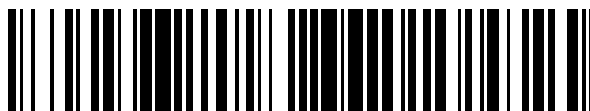


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 666 899**

51 Int. Cl.:

**H04N 19/30** (2014.01)

**H04N 19/136** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **25.03.2014** **PCT/US2014/031716**

87 Fecha y número de publicación internacional: **02.10.2014** **WO14160705**

96 Fecha de presentación y número de la solicitud europea: **25.03.2014** **E 14722483 (6)**

97 Fecha y número de publicación de la concesión europea: **14.03.2018** **EP 2936814**

54 Título: **Codificación de contenido de vídeo perceptualmente-cuantizado en codificación VDR multicapa**

30 Prioridad:

**26.03.2013 US 201361805388 P**  
**02.08.2013 US 201361861555 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**08.05.2018**

73 Titular/es:

**DOLBY LABORATORIES LICENSING CORPORATION (100.0%)**  
**100 Potrero Avenue**  
**San Francisco, CA 94103-4813, US**

72 Inventor/es:

**SU, GUAN-MING;**  
**CHEN, QIAN;**  
**KOEPPFER, HUBERT y**  
**QU, SHENG**

74 Agente/Representante:

**LEHMANN NOVO, María Isabel**

ES 2 666 899 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

Codificación de contenido de vídeo perceptualmente-cuantizado en codificación VDR multicapa

### 5 TECNOLOGÍAS

La presente invención se refiere, en general, al procesamiento de imágenes y, en particular, a la codificación, decodificación y representación de vídeo perceptualmente cuantizado.

### 10 ANTECEDENTES DE LA INVENCION

Se pueden utilizar capas múltiples para entregar datos de vídeo desde un dispositivo en flujo ascendente, tal como un codificador de vídeo de capas múltiples para dispositivos en flujo descendente. A modo de ejemplo, los datos de vídeo de margen dinámico visual (VDR) se incluyen en la combinación de una capa base y una capa mejorada (EL) de las capas múltiples para la visualización de la experiencia de las pantallas VDR.

Los datos de imagen BL pueden codificarse con datos de imagen de margen dinámico de nivel bajo o parcial derivados de imágenes VDR. En los datos de imagen de margen dinámico de nivel bajo o parcial, las palabras de código fuera de límite mapeados a partir de las imágenes VDR se pueden recortar en un margen representado objetivo. Las variaciones textuales y los detalles de imágenes, representados en las imágenes VDR, pueden perderse en los datos de imagen de margen dinámico de nivel bajo o parcial.

Los métodos descritos en esta sección son métodos que se podrían poner en práctica, pero no necesariamente métodos que hayan sido concebidos o realizados con anterioridad. Por lo tanto, a menos que se indique de otro modo, no se debe suponer que cualquiera de los métodos descritos en esta sección se califican como siendo de la técnica anterior simplemente en virtud de su inclusión en esta sección. De modo similar, las cuestiones identificadas con respecto a uno o más métodos no deben suponerse que han sido reconocidas en cualquier técnica anterior sobre la base de esta sección, a no ser que se indique lo contrario.

El documento WO 2010/105036 A1 da a conocer una compresión en capas de un vídeo de gama de amplios colores y alto margen dinámico, en donde se codifica un flujo dinámico basado en un flujo de margen dinámico de nivel bajo correspondiente. Un decodificador deriva el vídeo de gama de amplios colores y alto margen dinámico mediante el procesamiento del flujo de margen dinámico de nivel bajo y parámetros de mapeado de correspondencia de tonos.

### 35 SUMARIO

La invención se define por las reivindicaciones independientes, teniendo en cuenta debidamente cualquier elemento que sea equivalente a un elemento especificado en las reivindicaciones. Las reivindicaciones subordinadas se refieren a características opcionales de algunas formas de realización de la invención.

### 40 BREVE DESCRIPCIÓN DE LOS DIBUJOS

La presente invención se ilustra a modo de ejemplo, y no a modo de limitación, en las figuras de los dibujos adjuntos y en donde los mismos números de referencia se refieren a elementos similares, y en los que:

La Figura 1A ilustra un codificador de vídeo multicapa, de conformidad con una forma de realización de la invención;

La Figura 1B ilustra un decodificador de vídeo multicapa, de conformidad con una forma de realización de la invención;

La Figura 1C y la Figura 1D ilustran puestas en práctica de punto fijo de una unidad de componedor en un decodificador realizado de conformidad con las formas de realización de la invención.

La Figura 2A y la Figura 2B ilustran algoritmos, a modo de ejemplo, para buscar valores de parámetros de mapeado óptimos, de conformidad con una forma de realización de la invención;

La Figura 3A y la Figura 3B ilustran flujos de proceso, a modo de ejemplo, de conformidad con una forma de realización de la invención; y

La Figura 4 ilustra una plataforma de hardware, a modo de ejemplo, en la que un ordenador o un dispositivo informático pueden ponerse en práctica tal como se describe en este documento.

