

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 983 458**

51 Int. Cl.:

H04N 19/50 (2014.01)

H04N 19/30 (2014.01)

H04N 19/186 (2014.01)

H04N 19/463 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **17.03.2014** **PCT/EP2014/055333**

87 Fecha y número de publicación internacional: **16.10.2014** **WO14166705**

96 Fecha de presentación y número de la solicitud europea: **17.03.2014** **E 14712249 (3)**

97 Fecha y número de publicación de la concesión europea: **12.06.2024** **EP 2984836**

54 Título: **Método para codificar y método para descodificar una LUT y correspondientes dispositivos**

30 Prioridad:

08.04.2013 EP 13305453

15.07.2013 EP 13306010

27.01.2014 EP 14305109

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
23.10.2024

73 Titular/es:

DOLBY INTERNATIONAL AB (100.0%)
77 Sir John Rogerson's Quay, Block C, Grand
Canal Docklands
Dublin, D02 VK60, IE

72 Inventor/es:

BORDES, PHILIPPE;
ANDRIVON, PIERRE y
JOLLY, EMMANUEL

74 Agente/Representante:

LINAGE GONZÁLEZ, Rafael

ES 2 983 458 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Método para codificar y método para descodificar una LUT y correspondientes dispositivos

5 Campo de la invención

Un método para codificar una tabla de consulta definida como una retícula de vértices se divulga, en el que al menos un valor está asociado con cada vértice de la retícula. De manera ejemplar, el valor es un valor de color. Un correspondiente método de descodificación, dispositivo de codificación y dispositivo de descodificación se divulgan.

Antecedentes de la invención

Como se representa en la figura 1, la descodificación de vídeo escalable consiste en descodificar (respectivamente codificar) un flujo de bits de capa de base (BL) y al menos un flujo de bits de capa de mejora (EL). Normalmente, las imágenes de EL se predicen a partir de imágenes de BL descodificadas (posiblemente sobremuestreadas). Sin embargo, cuando las imágenes de EL y las imágenes de BL se representan con diferentes espacios de color y/o han sido clasificadas por colores de manera diferente, la predicción es menos eficiente. Con el fin de mejorar la predicción, es conocido aplicar una transformada de color en las imágenes de BL descodificadas. De manera más precisa, la transformada de color mapea los colores del espacio de color de BL (primer espacio de color) en los colores del espacio de color de EL (segundo espacio de color) usando información de color.

Como se representa en la figura 2, en distribución de contenido de vídeo, una transformada de color se aplica normalmente en las imágenes descodificadas de modo que las imágenes descodificadas transformadas se adaptan a la capacidad de renderización del dispositivo final.

Esta transformada de color también se conoce como función de mapeo de color (CMF). La CMF se aproxima, por ejemplo, mediante una matriz de ganancia de 3×3 más un desplazamiento (modelo de desplazamiento de ganancia). En este caso, la CMF está definida por 12 parámetros. Sin embargo, tal aproximación de la CMF no es muy precisa porque asume un modelo de transformada lineal. En consecuencia, la tabla de consulta 3D (también conocida como LUT 3D) se usa para describir tal CMF, sin ningún "a priori" en el modelo de la CMF. La LUT 3D es mucho más precisa porque su tamaño puede aumentarse dependiendo de la precisión requerida. Sin embargo, la LUT 3D puede de este modo representar un conjunto de datos enorme. Transmitir una LUT 3D a un receptor de este modo requiere codificar la LUT.

Una LUT que se aproxima a una CMF asocia con al menos un valor de color en el primer espacio de color otro valor de color en el segundo espacio de color. Una LUT permite fraccionamiento del primer espacio de color en un conjunto de regiones delimitadas por los vértices de la LUT. De manera ejemplar, una LUT 3D asocia con un triplete de valores de color en el primer espacio de color un conjunto de valores de color. El conjunto de valores de color puede ser un triplete de valores de color en el segundo espacio de color o un conjunto de valores de color representativos de la transformada de color (por ejemplo, parámetros de CMF definidos localmente) usados para transformar valores de color en el primer espacio de color en valores de color en el segundo espacio de color. En la figura 3, una LUT 3D cuadrada se representa como una retícula de $N \times N \times N$ vértices. Para cada vértice $V(c_1, c_2, c_3)$ de la LUT 3D, un correspondiente triplete de valores de color (V_{c1}, V_{c2}, V_{c3}) necesita almacenarse. La cantidad de datos asociados con la LUT 3D es $N \times N \times N \times K$, donde K es la cantidad de bits usada para almacenar un valor de triplete de LUT. El valor de triplete es, por ejemplo un triplete (R, G, B), un triplete (Y, U, V) o un triplete (Y, Cb, Cr), etc. Codificar todos los valores de vértice no es eficiente ya que representa una cantidad de datos enorme.

Los documentos EP2051529, EP0822708, JP2004208128, JP2010251940 divulgan tablas de consulta para una transformada de color. Los documentos EP1574996, US6281903 y YAN HUANG ET AL: "A Generic Scheme for Progressive Point Cloud Coding", IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, IEEE SERVICE CENTER, LOS ALAMITOS, CA, EE.UU., vol. 14, nº 2, 1 de marzo de 2008 (2008-03-01), páginas 440 a 453, divulgan el uso de una estructura de árbol octal para representar datos 3D.

Breve resumen de la invención

El propósito de la invención es superar al menos uno de los inconvenientes de la técnica anterior. Un método para codificar una LUT, un método de descodificación, un codificador, un descodificador y un flujo de bits obtenido mediante el método de codificación se divulga de acuerdo con las reivindicaciones adjuntas y la descripción detallada a continuación. La presente invención se define por las reivindicaciones independientes. Las reivindicaciones dependientes tienen que ver con características opcionales de algunas realizaciones.

Breve descripción de los dibujos

Otras características y ventajas de la invención aparecerán con la descripción siguiente de algunas de sus realizaciones, realizándose esta descripción en relación con los dibujos en los cuales:

- 5 la figura 1 ilustra una arquitectura de un decodificador de vídeo escalable que usa escalabilidad de gama de acuerdo con la técnica anterior;
- la figura 2 ilustra una arquitectura de un decodificador de vídeo que comprende transformada de color para adaptarse a las características del visualizador de renderización de acuerdo con la técnica anterior;
- 10 la figura 3 muestra esquemáticamente una LUT 3D cuadrada representada como una retícula de vértices $N \times N \times N$ de acuerdo con la técnica anterior;
- la figura 4 representa un diagrama de flujo de un método de codificación de una LUT 3D de acuerdo con un ejemplo útil o que comprende la invención;
- 15 la figura 5 muestra esquemáticamente una LUT 2D;
- la figura 6 representa un diagrama de flujo de un método de codificación de acuerdo con otra realización ejemplar de la invención;
- 20 la figura 7 muestra esquemáticamente a la izquierda una subdivisión recursiva de un cubo en octantes y a la derecha un correspondiente árbol octal;
- la figura 8 ilustra una interpolación de los valores de color de un vértice de acuerdo con la invención;
- 25 la figura 9 muestra esquemáticamente una LUT 3D representada como una retícula no uniforme;
- la figura 10 representa un diagrama de flujo de un método de decodificación de una LUT 3D de acuerdo con un ejemplo útil para entender la invención;
- 30 la figura 11 representa un diagrama de flujo de un método de decodificación de acuerdo con otra realización ejemplar de la invención;
- la figura 12 muestra esquemáticamente un codificador para codificar una LUT de acuerdo con la invención;
- 35 la figura 13 muestra esquemáticamente un decodificador para decodificar una LUT de acuerdo con la invención; y
- 40 la figura 14 representa la posición de los 8 vértices de un octante.

5. Descripción detallada de la invención

La figura 4 representa un diagrama de flujo de un método de codificación de una LUT 3D de acuerdo con un ejemplo útil para entender la invención. El método es para codificar una LUT definida como una retícula de 45 vértices, en el que al menos un valor, por ejemplo un valor de color, está asociado con cada vértice de la retícula. En referencia a la decodificación, los términos "reconstrucción" y "decodificación" se usan muy a menudo como sinónimos. A continuación, el término valor de color comprende valores de color de un espacio de color dado, tales como valores RGB, YUV o Y,Cb,Cr y comprende además valores representativos de una transformada de color, tal como parámetros de CMF, es decir, parámetros de matriz y valores de 50 desplazamiento.

En un paso 40, el al menos un valor de color de un vértice actual se predice a partir de valores de color reconstruidos asociados con vértices vecinos.

55 A modo de ejemplo, una LUT 2D que asocia con un vértice $V_0(c_{10}, c_{20})$ un correspondiente par de valores (V_{0c1}, V_{0c2}), por ejemplo valores de color, se representa en la figura 5. Los valores asociados con el vértice V_0 se predicen por ejemplo a partir de vértices espacialmente vecinos $V_1(c_{11}, c_{21})$, $V_2(c_{12}, c_{22})$, $V_3(c_{13}, c_{23})$ y $V_4(c_{14}, c_{24})$ con correspondiente par de valores (V_{ic1}, V_{ic2}) $_{i=1, \dots, 4}$. Una predicción $P(Pc1, Pc2)$ se computa por ejemplo usando interpolación de la siguiente manera:

$$Pc1 = 0.25 * (V_{1c1} + V_{2c1} + V_{3c1} + V_{4c1})$$

$$Pc2 = 0.25 * (V_{1c2} + V_{2c2} + V_{3c2} + V_{4c2})$$

De acuerdo con una variante, la LUT 2D asocia con el vértice $V0(c1_0, c2_0)$ un conjunto de parámetros ($m11, m12, m21, m22, o1, o2$) en lugar del par de valores ($V0_{c1}, V0_{c2}$). Este conjunto de parámetros puede usarse para reconstruir los valores ($V0_{c1}, V0_{c2}$) a partir de los valores ($c1_0, c2_0$) de $V0$ de la siguiente manera:

$$V0_{c1} \approx m11 * c1_0 + m12 * c2_0 + o1$$

5 $V0_{c2} \approx m21 * c1_0 + m22 * c2_0 + o2$

Los parámetros asociados con el vértice $V0$ se predicen por ejemplo a partir de vértices espacialmente vecinos $V1(c1_1, c2_1), V2(c1_2, c2_2), V3(c1_3, c2_3)$ y $V4(c1_4, c2_4)$ con correspondiente par de parámetros ($V1_{c1}, V1_{c2}$)= $1, \dots, 4$. Una predicción $P(Pc1, Pc2)$ se computa por ejemplo usando interpolación de la siguiente manera:

10 $Pc1 = 0.25 * (V1_{c1} + V2_{c1} + V3_{c1} + V4_{c1})$

$$Pc2 = 0.25 * (V1_{c2} + V2_{c2} + V3_{c2} + V4_{c2})$$

En un paso 42, al menos un residuo se determina entre el al menos un valor de color del vértice actual y su predicción y se codifica además en un flujo de bits F . El residuo se determina restando del al menos un valor de color del vértice actual su predicción. La codificación comprende normalmente codificación por entropía. De acuerdo con una variante la codificación comprende cuantificar el residuo con un cuantificador q y codificar por entropía el residuo cuantificado.

