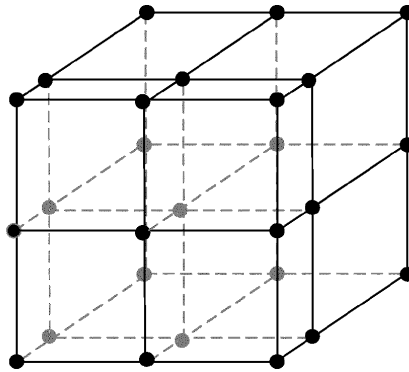
ФИГ. 7В



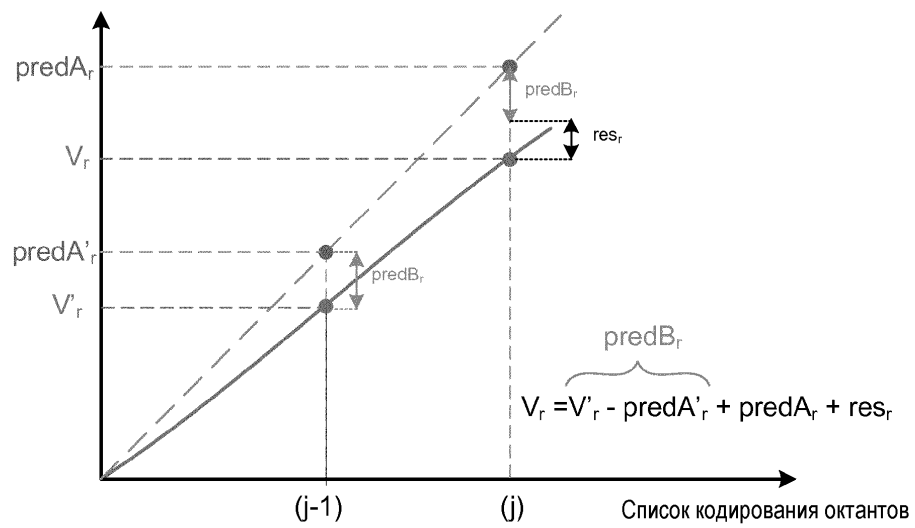
ФИГ. 8



5/9

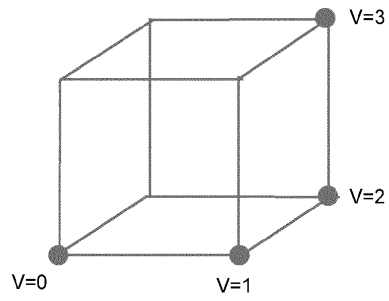


ФИГ. 9А

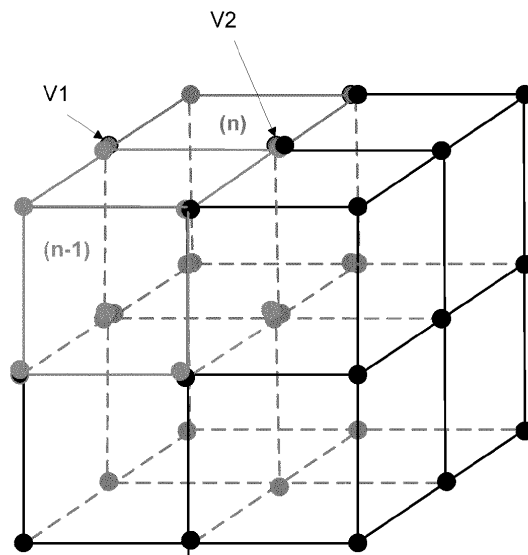


ФИГ. 9В

6/9

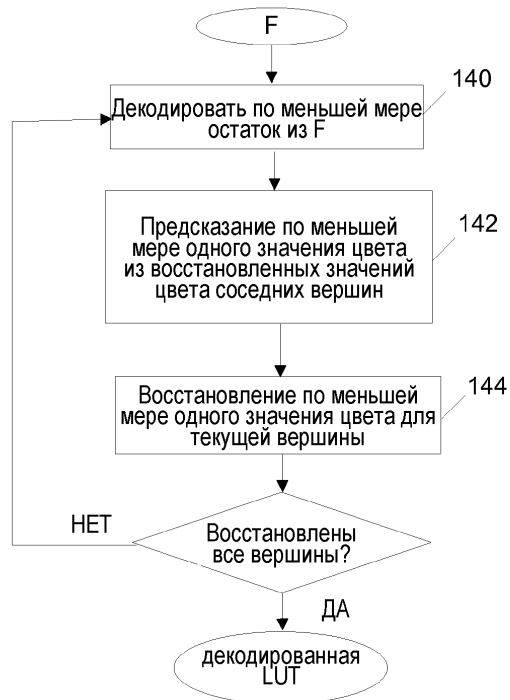


ФИГ. 9C



ФИГ. 9D

7/9

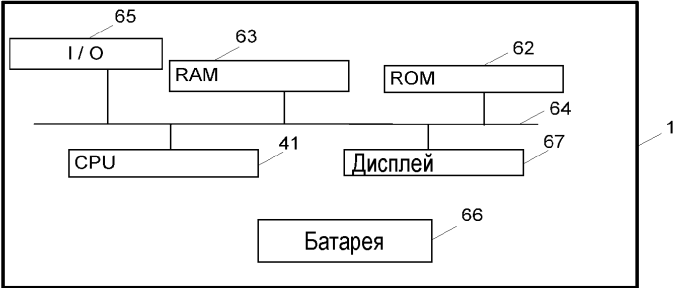


ФИГ. 10

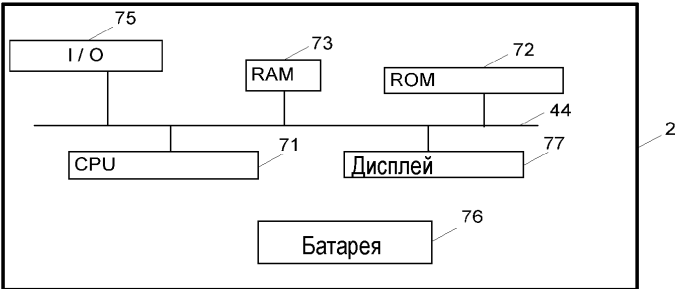
8/9



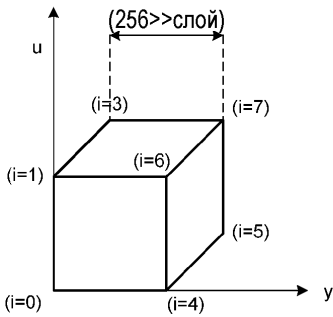
ФИГ. 11



ФИГ. 12



ФИГ. 13



ФИГ. 14