### DESCRIPCIÓN DETALLADA DE FORMAS DE REALIZACIÓN EJEMPLO

Las formas de realización, a modo de ejemplo, que se refieren a codificación, decodificación y representación de imágenes de margen dinámico variable, que incluyen datos de imagen de capa base perceptualmente cuantizados,

utilizando un códec multicapa, se describen en la presente memoria. En la siguiente descripción, para los fines de explicación, se exponen numerosos detalles específicos para proporcionar un entendimiento completo de la presente invención. Sin embargo, será evidente que la presente invención se puede poner en práctica sin estos detalles específicos. En otros casos, estructuras y dispositivos, bien conocidos, no se describen con detalles exhaustivos, con el fin no bloquear, oscurecer u ofuscar, de forma innecesaria, la presente invención.

Las formas de realización a modo de ejemplo se describen en este documento de conformidad con el siguiente esquema:

## 1. RESUMEN GENERAL

## 2. CODIFICACIÓN Y DECODIFICACIÓN DE CONTENIDO DE VÍDEO

## 3. ASIGNACIÓN DE PALABRAS DE CÓDIGO DISPONIBLES BASADAS EN EL CONTENIDO DE LA IMAGEN

## 4. CODIFICACIÓN DE VÍDEO MULTICAPA

## 5. DECODIFICACIÓN DE VÍDEO MULTICAPA

## 6. CONSERVACIÓN DE DETALLES VISUALES EN DATOS DE IMAGEN BL

## 7. DETERMINACIÓN EJEMPLO DEL VALOR DEL EXPONENTE EN LA FUNCIÓN DE POTENCIA

## 8. DETERMINACIÓN EJEMPLO DE VALORES DE PARÁMETROS EN CUANTIZACIÓN LINEAL, INCLUYENDO LA CUANTIZACIÓN LINEAL POR ELEMENTOS

## 9. EJEMPLO DE FLUJOS DE PROCESO

## 10. MECANISMOS DE PUESTA EN PRÁCTICA-DESCRIPCIÓN GENERAL DEL HARDWARE

## 11. EQUIVALENTES, EXTENSIONES, ALTERNATIVAS Y DISPOSICIONES VARIAS

### 1. RESUMEN GENERAL

Este resumen presenta una descripción básica de algunos aspectos de una forma de realización, a modo de ejemplo, de la presente invención. Conviene señalar que este resumen no es un resumen extenso o exhaustivo de aspectos de la forma de realización ejemplo. Además, debe observarse que este resumen no pretende interpretarse como identificando cualquier aspecto o elemento particularmente significativo de la forma de realización ejemplo, ni como delineamiento de ningún alcance de la forma de realización ejemplo en particular, ni de la invención en general. Este resumen simplemente presenta algunos conceptos que se relacionan con la forma de realización ejemplo en un formato condensado y simplificado, y debe entenderse como un mero preámbulo conceptual de una descripción más detallada de las formas de realización ejemplo que se detallan a continuación.

En algunas formas de realización, los códecs multicapa se pueden utilizar para generar o procesar una señal multimedia que comprende imágenes comprimidas (p.ej., imágenes de vídeo) para una amplia diversidad de pantallas (p.ej., pantallas VDR, etc.). Con el fin de proporcionar una calidad visual superior en una amplia variedad de pantallas de margen dinámico amplio y estrecho, el contenido de vídeo que se proporcionará a dispositivos de flujo descendente se puede cuantizar y codificar mediante un códec multicapa que pone en práctica técnicas de cuantización perceptual (PQ) tal como aquí se describe. Dicho contenido de vídeo, tal como es presentado por los dispositivos de flujo descendente, muestra características visuales espectacularmente diferentes de las mostradas por otras técnicas.

Tal como aquí se utiliza, el término "códec multicapa" se puede referir a un codificador o decodificador multicapa que pone en práctica una estructura de capas múltiples en una señal audiovisual (p.ej., un flujo de bits, una señal de difusión, un archivo multimedia, etc.) Las capas múltiples comprenden una capa base y una o más capas mejoradas. Las capas base y mejoradas comprenden datos de imagen que se derivan de las mismas imágenes origen (p.ej., perceptualmente codificada, etc.). Los datos de imagen en la capa base contienen datos de imagen comprimida de un margen dinámico de nivel bajo o parcial, que pueden no optimizarse para la presentación en pantallas de un margen dinámico relativamente estrecho, tal como un margen dinámico estándar o SDR. Los datos de imagen, en las capas múltiples, en combinación, contienen imágenes comprimidas de un margen dinámico amplio, que se pueden decodificar y visualizar en pantallas de un margen dinámico relativamente amplio, tal como un margen dinámico visual o VDR. El término "VDR" o "margen dinámico visual" tal como se utiliza en este documento, se puede referir a un margen dinámico más amplio que un margen dinámico estándar, y puede incluir, pero no estar limitado a, un margen dinámico amplio, hasta el margen dinámico perceptible de forma instantánea y una gama de color que la visión humana puede percibir en un instante.

Un códec multicapa tal como aquí se describe, puede ponerse en práctica con dos o más códecs de profundidad de bits más baja (p.ej., 8 bits, etc.) (p.ej., códecs de dominio gamma, etc.) en las capas múltiples con el fin de soportar imágenes de mayor profundidad de bits (p. ej., 12+ bits, etc.) en las capas múltiples como un conjunto y para soportar imágenes con menor profundidad de bits (p.ej., 8 bits, etc.) en la capa base. A modo de ejemplo, uno o más

5 codificadores de dominio gamma pueden desarrollarse en una ruta de procesamiento de señal BL y/o en una o más rutas de procesamiento de señal EL de un codificador multicapa, tal como se describe aquí. En algunas formas de realización, al menos un decodificador de dominio gamma se puede desplegar en la ruta de procesamiento de señal BL de un dispositivo de flujo descendente, que es un decodificador de capa base solamente.