20 Como ejemplo, en el caso de la LUT 2D, los residuos primero y segundo se computan para el vértice $V0$. El primer residuo es igual a $(V0_{c1} - Pc1)$ y el segundo residuo es igual a $(V0_{c2} - Pc2)$. Los residuos o los residuos cuantificados $(V0_{c1} - Pc1)/q$ y $(V0_{c2} - Pc2)/q$ se codifican entonces por entropía en el flujo de bits F . La codificación por entropía hace uso de técnicas de codificación binaria tradicionales tales como Exponencial-Golomb, Huffman, CABAC (acrónimo inglés de "Context Adaptive Binary Arithmetic Coding").

25 Los pasos 40 y 42 se iteran para codificar un vértice adicional de la LUT hasta que se codifican todos los vértices de la LUT.

Opcionalmente, el método de codificación comprende codificar en el flujo de bits F el valor cuantificador q . El al menos un valor de color del vértice actual se reconstruye además para usarse para la predicción de otros vértices. Cada al menos un valor de color reconstruido es igual al correspondiente valor de color original si no se usa cuantificador, es decir $q=1$. De lo contrario, el al menos un valor de color reconstruido se determina descuantiendo el residuo y añadiendo el residuo descuantiendo a la predicción.

35 De acuerdo con una variante, el método de codificación comprende además codificar en el flujo de bits F el tipo de interpolación usado para predecir el al menos un valor de color del vértice. De manera más precisa se descodifica un índice a partir del flujo de bits que identifica un tipo de interpolación. Como ejemplo, el índice 0 identifica la interpolación bilineal, un índice 1 identifica la interpolación de vecino más cercano, un índice 2 identifica interpolación lineal con los 2 vecinos más cercanos.

40 De acuerdo con una variante, no todos los vértices de la LUT se codifican en el flujo de bits F . Por ejemplo, si el valor absoluto de todos los residuos o de todos los residuos cuantificados de un vértice están por debajo de un valor de umbral TH entonces no se codifica ningún residuo para ese vértice, por ejemplo $TH=0$ o $TH=1$. Un indicador binario se codifica de este modo en el flujo de bits para cada vértice que indica si al menos un residuo se codifica para ese vértice o si ninguno de los residuos se codifica y de este modo se infiere que son cero.

45 De acuerdo con una variante, se codifica un indicador binario para cada valor de color de cada vértice que indica si un residuo está codificado para ese valor de color o si el residuo no está codificado y se infiere que es cero.

El tamaño de la LUT también se codifica opcionalmente en el flujo de bits.

50 La figura 6 representa un diagrama de flujo de un método de codificación de acuerdo con otra realización ejemplar de la invención. El método es para codificar una LUT 3D definida como una retícula de vértices usando un árbol octal, en el que un conjunto de n valores de color tales como un triplete de valores de color se asocian con cada vértice de la retícula, donde n es un número entero ≥ 1 . Un árbol octal es para fraccionar un espacio de color 3D subdividiéndolo recursivamente en ocho octantes como se representa en la figura 7. Un octante de nivel N tiene su octante padre en el nivel $N-1$. Los árboles octales son el análogo 3D de árboles cuádruples. El método de codificación se divulga para la codificación de un vértice actual V de la retícula que pertenece a un octante

actual. A continuación $n=3$.

En un paso 50, cada uno de los tres valores de color (V_r , V_g , V_b) asociado al vértice actual V de coordenadas (r , g , b) se predice a partir de valores de color reconstruidos asociados a vértices vecinos, es decir vértices que pertenecen a un octante padre del octante actual. (R , g , b) se usa en lugar de (c_1 , c_2 , c_3) para simplificar las notaciones. Pero la invención no está limitada al espacio de color (R , G , B). Puede aplicarse a (Y , U , V), (Y , Cb , Cr),... representaciones de espacios de color.

Se determina de este modo una predicción para cada valor de color usando por ejemplo una interpolación trilineal como se ilustra en la figura 8:

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

donde: (r_i , g_j , b_k) con $i=0, 1$, $j=0, 1$ y $k=0, 1$ son las coordenadas de los vértices del octante padre en el espacio de color 3D;

(r , g , b) son las coordenadas del vértice actual;

$\text{LUT}[r_i][g_j][b_k]$. r es el primer valor de color asociado con el vértice (r_i , g_j , b_k);

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

$s_0(t) = t_1 - t$, y $s_1(t) = t - t_0$ con $t = r$, g o b .

Se usan las mismas ecuaciones para g y b . Se puede usar otro tipo de interpolación tal como prismática, piramidal o tetraédrica.

La retícula no es necesariamente uniforme como se representa en la figura 9.

En un paso 52, se computan tres residuos para el vértice actual, uno para cada componente de color: $\text{res}_r = (V_r - \bar{V}_r)$, $\text{res}_g = (V_g - \bar{V}_g)$ y $\text{res}_b = (V_b - \bar{V}_b)$.

Los residuos después se codifican por entropía en un flujo de bits o se cuantifican antes de codificarse por entropía. La codificación por entropía hace uso de técnicas de codificación binaria tradicionales tales como Exponencial-Golomb, Huffman, CABAC (acrónimo inglés de "Context Adaptive Binary Arithmetic Coding").

El octante de nivel 0 no tiene octante padre. De la misma manera, el primer octante en la lista de octantes de codificación no tiene octante precedente. Por consiguiente, cada uno de los tres valores de color (V_r , V_g , V_b) asociados a un vértice actual V de este octante se predice a partir de un valor de color conocido, por ejemplo el valor 128. De acuerdo con una variante, se usan diferentes valores conocidos para los diferentes vértices del octante de nivel 0. De acuerdo con otra variante, la predicción a partir de valores de color conocidos se realiza también para otros octantes distintos del octante de nivel 0. De acuerdo con una variante, cada uno de los tres valores de color (V_r , V_g , V_b) asociado a un vértice actual V de este octante se predice a partir de valores de color ya reconstruidos asociados a vértices vecinos que pertenecen al mismo octante.

Opcionalmente, el método de codificación comprende codificar en el flujo de bits el valor cuantificador q . Los tres valores de color del vértice actual se reconstruyen adicionalmente y pueden usarse para la predicción de otros vértices. Cada uno de los valores de color reconstruidos es igual al valor de color original si no se usa cuantificador ($q=1$). De lo contrario, cada valor de color reconstruido se determina descuantificando el correspondiente residuo y añadiendo el residuo descuantificado a la correspondiente predicción.

De acuerdo con una variante, el método de codificación comprende además codificar en el flujo de bits el tipo de interpolación usado para predecir el al menos un valor de color del vértice. De manera más precisa se codifica un índice en el flujo de bits que identifica un tipo de interpolación. Como ejemplo, el índice 0 identifica la interpolación trilineal, el índice 1 identifica una interpolación prismática, el índice 2 identifica una interpolación piramidal y el índice 3 identifica una interpolación tetraédrica.

De acuerdo con una variante, no todos los vértices de la LUT están codificados en el flujo de bits. Por ejemplo, si los valores absolutos de todos los residuos o de todos los residuos cuantificados de un vértice están por debajo de un valor de umbral TH entonces no se codifica ningún residuo para ese vértice, por ejemplo $TH=0$ o $TH=1$. Un indicador binario se codifica de este modo en el flujo de bits para cada vértice que indica si al menos un residuo se codifica o no para ese vértice. De acuerdo con una variante, se codifica un indicador binario para cada valor

de color de cada vértice que indica si un residuo está codificado para ese valor de color o si el residuo no está codificado y se infiere que es cero. La LUT se codifica normalmente en un flujo de bits para usarse para transformar imágenes de un vídeo en aplicaciones tales como renderización de adaptación de color de visualización o escalabilidad de gama de color. La LUT puede codificarse y transmitirse con el vídeo. Si el método de codificación conoce que algunas partes de la LUT 3D no van a usarse por la aplicación, entonces los vértices que pertenecen a esta parte de la LUT 3D no están codificados. De la misma manera, si el método de codificación conoce que algunas partes de la LUT 3D tienen pequeños impactos en la renderización final del vídeo, entonces los vértices que pertenecen a esta parte de la LUT 3D no se codifican.

El método se aplica recursivamente para codificar toda la LUT 3D. En este caso, todos los octantes del árbol octal se codifican. Una vez que se codifican todos los vértices de un octante actual, los vértices del octante hijo del octante actual se codifican.

En una variante, la LUT 3D se preprocesa antes de codificarse. En este caso, un indicador de división se asocia con cada octante en el árbol octal y se establece inicialmente como falso. Durante el paso de preprocesamiento, los valores de indicador de división se determinan para cada octante. Si al menos un vértice a codificar de un octante actual tiene al menos un residuo posiblemente cuantificado mayor que TH, entonces el indicador de división de su octante padre se establece como "verdadero". El indicador de división de un octante actual de nivel N indica de este modo si sus hijos inmediatos (es decir hijos de nivel N+1) están codificados recursivamente o si todos los residuos de los vértices de todos sus hijos (es decir hijos de nivel N+k con k>0) todavía no codificados se infiere que son cero.

Durante el paso de codificación, los indicadores de división y los residuos se codifican en el flujo de bits. Una vez que se codifican todos los vértices de un octante actual, los vértices de los octantes hijos del octante actual se codifican si el indicador de división de octante actual es verdadero. Los vértices que pertenecen a dos octantes se codifican preferentemente solo una vez.