10 Las técnicas PQ tal como se describen en la presente memoria, se pueden utilizar para permitir que los códecs de dominio gamma conserven detalles visuales del contenido de vídeo original codificado perceptualmente. En lugar de requerir un convertidor de formato PQ a gamma en el lado del codificador de vídeo y un convertidor de formato gamma a PQ en el lado del decodificador, las técnicas descritas en este documento permiten códecs de dominio gamma en un codificador multicapa, un decodificador multicapa, un decodificador de capa base, etc., para preservar,

15 directamente, los detalles visuales del contenido de vídeo codificado perceptualmente sin dichos convertidores de formato. Las señales de vídeo perceptualmente cuantizado, que conservan estos detalles visuales, a la vez en márgenes dinámicos amplios y estrechos, se pueden proporcionar por un codificador multicapa a dispositivos de flujo descendente, de forma directa, sin utilizar una conversión de formato PQ a gamma. De modo similar, señales de vídeo perceptualmente cuantizado, tal como las recibidas por un dispositivo de flujo descendente, se pueden

20 decodificar y/o reproducir con los detalles visuales conservados por el dispositivo de flujo descendente directamente, sin el uso de una conversión de formato de gamma a PQ. En consecuencia, la complejidad de cálculo, la eficiencia de codificación y la calidad de percepción se pueden mejorar, de forma significativa, bajo las técnicas que se describen en el presente documento.

25 Las técnicas de puesta en práctica de códecs, tal como se describen en la presente memoria, se pueden configurar, además, para incluir capacidades de predicción entre capas para utilizar completamente la redundancia estadística entre datos de imagen de capa base (BL) y datos de imagen de entrada originales. Los datos de imagen EL pueden (sólo posiblemente) transmitir datos de imágenes residuales (o diferenciales), en lugar de incluir una gran cantidad de datos de imagen de margen dinámico amplio, sin aprovechar las correlaciones y la redundancia estadística en

30 datos de imagen de diferentes capas.

En algunas formas de realización a modo de ejemplo, los datos necesarios para otras aplicaciones (incluyendo, pero no limitado a aquellos no relacionados con operaciones de codificación perceptual, etc.) se pueden incluir, además, con datos de imagen de capa base y capa mejorada para proporcionarse desde un dispositivo de flujo ascendente a

35 un dispositivo de flujo descendente. En algunas formas de realización ejemplo, se pueden soportar características adicionales y/o características ortogonales, por las capas base y mejorada, tal como aquí se describe.

En algunas formas de realización ejemplo, mecanismos, como los descritos en este documento, forman una parte de un sistema de procesamiento multimedia, incluyendo, pero no limitándose a cualquiera de entre: un dispositivo portátil, máquina de juego, televisión, ordenador portátil, ordenador portátil de dimensiones reducidas, tableta electrónica, radioteléfono móvil, lector de libros electrónicos, terminal de punto de venta, ordenador de sobremesa, estación de trabajo informática, kiosco informático, o varios otras clases de terminales y unidades de procesamiento multimedia.

40

45 Diversas modificaciones a las formas de realización preferidas y los principios y características genéricos, descritos en este documento, serán evidentes para los expertos en la técnica. Por lo tanto, la idea inventiva dada a conocer no pretende limitarse a las formas de realización ilustradas, sino que debe otorgarse el alcance más amplio compatible con los principios y características aquí descritas.

## 50 2. CODIFICACIÓN Y DECODIFICACIÓN DE CONTENIDO DE VÍDEO

La visión humana puede no percibir una diferencia entre dos niveles de luminancia si los dos niveles de luminancia no son lo suficientemente distintos entre sí. Sin embargo, la visión humana solamente percibe una diferencia si los niveles de luminancia difieren no menos de una diferencia apenas perceptible (JND). Debido a la no linealidad

55 perceptual de la visión humana, las cantidades de JNDs individuales no se dimensionan o escalan, de forma uniforme, en un margen de niveles de luminancia, sino que varían con diferentes niveles de luminancia individuales.

Técnicas (p.ej., algoritmos, etc.) tal como aquí se describen se pueden utilizar por un codificador multicapa para codificar contenido de vídeo origen en una señal de vídeo de capas múltiples que preserva detalles visuales del contenido de vídeo origen en datos de imagen BL, así como en la combinación de datos de imagen BL y EL. En algunas formas de realización, el contenido de vídeo origen se codifica originalmente con las palabras de código origen en un formato de codificación origen (p.ej., según se define mediante una especificación VDR, etc.) sobre la base de técnicas de codificación perceptual (p.ej., técnicas de codificación VDR desarrolladas por Dolby Laboratories, Inc., San Francisco, California, etc.). En algunas formas de realización, las palabras de código origen,

60 en el formato de codificación origen, representan niveles de luminancia distribuidos de tal manera que los niveles de luminancia están espaciados o cuantizados, de forma óptima, para que coincidan con la no linealidad perceptual de

65

la visión humana.