La LUT 3D se codifica, por ejemplo, en el VPS ("conjunto de parámetros de vídeo"), el SPS ("conjunto de parámetros de secuencia"), el PPS ("conjunto de parámetros de imagen") o en un mensaje de SEI ("información de mejora suplementaria") como se define en las normas de codificación de vídeo AVC, HEVC, SVC o SHVC. La LUT 3D se codifica por ejemplo en un mensaje de SEI tal como el definido a continuación. El tamaño S de la LUT también se codifica opcionalmente en el flujo de bits. S es el número de vértices en una dirección.

De acuerdo con una variante, n LUT 3D compuestas por vértices con valor de color 1 se codifican en lugar de codificar una LUT 3D compuesta por vértices con n valores de color asociados, por ejemplo n=3, como se divulgó anteriormente.

De acuerdo con una variante, cuando el tamaño de la LUT 3D es S=2, se codifica un indicador que indica al decodificador que compute una matriz de ganancia de 3x3 más un desplazamiento de la LUT 3D decodificada y que la use como CMF en lugar de la LUT 3D.

De acuerdo con otra variante, una LUT 3D compuesta por vértices con n valores de color asociados se usa para codificar parámetros de una transformada de color definida localmente. De manera ejemplar, cada vértice de la LUT 3D está asociado con 12 valores de color representativos de una transformada de color. En lugar de asociar los 3 valores de color (V_r, V_g, V_b) con el vértice actual V, 12 valores de color representativos de una transformada de color están asociados con el vértice V(r,g,b), en el que los 12 valores de color [a_i, b_i, c_i, o_i] con i=0,1 o 2 se definen de modo que se verifica la siguiente ecuación:

$$\begin{pmatrix} V_r \\ V_g \\ V_b \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} + \begin{pmatrix} o_0 \\ o_1 \\ o_2 \end{pmatrix}$$

De acuerdo con una variante, solo los parámetros [a_i, b_i, c_i] con i=0,1 o 2 están asociados a los vértices.

En verdad, en el caso de una LUT 3D de tamaño 2 (un solo octante de 8 vértices), se pueden elegir los tres valores de color de los vértices de modo que la LUT 3D es equivalente al modelo de desplazamiento de ganancia definido por la ecuación siguiente:

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

Para ello, se tienen que establecer los 3 valores de color (y_{xu}x_v)_{x=A,...,H} de los 8 vértices A a H a los siguientes valores:

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

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

- 5 K es una constante que depende del número de bits usados para representar cada valor de color. K=255 si los valores de color se representan en 8 bits, K=1023 si los valores de color se representan en 10 bits, etc. Entonces, la interpolación trilineal es equivalente al modelo de desplazamiento de ganancia.

La figura 10 representa un diagrama de flujo de un método de descodificación de una LUT 3D de acuerdo con un ejemplo útil para entender la invención.

10

En un paso 140, al menos un residuo se descodifica a partir de un flujo de bits F. La descodificación comprende normalmente descodificación por entropía. De acuerdo con una variante la descodificación comprende descodificación por entropía de un residuo cuantificado y cuantificación inversa del residuo cuantificado con un cuantificador q. La descodificación por entropía hace uso de técnicas de descodificación binaria tradicionales

15

tales como Exponencial-Golomb, Huffman, CABAC (acrónimo inglés de "Context Adaptive Binary Arithmetic Coding").

Opcionalmente, el método de descodificación comprende descodificar a partir del flujo de bits F el valor cuantificador q.

20

De acuerdo con una variante, no todos los vértices de la LUT están codificados en el flujo de bits F. Opcionalmente, se descodifica un indicador binario del flujo de bits para cada vértice que indica si al menos un residuo está codificado o no para ese vértice. Si no se codifica ningún residuo, se asume que el residuo o residuos son cero para este vértice.

25

En un paso 142, el al menos un valor de color de un vértice actual se predice a partir de valores de color reconstruidos asociados con vértices vecinos. Como un ejemplo de una LUT 2D que asocia con un vértice V0(c1, c2) un correspondiente par de valores de color (V0c1, V0c2) se representa en la figura 5. Los valores de color asociados con el vértice V0 se predicen a partir de vértices espacialmente vecinos V1, V2, V3 y V4. Como ejemplo, un predictor P(Pc1, Pc2) se computa usando interpolación de la siguiente manera:

30

$$Pc1 = 0.25 * (V1_{c1} + V2_{c1} + V3_{c1} + V4_{c1})$$

$$Pc2 = 0.25 * (V1_{c2} + V2_{c2} + V3_{c2} + V4_{c2})$$

35

En un paso 144, se reconstruye un vértice. De manera más precisa, al menos un valor de color del vértice actual se reconstruye a partir de su predicción y el al menos un residuo descodificado. Como ejemplo, en el caso de la LUT 2D, dos residuos R_{c1} y R_{c2} se descodifican para el vértice V0 actual. El vértice actual se reconstruye de este modo computando sus dos valores de color de la siguiente manera: (R_{c1}+Pc1) y (R_{c2}+Pc2).

40

De acuerdo con una variante, el método de descodificación comprende además descodificar a partir del flujo de

bits F el tipo de interpolación usado para predecir el al menos un valor de color del vértice. De manera más precisa se descodifica un índice a partir del flujo de bits que identifica un tipo de interpolación. Como ejemplo, el índice 0 identifica la interpolación bilineal y un índice 1 identifica una interpolación de valor de vértices más cercanos.

5

El tamaño de la LUT también se descodifica opcionalmente a partir del flujo de bits.

10

La figura 11 representa un diagrama de flujo de un método de descodificación de acuerdo con otra realización ejemplar de la invención. El método es para descodificar una LUT 3D definida como una retícula de vértices usando un árbol octal, en el que un triplete de valores de color está asociado con cada vértice de la retícula. El método de descodificación se divulga para descodificar un vértice actual V de la retícula que pertenece a un octante actual.

15

En un paso 150, tres residuos res_r , res_g , res_b se descodifican a partir de un flujo de bits F. La descodificación comprende normalmente descodificación por entropía. De acuerdo con una variante la descodificación comprende descodificación por entropía de un residuo cuantificado y cuantificación inversa del residuo cuantificado con un cuantificador q. La descodificación por entropía hace uso de técnicas de descodificación binaria tradicionales tales como Exponencial-Golomb, Huffman, CABAC (acrónimo inglés de "Context Adaptive Binary Arithmetic Coding").

20

Opcionalmente, el método de descodificación comprende descodificar a partir del flujo de bits F el valor cuantificador q.

25

De acuerdo con una variante, no todos los vértices de la LUT están codificados en el flujo de bits F. Opcionalmente, se descodifica un indicador binario a partir del flujo de bits para cada vértice que indica si al menos un residuo está codificado o no para ese vértice. Si no se codifica ningún residuo, se asume que el residuo o residuos son cero para este vértice.

30

En un paso 152, cada uno de los tres valores de color (V_r , V_g , V_b) asociados al vértice actual V de coordenadas (r, g, b) se predice a partir de valores de color reconstruidos asociados a vértices vecinos, es decir vértices que pertenecen a un octante padre del octante actual. (R, g, b) se usa en lugar de (c1, c2, c3) para simplificar las notaciones. Pero la invención no está limitada al espacio de color (R, G, B). Puede aplicarse a representaciones de espacios de color (Y,U,V), (Y,Cb,Cr)...

35

Una predicción se determina de este modo para cada valor de color.

Una predicción se determina de este modo para cada valor de color usando por ejemplo una interpolación trilineal como se ilustra en la figura 8:

40

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

donde: (r_i , g_j , b_k) con $i=0, 1$, $j=0, 1$ y $k=0, 1$ son las coordenadas de los vértices del octante padre en el espacio de color 3D;

45

(r, g, b) son las coordenadas del vértice actual;

$LUT[r_i][g_j][b_k].r$ es el primer valor de color asociado al vértice (r_i , g_j , b_k);

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

50

$s_0(t) = t_1 - t$, y $s_1(t) = t - t_0$ con $t = r, g$ o b .

Se usan las mismas ecuaciones para g y b. Se puede usar otro tipo de interpolación tal como prismática, piramidal o tetraédrica. La retícula no es necesariamente uniforme como se representa en la figura 9.

55

En un paso 154, se computan tres valores de color de este modo reconstruidos para el vértice actual a partir de su predicción y de los correspondientes residuos descodificados (res_r , res_g , res_b). El vértice actual se reconstruye de este modo computando sus tres valores de color de la manera siguiente: ($res_r + \bar{V}_r$), ($res_g + \bar{V}_g$) y ($res_b + \bar{V}_b$).

60

El octante de nivel 0 no tiene octante padre. De la misma manera, el primer octante en la lista de octantes de codificación no tiene octante precedente. Por consiguiente, cada uno de los tres valores de color (V_r , V_g , V_b) asociados a un vértice V actual de este octante se predice a partir de un valor de color conocido, por ejemplo el valor 128. De acuerdo con una variante, cada uno de los tres valores de color (V_r , V_g , V_b) asociados a un vértice

actual V de este octante se predice a partir de valores de color ya reconstruidos asociados a vértices vecinos que pertenecen al mismo octante.

5 El método se aplica recursivamente para descodificar toda la LUT 3D. Los vértices que pertenecen a dos octantes se descodifican preferentemente sólo una vez.

Una vez que todos los vértices de un octante actual (nivel N) se descodifican, los vértices de los octantes hijos (nivel N+1) del octante actual se descodifican.

10 De acuerdo con una variante, se descodifica un señalizador de división para un octante actual de nivel N que indica si sus hijos inmediatos (es decir, hijos de nivel N+1) se descodifican recursivamente o si todos los residuos de los vértices de todos los hijos (es decir, hijos de nivel N+k con k>0) todavía no descodificados se infiere que son cero.

15 De acuerdo con una variante, el método de descodificación comprende además descodificar a partir del flujo de bits F el tipo de interpolación usado para predecir el al menos un valor de color del vértice. De manera más precisa se descodifica un índice a partir del flujo de bits que identifica un tipo de interpolación. Como ejemplo, el índice 0 identifica la interpolación trilineal, el índice 1 identifica una interpolación prismática, el índice 2 identifica una interpolación piramidal y el índice 3 identifica una interpolación tetraédrica.

20 La LUT 3D por ejemplo se descodifica a partir de los VPS, SPS, PPS o en un mensaje de SEI tal como se define en las normas de codificación de video AVC, HEVC, SVC o SHVC. El tamaño de la LUT también se descodifica opcionalmente a partir del flujo de bits. La LUT 3D por ejemplo se descodifica en un mensaje de SEI tal como el definido a continuación.

25 De acuerdo con una variante, n LUT 3D compuestas por vértices con valor de color 1 se descodifican en lugar de descodificar una LUT 3D compuesta por vértices con n valores de color, por ejemplo n=3. De acuerdo con una variante, cuando el tamaño de la LUT 3D es S=2, se descodifica un indicador que indica al descodificador que compute una matriz de ganancia de 3x3 más un desplazamiento de la LUT 3D descodificada y que la use como CMF en lugar de la LUT 3D.

Las mismas variantes divulgadas con respecto al método de codificación de acuerdo con las figuras 4 y 6 pueden aplicarse en el lado del descodificador.

35 En las figuras 4, 6, 10 y 11, las cajas representadas son entidades puramente funcionales, que no corresponden necesariamente a entidades físicas separadas. Como apreciará una persona experta en la técnica, aspectos de los presentes principios pueden realizarse como un sistema, método o medio legible por ordenador. Por consiguiente, aspectos de los presentes principios pueden tomar la forma de una realización completamente de hardware, una realización completamente de software (incluyendo firmware, software residente, microcódigo y demás), o una realización que combina aspectos de software y hardware que pueden denominarse todos en general en el presente documento como un "circuito", "módulo" o "sistema". Además, aspectos de los presentes principios pueden tomar la forma de un medio de almacenamiento legible por ordenador. Cualquier combinación de uno o más medios de almacenamiento legibles por ordenador puede usarse.

45 El diagrama de flujo y/o los diagramas de bloques en las figuras ilustran la configuración, funcionamiento y funcionalidad de posibles implementaciones de sistemas, métodos y productos de programa informático de acuerdo con diversas realizaciones de la presente invención. A este respecto, cada bloque en el diagrama de flujo o diagramas de bloques puede representar un módulo, segmento o porción de código, que comprende una o más instrucciones ejecutables para implementar la función o funciones lógicas especificadas. También debe observarse que, en algunas implementaciones alternativas, las funciones indicadas en el bloque pueden producirse fuera del orden indicado en las figuras. Por ejemplo, dos bloques mostrados en sucesión pueden, de hecho, ejecutarse sustancialmente de manera simultánea, o los bloques pueden ejecutarse a veces en el orden inverso, o bloques pueden ejecutarse en un orden alternativo, dependiendo de la funcionalidad implicada. También se observará que cada bloque de los diagramas de bloques y/o la ilustración de diagrama de flujo, y combinaciones de los bloques en los diagramas de bloques y/o la ilustración de diagrama de flujo, pueden implementarse por sistemas basados en hardware de propósito especial que realizan las funciones o actos especificados, o combinaciones de hardware de propósito especial e instrucciones informáticas. Aunque no se describe explícitamente, las presentes realizaciones pueden emplearse en cualquier combinación o subcombinación.

60 También se divulga un flujo de bits que codifica una LUT tal como una LUT 3D. El flujo de bits producido por el método de codificación de figura que codifica al menos una LUT definida como una retícula de vértices, en el que al menos un valor de color está asociado con cada vértice de la retícula, en el que dicho flujo de bits comprende codificado en él al menos un residuo computado entre el al menos un valor de color de un vértice actual y su predicción.

Se propone una realización ejemplar en el marco de la norma de codificación HEVC definida en el documento JCTVC-L1003 del equipo colaborativo conjunto de codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11 o la norma de codificación SHVC que es la extensión escalable de la norma de codificación HEVC definida en el documento JCTVC-L1008 del equipo colaborativo conjunto de codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11. La norma define una sintaxis que cualquier flujo de datos codificados debe cumplir para ser compatible con esta norma. La sintaxis define en particular cómo las diversas entradas de información se codifican (por ejemplo los datos relativos a las imágenes incluidas en la secuencia, los vectores de movimiento, etc.). En el contexto de la norma de codificación SHVC, la LUT puede codificarse en el PPS o el VPS. El elemento sintáctico, `use_color_prediction`, se usa para indicar el uso de predicción de color en la imagen actual como se muestra en la tabla 1.

Tabla 1 - señalización de los parámetros de predicción

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

Si el indicador es `use_color_prediction` es igual a '1', la función `3D_LUT_color_data` es llamada para señalar datos de LUT 3D como se muestra en la tabla 2.

Tabla 2 - codificación de datos de color de la LUT 3D

<code>3D_LUT_color_data () {</code>	Descriptor
nbpCode	<code>u(3)</code>
<code>coding_octant(0, 0, 0)</code>	
<code>}</code>	

`nbpCode` indica el tamaño de la LUT 3D como se enumera en la tabla 4 para el valor dado de `nbpCode`. El valor cuantificador puede codificarse por la función `3D_LUT_color_data ()`.

De acuerdo con una variante, `3D_LUT_color_data()` se define en la tabla 3 de la siguiente manera.

Tabla 3: codificación de datos de color de la LUT 3D

<code>3D_LUT_color_data () {</code>	Descriptor
nbpCode	<code>u(3)</code>
NbitsPerSample	<code>u(5)</code>
<code>coding_octant(0, 0, 0)</code>	
<code>}</code>	

`nbpCode` indica el tamaño de la LUT 3D como se enumera en la tabla 4 para el valor dado de `nbpCode`. El valor cuantificador puede codificarse por la función `3D_LUT_color_data()`. `NbitsPerSample` indica un número de bits usados para representar los valores de color.

Tabla 4 - interpretación de `nbpCode`

nbpCode	Tamaño de LUT 3D
0	2
1	3
2	5
3	9
4	17
5	33

La descodificación del octante (capa, y, u, v) es una función recursiva como se muestra en la tabla 5. Cada octante está compuesto por 8 vértices asociados con un indicador (encoded_flag[i]) que indica si los valores de color residuales están codificados o se infiere que todos son cero. Los valores de color se reconstruyen añadiendo los residuos a la predicción de los valores de color. La predicción de los valores de color se computa usando por ejemplo interpolación trilineal de los 8 vértices vecinos de layer_id-1.

Tabla 5 - elementos de sintaxis para coding_octant()

coding_octant (layer_id, y,u,v) {	Descriptor
for(i = 0; i < 8 ; i++) {	
encoded_flag[i]	u(1)
if (encoded_flag[i]) {	
resY[i]	ue(v)
resU[i]	ue(v)
resV[i]	ue(v)
}	
}	
split_flag	u(1)
if (split_flag) {	
for(i = 0; i < 8 ; i++) {	
coding_octant (layer_id+1, y+dy[i],u+du[i],v+dv[i])	
}	
}	
}	

- 10 De acuerdo con otra realización ventajosa, la LUT se codifica en un mensaje de SEI (SEI significa "información de mejora suplementaria"). La norma HEVC define en su anexo D la forma en que se codifica información adicional denominada SEI. Esta información adicional se referencia en la sintaxis por un campo llamado payloadType. Los mensajes de SEI ayudan por ejemplo a procesos relacionados con la visualización. Obsérvese que si el dispositivo de descodificación no posee las funcionalidades necesarias para su uso, esta información se
- 15 ignora. De acuerdo con una realización específica de la invención, se define un nuevo tipo de mensaje de SEI como para codificar información adicional relacionada con la LUT 3D. Para este propósito, un nuevo valor para el campo payloadType se define de entre los valores no usados todavía (por ejemplo payloadType es igual a 24).

La sintaxis de los datos de SEI (es decir, sei_payload) se extiende de la siguiente manera:

20

Tabla 6 - mapeo de color de mensaje de SEI

color_mapping_info(payloadSize) {	Descriptor
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)
if(colour_description_present_flag) {	
color_primaries_input_id	u(8)
color_primaries_output_id	u(8)
}	
color_output_rgb	u(1)
lut_bit_depth_minus8	u(4)
3D_LUT_color_data ()	
}	
}	

De acuerdo con una variante, el mensaje de SEI comprende además un indicador color_interpolator_id, por ejemplo después color_description_present_flag, cuyo valor indica un tipo de interpolación como se especifica en la tabla 7.

Tabla 7 - tipo de interpolación

color_interpolator_id	Tamaño de LUT 3D
0	Trilineal
1	Tetraédrica

Este mensaje de SEI proporciona información para permitir el remapeo de las muestras de color de las imágenes decodificadas de salida para personalización a entornos de visualización particulares. El proceso de remapeo mapea valores de muestra codificados en el espacio de color RGB a valores de muestra objetivo. Los mapeos se expresan en el dominio del espacio de color luma o RGB, y deben aplicarse al componente luma o a cada componente RGB producido por conversión del espacio de color de la imagen decodificada en consecuencia.

3D_LUT_color_data() se define en la tabla 2 o 3.

La LUT 3D decodificada se aplica a imágenes decodificadas que pertenecen a una capa identificada por ejemplo por el índice nuh_layer_id de la cabecera de unidad de NAL (véase la sección 7.3.1.2 del documento de la norma de codificación HEVC definida en el documento JCTVC-L1003 del equipo de colaboración conjunta de codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11 del mensaje de SEI.

color_map_id contiene un número de identificación que puede usarse para identificar el propósito del modelo de mapeo de color. Valores de color_map_id pueden usarse como determinados por la aplicación. El map_id de color puede usarse para soportar operaciones de mapeo de color que son adecuadas para diferentes escenarios de visualización. Por ejemplo, diferentes valores de color_map_id pueden corresponder a diferentes profundidades de bits de visualización.

color_map_cancel_flag igual a 1 indica que el mensaje de SEI de información de mapeo de color cancela la persistencia de cualquier mensaje de SEI de información de mapeo de color anterior en el orden de salida. color_map_cancel_flag igual a 0 indica que a información de mapeo de color sigue.

color_map_repetition_period especifica la persistencia del mensaje de SEI de información de mapeo de color y puede especificar un intervalo de recuento de orden de imagen dentro del cual otro mensaje de SEI de información de mapeo de color con el mismo valor de color_map_id o el final de la secuencia de vídeo codificada estará presente en el flujo de bits. color_map_repetition_period igual a 0 especifica que la información de mapa de color se aplica únicamente a la imagen decodificada actual.

color_map_repetition_period igual a 1 especifica que la información de mapa de color persiste en el orden de

salida hasta que cualquiera de las siguientes condiciones es verdadera:

- Una nueva secuencia de vídeo codificada empieza.

5 - Una imagen en una unidad de acceso que contiene un mensaje de SEI de información de mapeo de color con el mismo valor de color_map_id es emitida teniendo un recuento de orden de imagen (conocido como POC) mayor que el POC de la imagen descodificada actual, denominado PicOrderCnt(CurrPic).