La señal de vídeo multicapa generada por un codificador multicapa, según se describe en el presente documento, puede transmitirse, directa o indirectamente a, o recibirse por una amplia diversidad de dispositivos de flujo descendente que incluyen, pero no se limitan a, cualquiera de entre: sistemas de visualización con decodificadores multicapa, sistemas de visualización con decodificadores de capa base, etc.

Un dispositivo de flujo descendente (p.ej., un decodificador multicapa, etc.), que admite operaciones de visualización de margen dinámico relativamente amplio que puede derivar y/o representar una versión de margen dinámico amplio del contenido de vídeo origen sobre la base de la señal de vídeo multicapa recibida. La versión de margen dinámico amplio decodificada del contenido de vídeo origen representa niveles de luminancia objetivo (p.ej., niveles de luminancia específicos del dispositivo, etc.) que se aproximan a los niveles de luminancia representados por las palabras de código origen, en el contenido de vídeo origen.

Un dispositivo de flujo descendente (p.ej., un decodificador BL, etc.), que soporta operaciones de visualización de margen dinámico relativamente estrecho que puede derivar y/o reproducir una versión de margen dinámico estrecho decodificado, con detalles visuales, del contenido de vídeo origen basado en datos de imagen BL de la señal de vídeo multicapa recibida. La versión del margen dinámico estrecho decodificada del contenido de vídeo origen representa niveles de luminancia objetivo (p.ej., niveles de luminancia específicos del dispositivo, etc.) que son objeto de mapeado a partir de niveles de luminancia representados por las palabras de código origen en el contenido de vídeo origen, aunque dentro de un margen dinámico relativamente estrecho soportado por el dispositivo de flujo descendente.

Mientras que la versión del margen dinámico estrecho decodificada y la versión de margen dinámico amplio decodificada del contenido de vídeo origen representan, ambas, niveles de luminancia objetivo (p.ej., niveles de luminancia específicos del dispositivo, etc.) que son objeto de mapeado a partir de los niveles de luminancia representados por las palabras de código origen, en el contenido de vídeo origen, la versión de margen dinámico estrecho decodificada del contenido de vídeo origen puede comprender más errores (p.ej., errores de cuantización debido a una profundidad de bits relativamente baja, recortes a niveles de luminancia bajos o altos, etc.) que la versión de margen dinámico amplio decodificada, del contenido de vídeo de origen, en comparación con los niveles de luminancia representados por las palabras de código origen en el contenido de vídeo origen.

### 3. ASIGNACIÓN DE PALABRAS DE CÓDIGO DISPONIBLES BASADAS EN EL CONTENIDO DE LA IMAGEN

Un códec de vídeo, sin utilizar técnicas tal como aquí se describen, asigna palabras de código disponibles a una profundidad de bits particular (p.ej., 8 bits, etc.) de un modo que no conserva detalles visuales de contenido de vídeo origen inicialmente codificado de forma perceptual. A modo de ejemplo, un dispositivo multimedia que comprende un códec de dominio gamma, sin utilizar técnicas aquí descritas, puede sobre-asignar palabras de código a sub-márgenes de alta luminancia (p.ej., elementos brillantes, iluminaciones, etc.) en un margen dinámico específico del dispositivo y asignar menos palabras de código a un sub-margen de luminancia baja (p.ej., elementos oscuros, áreas oscuras, etc.) en el margen dinámico específico del dispositivo. Como resultado, detalles visuales del contenido de vídeo origen codificado perceptualmente se pierden, de forma innecesaria, utilizando estas otras técnicas.

Un códec de vídeo en un códec multicapa, con las técnicas aquí descritas, asigna palabras de código disponibles a una profundidad de bits particular (p.ej., 8 bits, etc.) de un modo que conserva detalles visuales de contenido de vídeo origen codificado perceptualmente, en comparación con un códec de vídeo sin el uso de estas técnicas. Como resultado, los detalles de percepción, en el contenido de vídeo origen se conservan mejor en el códec multicapa que se describe en este documento, que en otros códecs de vídeo sin el uso de estas técnicas.

Con las técnicas aquí descritas, un códec multicapa selecciona/determina valores de parámetros específicos (p.ej., valores de exponente en funciones de potencia, gradientes en cuantización lineal, pivotes en cuantización lineal por segmentos, etc.) en función del contenido de imagen transmitido por tramas de imagen (p.ej., en una escena, etc.) en contenido de vídeo origen. Si el contenido de imagen comprende más elementos resaltados o más niveles de luminancia, los valores de los parámetros pueden seleccionarse/determinarse para hacer que estén disponibles más niveles de luminancia representados en el sub-margen de alta luminancia para las operaciones de codificación o decodificación. Si el contenido de la imagen comprende menos elementos resaltados o menos niveles de luminancia, los valores de parámetros se pueden seleccionar/determinar para provocar que haya menos niveles de luminancia representados en el sub-margen de alta luminancia disponible para las operaciones de codificación o decodificación. De modo similar, si el contenido de la imagen comprende más elementos oscuros o más niveles de luminancia, el parámetro puede seleccionarse/determinarse para hacer que estén disponibles más niveles de luminancia representados en el sub-margen de baja luminancia para las operaciones de codificación o decodificación. Si el contenido de imagen comprende menos elementos oscuros o menos niveles de luminancia, los valores de parámetros pueden seleccionarse/determinarse para hacer que haya menos niveles de luminancia representados en el sub-margen de baja luminancia disponibles para las operaciones de codificación o decodificación.