10 color_map_repetition_period igual a 0 o igual a 1 indica que otro mensaje de SEI de información de mapeo de color con el mismo valor de color_map_id puede estar presente o no.

color_map_repetition_period mayor que 1 especifica que la información de mapa de color persiste hasta que cualquiera de las siguientes condiciones es verdadera:

15 - Una nueva secuencia de vídeo codificada empieza.

- Una imagen en una unidad de acceso que contiene un mensaje de SEI de información de mapeo de color con el mismo valor de color_map_id es emitida teniendo un POC mayor que PicOrderCnt(CurrPic) y menor que o igual a PicOrderCnt(CurrPic) + color_map_repetition_period.

20 color_map_repetition_period mayor que 1 indica que otro mensaje de SEI de información de mapeo de color con el mismo valor de color_map_id estará presente para una imagen en una unidad de acceso que se emite que tiene un POC mayor que PicOrderCnt(CurrPic) y menor o igual que PicOrderCnt(CurrPic) + color_map_repetition_period; a menos que finalice el flujo de bits o una nueva secuencia de vídeo codificada comience sin emitir tal imagen.

25 color_description_present_flag igual a 1 especifica que colour_primaries_input_id y colour_primaries_output_id están presentes. colour_description_present_flag igual a 0 especifica que colour_primaries_input_id y colour_primaries_output_id no están presentes.

30 color_primaries_input_id indica las coordenadas de cromaticidad de los primarios de fuente como se especifica en la tabla 8 en términos de la definición de CIE 1931 de x e y como se especifica por la ISO 11664-1.

35 color_primaries_output_id indica las coordenadas de cromaticidad de los primarios mapeados en color como se especifica en la tabla 8 en términos de la definición de CIE 1931 de x e y como se especifica por la ISO 11664-1, una vez que se aplica la LUT de color 3D.

40 color_output_rgb igual a 1 especifica que las muestras de color emitidas son señales de luma y croma. color_output_rgb igual a 0 especifica que las muestras de color emitidas son valores de verde, rojo, azul.

lut_bit_depth_minus8 especifica la profundidad de bits de las muestras de LUT 3D.

45 nbp_code indica el tamaño de la LUT 3D nbp como se enumera en la tabla 4 para el valor dado de nbp_code.

La salida de la descodificación de la LUT 3D es una LUT de serie de 3 dimensiones de tamaño nbp x nbp x nbp. Cada elemento de la serie de la LUT se denomina vértice y se asocia con 3 valores de muestra reconstruidos (recSamplesY, recSamplesU, recSamplesV) de profundidad de bits igual a (lut_bit_depth_minus8+8). Se dice que un vértice lut[i][j][k] pertenece a capa layer_id si los valores de i%(nbp>>layer_id), j%(nbp>>layer_id), k%(nbp>>layer_id) son iguales a cero. Un vértice puede pertenecer a varias capas. Un octante de capa layer_id está compuesto por 8 vértices vecinos pertenecientes a layer_id (figura 14).

55 La descodificación del octante(layer_id, y, u, v) es una función recursiva. Cada octante está compuesto por 8 vértices (i=0,... 7) asociados con un indicador (encoded_flag[i]) que indica si los valores de componentes residuales (resY[i], resU[i], resV[i]) se codifican o se infiere que todos son cero. Los valores de componentes se reconstruyen añadiendo los residuos a la predicción de los valores de componentes. La predicción de los valores de componentes se computa usando interpolación trilineal de los 8 vértices vecinos de la layer_id-1. Una vez reconstruido, un vértice se marca como reconstruido.

60 Donde (y+dy[i]), (u+du[i]) y (v+dv[i]) son las 8 coordenadas de octantes hijos (coordenadas del primer vértice de color 3D) del octante actual (que tiene (y,u,v) como coordenadas del primer vértice (i=0)). Los valores dy[i], du[i] y dv[i] para una capa dada se representan en la tabla 9.

Tabla 8 - primarios de color (color_primaries_input_id y color_primaries_output_id)

Valor	Primarios			Comentario informativo
0	Reservado			Para uso futuro por ITU-T ISO/IEC
1	primario verde azul rojo blanco D65	x 0.300 0.150 0.640 0.3127	y 0.600 0.060 0.330 0.3290	ITU-R Rec. BT.709-5 ITU-R Rec. BT.1361 sistema de gama de color convencional y sistema de gama de color extendida IEC 61966-2-1 (sRGB o sYCC) IEC 61966-2-4 Sociedad de imagen en movimiento e ingenieros de televisión RP 177 (1993) Anexo B
2	Sin especificar			Se desconocen características de imagen o son determinadas por la aplicación
3	Reservado			Para uso futuro por ITU-T ISO/IEC
4	primario verde azul rojo blanco C	x 0.21 0.14 0.67 0.310	y 0.71 0.08 0.33 0.316	ITU-R Rec. BT.470-6 Sistema M (histórico) Recomendación de 1953 del comité del sistema de televisión nacional de Estados Unidos para normas de transmisión para televisión en color comisión de comunicaciones federal de Estados Unidos título 47 código de regulaciones federal (2003) 73.682 (a) (20)
5	primario verde azul rojo blanco D65	x 0.29 0.15 0.64 0.3127	y 0.60 0.06 0.33 0.3290	ITU-R Rec. BT.470-6 Sistema B, G (histórico) ITU-R Rec. BT.601-6 625 ITU-R Rec. BT.1358 625 ITU-R Rec. BT.1700 625 PAL y 625 SECAM

(continúa)

Valor	Primarios			Comentario informativo
6	primario	x	y	ITU-R Rec. BT.601-6 525
	verde	0.310	0.595	ITU-R Rec. BT.1358 525
	azul	0.155	0.070	ITU-R Rec. BT.1700 NTSC
	rojo	0.630	0.340	Sociedad de imagen en movimiento e ingenieros de televisión 170M (2004)
	blanco D65	0.3127	0.3290	(funcionalmente la misma que el valor 7)
7	primario	x	y	Sociedad de imagen en movimiento e ingenieros de televisión 240M (1999) (funcionalmente la misma que el valor 6)
	verde	0.310	0.595	
	azul	0.155	0.070	
	rojo	0.630	0.340	
	blanco D65	0.3127	0.3290	
8	primario	x	y	Película genérica (filtros de color que usan iluminante C)
	verde (Wratten 58)	0.243	0.692	
	azul (Wratten 47)	0.145	0.049	
	rojo (Wratten 25)	0.681	0.319	
	blanco C	0.310	0.316	
9	primario	x	y	Rec. ITU-R BT.2020
	verde	0.170	0.797	
	azul	0.131	0.046	
	rojo	0.708	0.292	
	blanco D65	0.3127	0.3290	
10.. 255	Reservado			Para uso futuro por ITU-T ISO/IEC

Tabla 9: valores $dy[i]$, $du[i]$ y $dv[i]$ en función del índice i , para vértices que pertenecen a capa = layer_id.

i	$dy[i]$	$du[i]$	$dv[i]$
0	0	0	0
1	0	0	$nbp \gg layer_id$
2	0	$nbp \gg layer_id$	0
3	0	$nbp \gg layer_id$	$nbp \gg layer_id$
4	$nbp \gg layer_id$	0	0
5	$nbp \gg layer_id$	0	$nbp \gg layer_id$
6	$nbp \gg layer_id$	$nbp \gg layer_id$	0
7	$nbp \gg layer_id$	$nbp \gg layer_id$	$nbp \gg layer_id$

5

Las muestras de LUT de color 3D reconstruidas ($recSamplesY[i]$, $recSamplesU[i]$, $recSamplesV[i]$) para el vértice $((y+dy[i]), (u+du[i]), (v+dv[i]))$ que pertenece a un octante de la capa = layer_id viene dada por:

$$recSamplesY[i] = resY[i] + predSamplesY[i]$$

10

donde el valor de $predSampleY[i]$ se deriva usando interpolación trilineal con los vértices del octante de capa=layer_id-1 que contiene el octante actual.

15

La figura 12 representa una arquitectura ejemplar de un codificador 1. El codificador está configurado para ejecutar los pasos del método de codificación. El codificador 1 comprende siguientes elementos que están enlazados entre sí por un bus 64 de datos y direcciones:

- un microprocesador 61 (o CPU), que es, por ejemplo, un DSP (o procesador de señal digital);

- una ROM (o memoria de sólo lectura) 62;

5 - una RAM (o memoria de acceso aleatorio) 63;

- uno o varios dispositivos de E/S (entrada/salida) 65 tales como por ejemplo un teclado, un ratón, una cámara web; y

10 - una fuente de alimentación 66.

De acuerdo con una variante, la fuente de alimentación 66 es externa al codificador. Cada uno de estos elementos de la figura 12 es bien conocido por las personas expertas en la técnica y no se divulgará adicionalmente. En cada memoria mencionada, la palabra "registro" usada en la memoria descriptiva designa en
15 cada una de las memorias mencionadas, tanto una zona de memoria de baja capacidad (algunos datos binarios) como una zona de memoria de gran capacidad (que permite almacenar un programa completo o todos o parte de los datos representativos de los datos computados o a visualizar). La ROM 62 comprende un programa y parámetros de codificación (tales como el umbral TH). El algoritmo del método de codificación de acuerdo con la invención está almacenado en la ROM 62. Cuando se enciende, la CPU 61 carga el programa 620 en la RAM y
20 ejecuta las instrucciones correspondientes.

La RAM 63 comprende, en un registro, el programa ejecutado por la CPU 61 y cargado después de encender el codificador 1, datos de entrada en un registro, datos codificados en diferente estado del método de codificación en un registro y otras variables usadas para codificación en un registro.

25 La figura 13 representa una arquitectura ejemplar de un decodificador 2. El decodificador está configurado para ejecutar los pasos del método de decodificación. El decodificador 2 comprende los siguientes elementos que están enlazados entre sí por un bus 74 de datos y direcciones:

30 - un microprocesador 71 (o CPU), que es, por ejemplo, un DSP (o procesador de señal digital);

- una ROM (o memoria de sólo lectura) 72;

35 - una RAM (o memoria de acceso aleatorio) 73;

- una interfaz de E/S 75 para recepción de datos a transmitir, desde una aplicación; y

- una batería 76.

40 De acuerdo con una variante, la batería 76 es externa al codificador. Cada uno de estos elementos de la figura 13 es bien conocido por las personas expertas en la técnica y no se divulgará adicionalmente. En cada memoria mencionada, la palabra "registro" usada en la memoria descriptiva puede corresponder a un área de pequeña capacidad (algunos bits) o a un área muy grande (por ejemplo, un programa completo o una gran cantidad de datos recibidos o decodificados). La ROM 72 comprende al menos un programa y parámetros de
45 decodificador. El algoritmo del método de decodificación de acuerdo con la invención se almacena en la ROM 72. Cuando se enciende, la CPU 71 carga el programa 720 en la RAM y ejecuta las correspondientes instrucciones.

La RAM 73 comprende, en un registro, el programa ejecutado por la CPU 71 y cargado después de encender el
50 decodificador 2, datos de entrada en un registro, datos decodificados en diferentes estados del método de decodificación en un registro, y otras variables usadas para decodificación en un registro.

Las implementaciones descritas en el presente documento pueden implementarse, por ejemplo, en un método o un proceso, un aparato, un programa de software, un flujo de datos o una señal. Incluso si solo se analiza en el
55 contexto de una única forma de implementación (por ejemplo, se analiza solo como un método o un dispositivo), la implementación de características analizadas también puede implementarse en otras formas (por ejemplo, un programa). Un aparato puede implementarse, por ejemplo, en hardware, software y firmware apropiados. Los métodos pueden implementarse, por ejemplo, en un aparato tal como, por ejemplo, un procesador, que se refiere a dispositivos de procesamiento en general, incluyendo, por ejemplo, un ordenador, un microprocesador, un
60 circuito integrado o un dispositivo lógico programable. Los procesadores también incluyen dispositivos de comunicación, tales como, por ejemplo, ordenadores, teléfonos móviles, asistentes digitales portátiles/personales ("PDA"), y otros dispositivos que facilitan comunicación de información entre usuarios finales.

Implementaciones de los diversos procesos y características descritos en el presente documento pueden
65 realizarse en una variedad de equipos o aplicaciones diferentes, particularmente, por ejemplo, equipos o aplicaciones. Ejemplos de tal equipo incluyen un codificador, un decodificador, una salida de procesamiento al

postprocesador desde un decodificador, un preprocesador que proporciona entrada a un codificador, un codificador de vídeo, un decodificador de vídeo, un códec de vídeo, un servidor web, un convertidor, un ordenador portátil, un ordenador personal, un teléfono móvil, una PDA, y otros dispositivos de comunicación. Como debe quedar claro, el equipo puede ser móvil e incluso instalarse en un vehículo móvil.

5

Adicionalmente, los métodos pueden implementarse mediante instrucciones que se realizan por un procesador, y tales instrucciones (y/o valores de datos producidos por una implementación) pueden almacenarse en un medio legible por procesador tal como, por ejemplo, un circuito integrado, un soporte de software u otro dispositivo de almacenamiento tal como, por ejemplo, un disco duro, un disquete compacto ("CD"), un disco óptico (tal como, por ejemplo, un DVD, a menudo denominado disco versátil digital o disco de vídeo digital), una memoria de acceso aleatorio ("RAM") o una memoria de solo lectura ("ROM"). Las instrucciones pueden formar un programa de aplicación materializado de manera tangible en un medio legible por procesador. Instrucciones pueden ser, por ejemplo, hardware, firmware, software o una combinación. Instrucciones pueden encontrarse, por ejemplo, en un sistema operativo, una aplicación separada, o una combinación de los dos. Un procesador puede caracterizarse, por lo tanto, como, por ejemplo, tanto un dispositivo configurado para llevar a cabo un proceso como un dispositivo que incluye un medio legible por procesador (tal como un dispositivo de almacenamiento) que tiene instrucciones para llevar a cabo un proceso. Además, un medio legible por procesador puede almacenar, además de o en lugar de instrucciones, valores de datos producidos por una implementación.

10

15

20

25

30

Como será evidente para una persona experta en la técnica, implementaciones pueden producir una variedad de señales formateadas para transportar información que puede, por ejemplo, almacenarse o transmitirse. La información puede incluir, por ejemplo, instrucciones para realizar un método, o datos producidos por una de las implementaciones descritas. Por ejemplo, una señal puede formatearse para transportar como datos las reglas de escritura o lectura de la sintaxis de una realización descrita, o para transportar como datos los valores de sintaxis reales escritos por una realización descrita. Tal señal puede formatearse, por ejemplo, como una onda electromagnética (por ejemplo, usando una porción de radiofrecuencia del espectro) o como una señal de banda de base. El formateo puede incluir, por ejemplo, codificar un flujo de datos y modular una portadora con el flujo de datos codificado. La información que transporta la señal puede ser, por ejemplo, información analógica o digital. La señal puede transmitirse a través de una variedad de enlaces cableados o inalámbricos diferentes, como se conoce. La señal puede almacenarse en un medio legible por procesador.

35

40

Se han descrito un número de implementaciones. No obstante, se entenderá que se pueden hacer diversas modificaciones. Por ejemplo, elementos de diferentes implementaciones pueden combinarse, suplementarse, modificarse o eliminarse para producir otras implementaciones. Adicionalmente, una persona experta entenderá que otras estructuras y procesos pueden sustituirse por aquellas divulgadas y las implementaciones resultantes realizarán al menos sustancialmente la misma función o funciones, en al menos sustancialmente la misma forma o formas, para lograr al menos sustancialmente el mismo resultado o resultados que las implementaciones divulgadas. Por consiguiente, estas y otras implementaciones se contemplan en esta solicitud. Específicamente, el método para codificar una LUT de acuerdo con la invención puede usarse para codificar cualquier tipo de LUT cualquiera que sea el tipo de valores asociados con cada vértice.

REIVINDICACIONES

1. Un método para codificar una tabla de consulta de color tridimensional definida como una retícula de vértices que usa un árbol octal, en el que cada vértice está asociado con un valor representativo de una transformada de color, que comprende:
5 codificar un indicador de división para un octante actual de dicho árbol octal que indica si al menos uno de sus octantes hijos inmediatos debe codificarse recursivamente;
- 10 predecir al menos un valor de un vértice en dicho octante actual a partir de una reconstrucción de al menos un valor de al menos un vértice en un octante padre para obtener una predicción;
determinar un residuo computado entre el al menos un valor del vértice y la predicción;
- 15 codificar dicho residuo; y
codificar los octantes hijos inmediatos de dicho octante actual si dicho indicador de división indica que al menos uno de sus octantes hijos inmediatos debe codificarse recursivamente;
- 20 en el que dicho indicador de división para el octante actual se establece para indicar que al menos uno de los octantes hijos inmediatos debe codificarse recursivamente si al menos un vértice a codificar de un octante hijo inmediato tiene al menos un residuo que es mayor que un umbral.
2. El método de la reivindicación 1, en el que codificar dicho residuo comprende:
25 cuantificar el residuo para obtener un residuo cuantificado; y
codificar por entropía el residuo cuantificado.
- 30 3. El método de acuerdo con la reivindicación 1, que comprende además codificar un indicador para dicho vértice que indica si al menos un residuo está o no codificado para dicho vértice.
4. Un método para descodificar una tabla de consulta de color tridimensional definida como una retícula de vértices que usa un árbol octal, en el que cada vértice está asociado con un valor representativo de una transformada de color, que comprende:
35 descodificar un indicador de división para un octante actual de dicho árbol octal que indica si al menos uno de sus octantes hijos inmediatos debe descodificarse recursivamente;
- 40 predecir al menos un valor de un vértice en dicho octante actual a partir de una reconstrucción de al menos un valor de al menos un vértice en un octante padre para obtener una predicción;
descodificar un residuo;
- 45 reconstruir el al menos un valor del vértice en dicho octante actual a partir de la predicción y el residuo; y
descodificar los octantes hijos inmediatos de dicho octante actual si dicho indicador de división indica que al menos uno de sus octantes hijos inmediatos debe descodificarse recursivamente;
- 50 en el que dicho indicador de división para el octante actual indica que al menos uno de los octantes hijos inmediatos debe descodificarse recursivamente si al menos un vértice a descodificar de un octante hijo inmediato tiene al menos un residuo que es mayor que un umbral.
5. El método de la reivindicación 4, en el que descodificar dicho residuo comprende:
55 descodificar por entropía un residuo cuantificado codificado por entropía para obtener un residuo cuantificado; y
cuantificar inversamente el residuo cuantificado para obtener el residuo.
- 60 6. El método de acuerdo con la reivindicación 4, que comprende además descodificar un indicador para dicho vértice que indica si al menos un residuo está o no descodificado para dicho vértice.
7. Un codificador para codificar una tabla de consulta de color tridimensional definida como una retícula de vértices que usa un árbol octal, en el que cada vértice está asociado con un valor representativo de una transformada de color, comprendiendo dicho codificador una interfaz para acceder a dicha tabla de consulta de color tridimensional y al menos un procesador configurado para realizar el método de una cualquiera de las
65

reivindicaciones 1 a 3.

5 8. Un decodificador para decodificar una tabla de consulta de color tridimensional definida como una retícula de vértices que usa un árbol octal, comprendiendo dicho decodificador una interfaz para acceder a un flujo de bits y al menos un procesador configurado para realizar el método de una cualquiera de las reivindicaciones 4 a 6.