## 4. CODIFICACIÓN DE VÍDEO MULTICAPA

Tal como se indicó con anterioridad, una señal de vídeo multicapa (p.ej., flujo de bits codificado, etc.), que comprende una capa base y una o más capas mejoradas, se puede utilizar por un dispositivo de flujo ascendente (p.ej., un codificador multicapa 102 de la Figura 1A) para entregar contenido de vídeo codificado a dispositivos de flujo descendente (uno de los cuales puede ser, a modo de ejemplo, un decodificador multicapa 152 de la Figura 1B, etc.). En algunas formas de realización, el contenido de vídeo proporcionado por las capas múltiples comprende datos de imagen BL (p.ej., 106 de la Figura 1A, Figura 1B, etc.) de una profundidad de bits relativamente baja y datos de imagen EL (p.ej., 108 de la Figura 1A, y la Figura 1B, etc.) que son complementarios para los datos de imagen BL. Tanto los datos de imagen BL (106) como los datos de imagen EL (108) se derivan/cuantizan a partir de la profundidad de bit relativamente alta (p.ej., VDR de 12+ bits, etc.) del contenido de vídeo origen (104).

En algunas formas de realización, los datos de imagen BL (106) se colocan en un contenedor de capa base (p.ej., un contenedor YCbCr 4:2:0 de 8 bits, etc.). En algunas formas de realización, los datos de imagen EL (108) comprenden datos de imagen residual (p.ej., VDR, etc.) del contenido de vídeo origen (104) con relación a datos de imágenes previstos, generados a partir de los datos de imágenes BL (106). En algunas formas de realización, los datos de imagen EL (108) se colocan en uno o más contenedores de capa mejorada (p.ej., uno o más contenedores de datos residuales de 8 bits, etc.). Los datos de imágenes BL (106) y los datos de imágenes EL (108) pueden ser recibidos y utilizados por los dispositivos de flujo descendente (p.ej., 152 de la Figura 1B, etc.) para reconstruir una profundidad de bits relativamente alta (p.ej., 12+ bits, etc.) de la versión decodificada (p.ej., una versión de VDR perceptualmente codificada, etc.) del contenido de vídeo origen (p.ej., VDR, etc.) (104).

Tal como se utiliza en este documento, el término "profundidad de bits" se refiere al número de bits proporcionados en un espacio de codificación, que proporciona palabras de código disponibles para codificar o cuantizar datos de imágenes; un ejemplo de baja profundidad de bits es 8 bits; un ejemplo de alta profundidad de bits es 12 bits o más. En particular, el término "una baja profundidad de bits" o "una alta profundidad de bits" no se refieren a los bits menos significativos, o los bits más significativos, de una palabra de código.

En una forma de realización ejemplo, el codificador multicapa (102) está configurado para recibir el contenido de vídeo origen (v.g., VDR, etc.) (104) que comprende una secuencia de imágenes VDR de entrada. La secuencia de imágenes VDR de entrada representa una o más escenas, cada una de las cuales comprende una pluralidad de imágenes de entrada en la secuencia de imágenes VDR. Tal como aquí se utiliza, una "imagen VDR de entrada" se refiere, en general, a datos de imágenes de margen dinámico alto o amplio, que se pueden utilizar para derivar una versión VDR de una imagen origen (p.ej., una imagen referida a escena captada por un dispositivo de adquisición de imágenes de alta gama, etc.) Una imagen VDR de entrada puede estar en cualquier espacio de color que admita una gama de colores de alto margen dinámico. En algunas formas de realización, la imagen VDR de entrada es la única entrada, con respecto a la imagen origen, que proporciona datos de imagen para la codificación por el codificador de imagen VDR (102); datos de imagen de entrada, con respecto a la imagen origen, para el procesamiento de la capa base, según las técnicas descritas en este documento, se pueden generar sobre la base de la imagen de VDR de entrada utilizando técnicas de cuantización perceptual.

En una forma de realización ejemplo, una imagen VDR de entrada decodificada a partir del contenido de vídeo origen (p.ej., VDR, etc.) (104) es una imagen de YCbCr de 12+ bits en un espacio de color YCbCr. En una realización a modo de ejemplo, cada píxel representado en la imagen VDR de entrada comprende palabras de código para todos los canales (p.ej., canal de luminancia Y, canales de croma Cb y Cr, etc.) definidos para un espacio de color (p.ej., un espacio de color YCbCr). Cada palabra de código comprende palabras de código submuestreadas o sobremuestreadas para uno o más de los canales en el espacio de color.

En una forma de realización ejemplo, el codificador multicapa (102) está configurado para transformar palabras de código de una imagen VDR de entrada, a partir de un primer espacio de color (p.ej., un espacio de color RGB, etc.) a un segundo espacio de color diferente (p.ej., un espacio de color YCbCr, etc.).

En una forma de realización ejemplo, el codificador multicapa (102) está configurado para sobre-muestrear o sub-muestrear una imagen VDR de entrada en un primer formato de muestreo (p.ej., en un formato de muestreo 4:4:4, etc.) a un segundo formato de muestreo diferente (p.ej., en un formato de muestreo 4:2:0, etc.).