9. Un flujo de bits obtenido por el método de una cualquiera de las reivindicaciones 1 a 3.

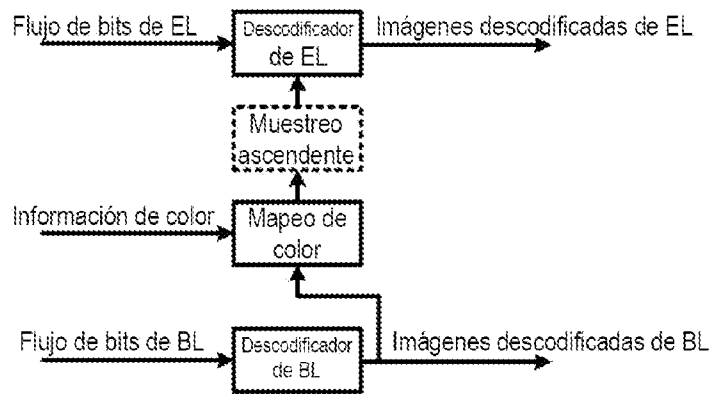


FIGURA 1 – Técnica anterior

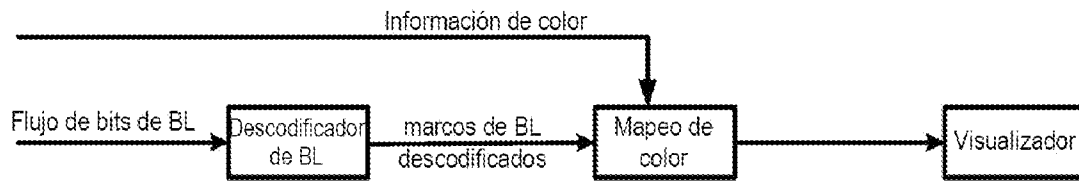


FIGURA 2 – Técnica anterior

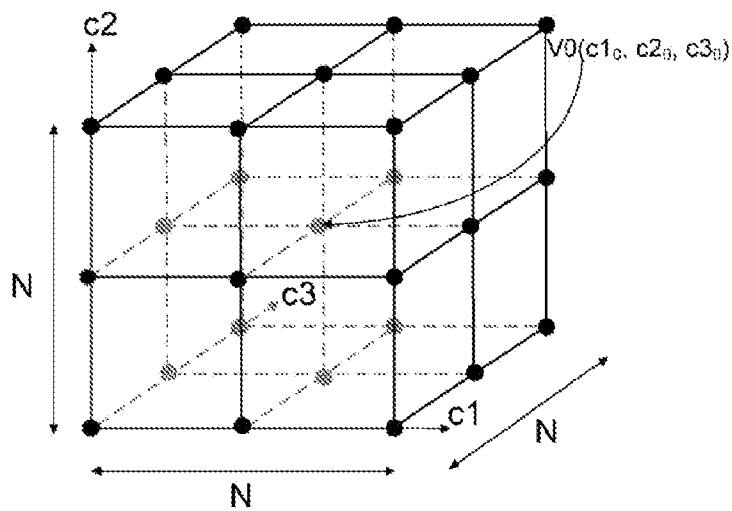


FIGURA 3 – Técnica anterior

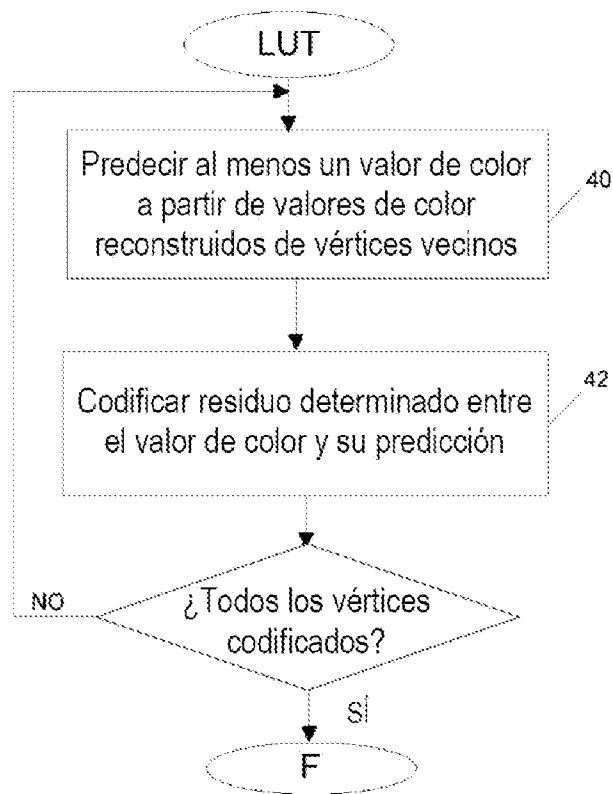


FIGURA 4

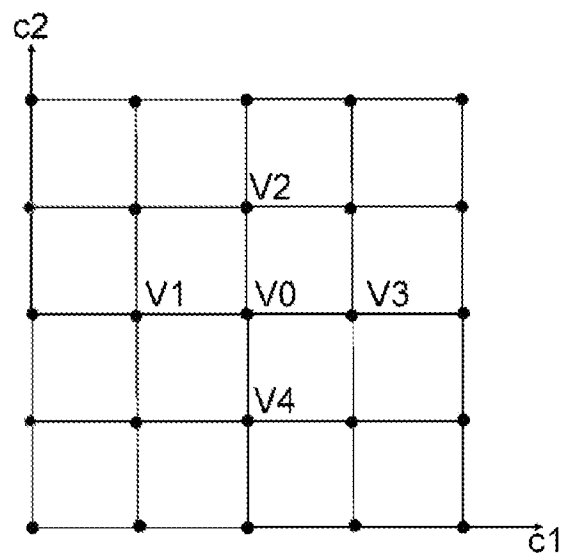


FIGURA 5

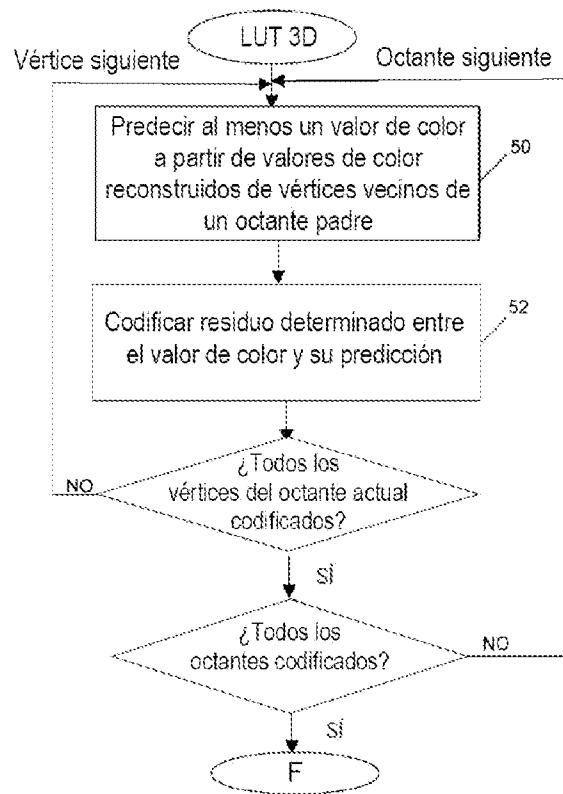


FIGURA 6

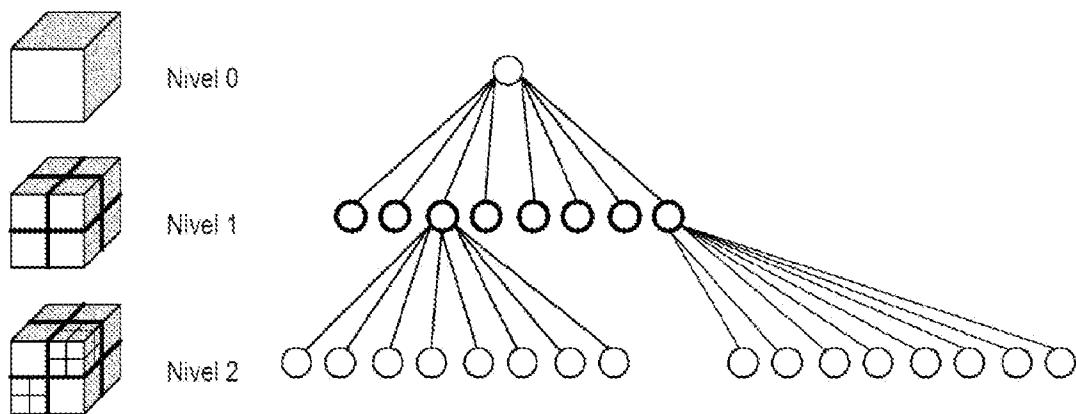


FIGURA 7

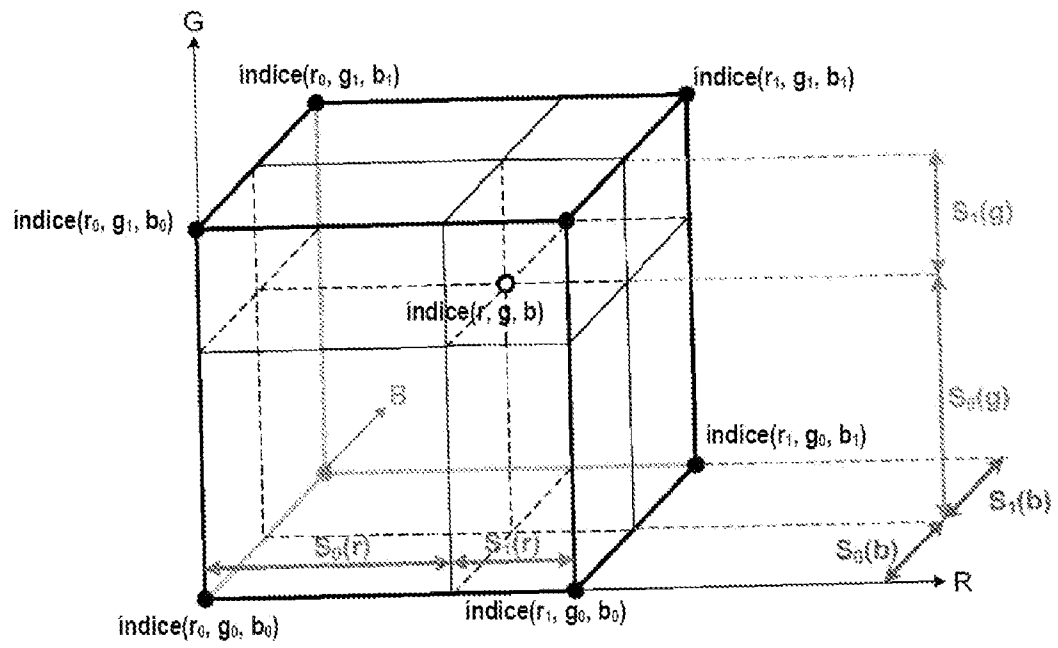


FIGURA 8

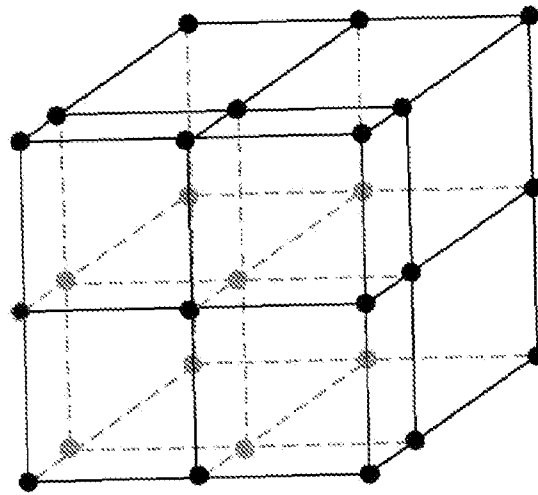


FIGURA 9

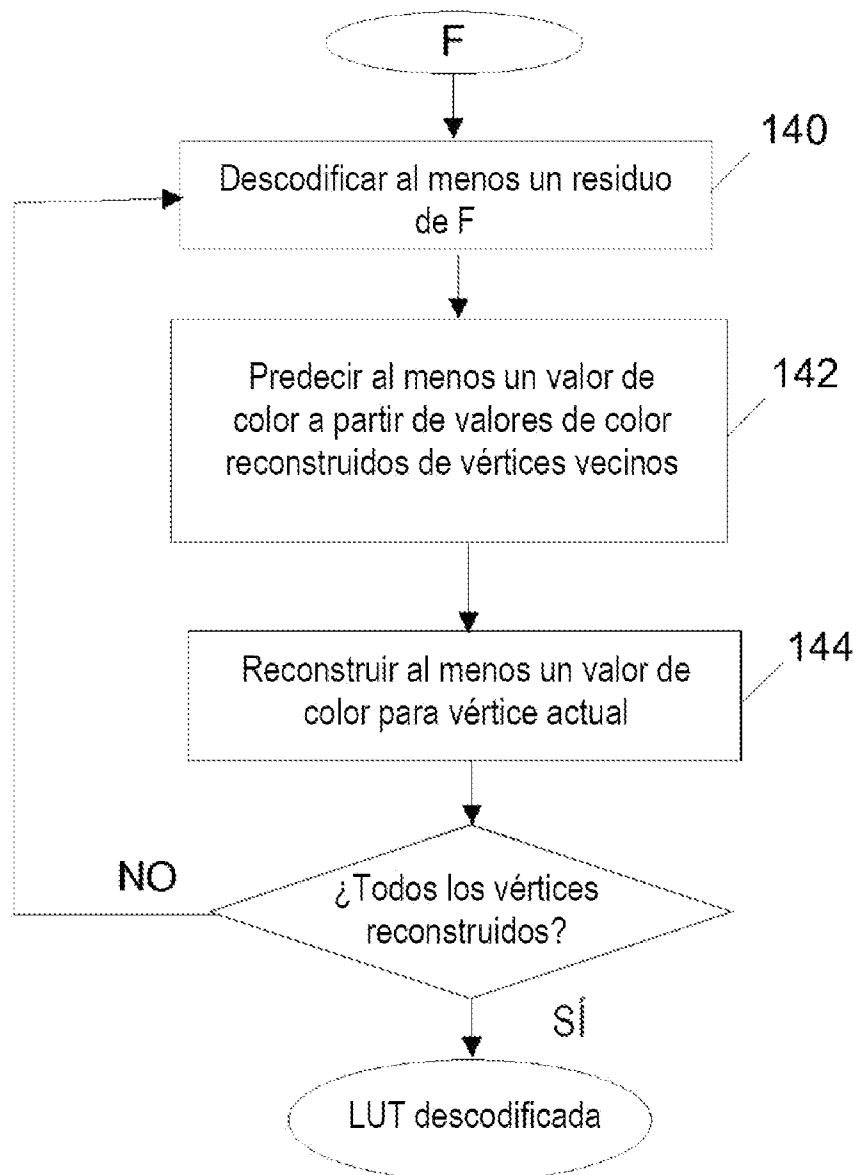


FIGURA 10

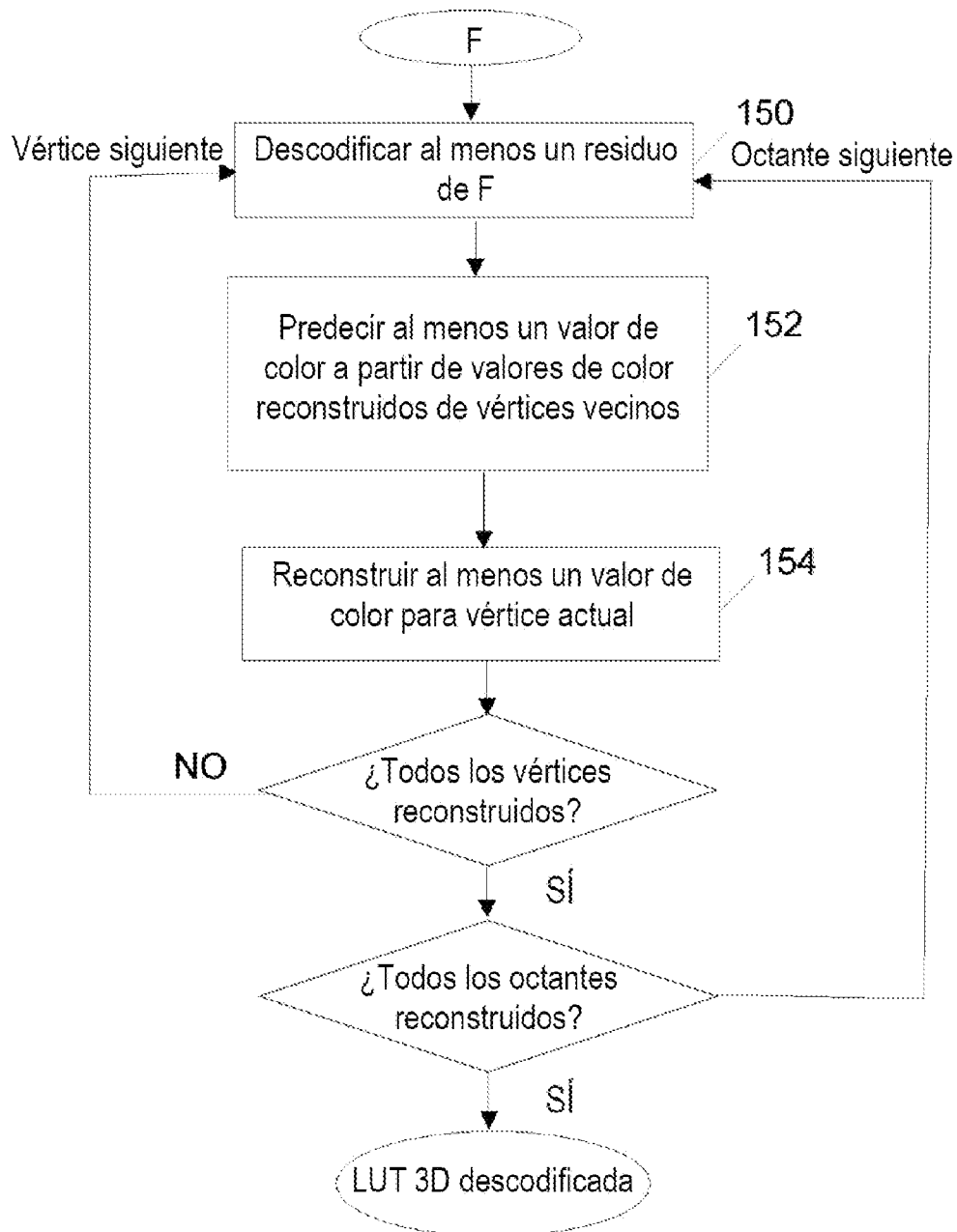


FIGURA 11

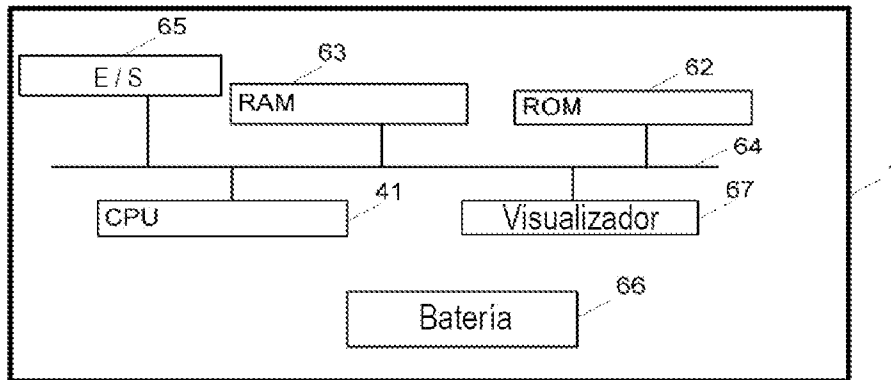


FIGURA 12

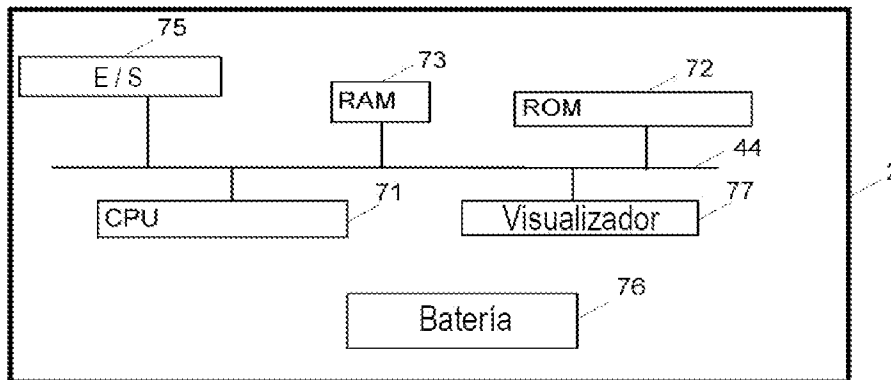


FIGURA 13

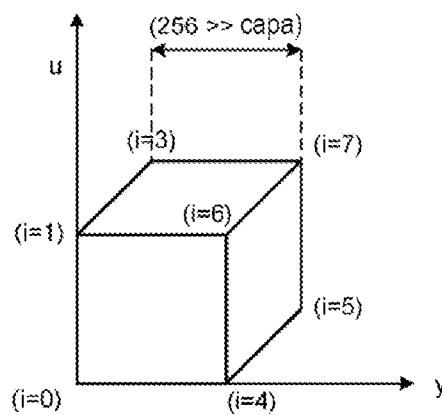


FIGURA 14