En una forma de realización ejemplo, el codificador multicapa (102) está configurado para realizar operaciones relacionadas con la cuantización perceptual en el contenido de vídeo origen (104) con el fin de generar los datos de imagen BL (p.ej., 8 bits, etc.) (106). Las operaciones relacionadas con la cuantización perceptual incluyen, pero no se limitan a una o más de entre: operaciones de mapeado (114), operaciones de recorte (116), operaciones de codificación BL (118), etc.

En las operaciones de mapeado (114), el codificador multicapa (102) está configurado para utilizar una función de mapeado (p.ej., 12 bit +, etc.) para efectuar el mapeado de palabras de código VDR de entrada (p.ej., palabras de código de luminancia VDR  $v^Y$ , palabras de código de croma VDR  $v^C$ , etc.) en una o más imágenes VDR de entrada

(p.ej., VDR, etc.) en el contenido de vídeo de origen (104) en palabras de código mapeadas (p.ej., palabras de código de luminancia mapeadas  $c^Y$ , palabras de código de croma mapeadas  $c^C$ , etc.) en una o más imágenes mapeadas que corresponden a las una o más imágenes VDR.

5 En las operaciones de recorte (116), el codificador multicapa (102) está configurado para recortar palabras de código mapeadas que están por debajo de un valor mínimo o por encima de un valor máximo. El número (p.ej.,  $\leq 256$  a una profundidad de bit de 8 bits, etc.) de palabras de código disponibles en la profundidad de bit de los datos de imagen BL (106) puede no ser lo suficientemente grande para tener espacio para el margen numérico completo de las palabras de código mapeadas no recortadas. Por lo tanto, algunas de las palabras de código mapeadas más  
10 altas y/o las palabras de código mapeadas más bajas, en las una o más imágenes mapeadas, se pueden recortar al valor máximo y/o al valor mínimo en la operación de recorte (116). El codificador multicapa (102) está configurado para determinar/seleccionar, sobre la base de las una o más imágenes mapeadas y uno o más factores, valores mínimos y máximos óptimos. Los uno o más factores incluyen, pero no se limitan a: minimizar el número de palabras de código mapeadas que deben recortarse, maximizar los detalles representados/perceptuales de uno o más  
15 elementos (p.ej., saliente, central, etc.) de las una o más imágenes mapeadas después de las operaciones de recorte, etc.

En las operaciones de codificación BL (118), el codificador multicapa (102) está configurado para determinar un valor de exponente óptimo de una función de potencia utilizada por un códec BL en el codificador multicapa (102) para mapear palabras de código origen (por ej.,  $v^Y$ , etc.) a las correspondientes palabras de código mapeadas (p.ej.,  $c^Y$ , etc.) en un canal específico (p.ej., un canal de luminancia o un canal Y, etc.) de un espacio de color (p.ej., un espacio de color YCbCr, etc.) En algunas formas de realización, el codificador multicapa (102) está configurado para determinar valores de parámetros de funciones de mapeado lineal por elementos (PWL) utilizados por el códec BL para mapear palabras de código origen (p.ej.,  $v^C$ , etc.) para las correspondientes palabras de código mapeadas (p.  
25 ej.,  $c^C$ , etc.) en un canal específico (p.ej., un canal croma o un canal Cb/Cr, etc.) de un espacio de color (p. ej., un espacio de color YCbCr, etc.).

En las operaciones de codificación BL (118), el codificador multicapa (102) está configurado para realizar el mapeado/compresión, sobre la base del valor de exponente de la función de potencia y/o los valores de los  
30 parámetros de las funciones de mapeado de correspondencia PWL, las palabras de código origen decodificadas a partir del contenido de vídeo de origen (104) a las palabras de código mapeadas y formato de palabras de código mapeadas. Algunas de las palabras de código objeto de mapeado están recortadas. Algunas o la totalidad de las palabras de código mapeadas contienen errores de cuantización causados por efectuar el mapeado de palabras de código VDR de profundidad de bits alta, con palabras de código mapeadas de profundidad de bits baja. El  
35 codificador multicapa (102) está configurado además para dar formato a los datos de imagen BL (106) (p.ej., 8 bits, etc.) en uno o más contenedores de imagen BL (p.ej., 8 bits, etc.) en un formato de muestreo (p.ej., 4:2:0, etc.).

En una forma de realización ejemplo, el codificador multicapa (102) proporciona, a la salida, los datos de imagen BL (106) en los contenedores de imágenes BL como una parte de una señal de vídeo multicapa a un dispositivo de flujo  
40 descendente (p.ej., el decodificador multicapa 152 de la Figura 1B, etc.).

En una forma de realización ejemplo, el codificador multicapa (102) está configurado para realizar una o más operaciones EL. Las una o más operaciones EL incluyen, pero no se limitan a, cualquiera de entre: operaciones de decodificación BL (120), operaciones de mapeado inverso (122), operaciones de resta (124), operadores de  
45 cuantización no lineal (126), operaciones de codificación EL (128), etc.

En las operaciones de decodificación BL (120), el codificador multicapa (102) está configurado para decodificar (p.ej., 8 bits, etc.) los datos de imagen BL (106) en los contenedores de imagen BL (p.ej., 8 bits, etc.) de nuevo en palabras de código mapeadas en una o más imágenes mapeadas que corresponden a una o más imágenes VDR de  
50 entrada.

En las operaciones de mapeado inverso (122), el codificador multicapa (102) está configurado para mapear, de forma inversa, basándose en una o más tablas de consulta (LUT), las palabras de código mapeadas en palabras de código VDR. Las tablas LUTs utilizadas para el mapeado inverso se pueden derivar, de forma analítica o numérica,  
55 a partir de la función de potencia con los valores exponenciales previamente determinados y/o las funciones de mapeado PWL con los valores de parámetro previamente determinados.

En algunas formas de realización, algunas o la totalidad de estas tablas LUTs comprenden claves de búsqueda (p.ej., palabras de código mapeadas, etc.) y valores de búsqueda (p.ej., palabras de código VDR, etc.) correspondientes a las claves de búsqueda. A modo de ejemplo, para fines de ilustración solamente, se puede  
60 construir una 1D LUT inversa, que se puede utilizar para realizar el mapeado, inversamente, de palabras de códigos mapeados a palabras de códigos VDR, mediante la sustitución de  $s_i = [0, 1, 2, \dots, 255]$  por la siguiente expresión:

$$LUT(s_i) = (v_H - v_L) \left( \frac{clip(s_i, c_L, c_H) - c_L}{c_H - c_L} \right)^{1/\alpha} + v_L \quad (1)$$

en donde  $c_L$  y  $c_H$  son valores mínimos y máximos, respectivamente, de las palabras de código mapeadas en el canal de luminancia, tal como se determina en las operaciones de mapeado y/o las operaciones de recorte;  $v_L$  y  $v_H$  son valores mínimos y máximos, respectivamente, de las palabras de código VDR en el canal de luminancia, según se determina en las operaciones de mapeado y/o las operaciones de recorte; y clip(...) es una función de recorte para asegurar que cualquier palabra de código fuera de margen (p.ej., fuera del margen [ $c_L$ ,  $c_H$ ], etc.) después de la compresión con pérdida (que puede ocurrir, a modo de ejemplo, en las operaciones de codificación BL, etc.) puedan todavía ser objeto de mapeado inverso con la palabra de código VDR válida más próxima.

Otros tipos de tablas LUTs (p.ej., basadas en PWL, etc.) también se pueden construir numéricamente de un modo similar. Cada una de las tablas LUTs, tal como se describe en la presente memoria, se puede utilizar para mapear inversamente palabras de código mapeadas decodificadas a partir de los datos de imagen BL (106) a palabras de código VDR correspondientes en uno o más canales específicos del espacio de color aplicable.

En algunas formas de realización, el codificador multicapa (102) está configurado para poner en práctica la predicción BL-a-EL (p.ej., predicción intra y/o inter, etc.) con el fin de reducir la cantidad de datos de imágenes EL que necesitan incluirse en las capas EL para reconstruir imágenes VDR mediante un dispositivo de flujo descendente (p.ej., un decodificador multicapa 152 de la Figura 1B, etc.). El codificador multicapa (102) puede generar, sobre la base, al menos en parte, en las palabras de código VDR obtenidas a través de las operaciones de mapeado inverso (122), datos de imagen de predicción.

En las operaciones de resta (124), el codificador multicapa (102) está configurado para generar valores residuales entre las palabras de código VDR decodificadas a partir del contenido de vídeo origen (104), y las palabras de código VDR representadas en los datos de imagen de predicción. Los valores residuales en un canal específico (p.ej., un canal de luminancia, un canal Y, etc.) pueden ser diferencias producidas por las operaciones de resta (124) en un dominio logarítmico o en un dominio lineal.

En las operaciones de cuantización no lineal (126), el codificador multicapa (102) está configurado para cuantizar los valores residuales en una primera representación digital (p.ej., 12+ bit, etc.) en una segunda representación digital (ej., 8 bits, etc.) en un espacio de color (p. ej., YCbCr, etc.) utilizando uno o más parámetros NLQ.

En las operaciones de codificación EL (128), el codificador multicapa (102) está configurado para codificar los valores residuales (p.ej., 8 bits, etc.) generados por las operaciones de cuantización no lineal, en los datos de imagen EL (108) en un contenedor de imagen EL (p.ej., 8 bits, etc.) en un formato de muestreo (p.ej., 4:2:0, etc.). El contenedor de imagen EL, en la capa mejorada, puede estar lógicamente separado del contenedor de imagen BL en la capa base, aunque ambos contenedores de imágenes pueden estar contenidos simultáneamente en una única señal de vídeo digital (p.ej., un único flujo de bits codificado, un fichero multimedia único, una única emisión, etc.).

En una forma de realización ejemplo, el codificador multicapa (102) proporciona, a la salida, los datos de imagen EL (108) en los contenedores de imagen EL como una parte de una señal de vídeo multicapa para un dispositivo de flujo descendente (p.ej., el decodificador multicapa 152 de la Figura 1B, etc.).

En una forma de realización ejemplo, el codificador multicapa (102) emite metadatos (130) que comprenden algunos o la totalidad de los parámetros operativos utilizados en las operaciones del codificador multicapa (102) como una parte de una señal de vídeo multicapa para un dispositivo de flujo descendente (p.ej., el decodificador multicapa 152 de la Figura 1B, etc.). Los parámetros operativos en los metadatos (130), transmitidos a dispositivos de flujo descendente incluyen, pero no están limitados a, cualquiera de entre: uno o más de los parámetros de mapeado, parámetros de recorte, valores exponenciales utilizados en funciones de potencia para compresión gamma, parámetros de mapeado inverso, tablas LUTs, valores pivote en funciones PWL, parámetros de cuantización no lineal, etc., mediante el mapeado de parámetros (134) y los parámetros NLQ (132), etc. Los metadatos (130) pueden ser una parte de los datos transmitidos en las capas EL y/o la capa BL, o en un sub-flujo de bits separado de un flujo de bits de vídeo global, a modo de ejemplo, como una parte de la información mejorada suplementaria (SEI) u otras transferencias de metadatos similares disponibles en el flujo de bits de vídeo. Un ejemplo de sub-flujo de bits puede ser un flujo de unidad de procesamiento de referencia (RPU) que se desarrolla por Dolby Laboratories, Inc.

A modo de ejemplo, las tablas LUTs, utilizadas en las operaciones de mapeado inverso (122), se pueden transmitir a dispositivos de flujo descendente como una parte de los metadatos (130). En algunas formas de realización, las claves de búsqueda y los valores correspondientes a las claves de búsqueda, en una LUT, se transmiten a dispositivos en flujo descendente como una parte de los metadatos (130). En algunas formas de realización, al menos una LUT puede ser representable mediante una función analítica o una función analítica de múltiples segmentos. En lugar de transmitir las claves de búsqueda y los valores correspondientes a las claves de búsqueda en la LUT a los dispositivos de flujo descendente, los parámetros que definen la función analítica se envían a los dispositivos de flujo descendente como parte de los metadatos (130), con el fin de reducir la cantidad de metadatos (130) en la transmisión. En algunas formas de realización, los parámetros en una función de mapeado, utilizados para derivar numéricamente una LUT, se transmiten a dispositivos de flujo descendente como una parte de los metadatos (130), en lugar de transmitir claves de búsqueda y valores correspondientes a las claves de búsqueda en

la LUT. Un dispositivo de flujo descendente puede utilizar los parámetros para derivar la tabla LUT, según se ilustra con la expresión (1). Una especificación de códec de vídeo que administra las operaciones de codificación y decodificación, puede incluir elementos sintácticos para transmitir uno o más parámetros, tal como se describe aquí, desde un dispositivo de flujo ascendente (p.ej., el codificador multicapa 102, etc.) a dispositivos de flujo descendente (p.ej., el decodificador multicapa 152, etc.).

Los datos de imagen EL (108), los datos de imagen BL (106) y los metadatos se pueden utilizar por el dispositivo de flujo descendente, con el fin de generar imágenes de una versión decodificada de margen dinámico relativamente amplio (p.ej., VDR, HDR, etc.) que representan las imágenes de entrada de margen dinámico relativamente amplio en el contenido de vídeo origen (104).

Una o más de las operaciones tales como las operaciones de codificación BL (118), las operaciones de decodificación BL (120), las operaciones de codificación EL (128), etc., se pueden poner en práctica utilizando uno o más de una pluralidad de códecs, incluyendo, sin limitación, cualquier combinación de: H.264/AVC/HEVC, MPEG-2, VP8, VC-1 y/o otros.

## 5. DECODIFICACIÓN DE VÍDEO MULTICAPA

Una señal de vídeo multicapa (p.ej., flujo de bits codificado, etc.) que comprende contenido de vídeo, se puede recibir mediante un decodificador multicapa (p.ej., 152 de la Figura 1B, etc.). En algunas formas de realización, el contenido de vídeo que se recibe por el decodificador multicapa (152) comprende datos de imagen BL (p.ej., 106 de la Figura 1A, Figura 1B, etc.) de una profundidad de bit relativamente baja y datos de imagen EL (p.ej., 108 de la Figura 1A y Figura 1B, etc.). En algunas formas de realización, tanto los datos de imagen BL (106) como los datos de imagen EL (108) se derivan/cuantizan a partir de la profundidad de bits relativamente alta (p.ej., VDR de 12+ bits, etc.) del contenido de vídeo origen (p.ej., 104 de la Figura 1A, etc.). En algunas formas de realización, el decodificador multicapa (152) está configurado para recibir metadatos (130) que comprenden algunos o la totalidad de los parámetros operativos utilizados en operaciones que generan los datos de imagen BL (106) y los datos de imagen EL (108) como una parte de la señal de vídeo multicapa.

En algunas formas de realización, los datos de imagen BL (106) se colocan en un contenedor de capa base (p.ej., un contenedor YCbCr 4:2:0 de 8 bits, etc.). En algunas formas de realización, los datos de imagen EL (108) comprenden datos de imagen residual (p.ej., VDR, etc.) del contenido de vídeo origen (104) con relación a datos de imágenes previstos, generados a partir de los datos de imágenes BL (106). En algunas formas de realización, los datos de imagen EL (108) se colocan en uno o más contenedores de capa mejorada (p.ej., uno o más contenedores de datos residuales de 8 bits, etc.).

En una forma de realización ejemplo, el decodificador multicapa (152) está configurado para realizar operaciones de decodificación basadas en PQ, en los datos de imagen BL (106) y los datos de imagen EL (108), con el fin de generar una o más imágenes de margen dinámico amplio (p.ej., VDR, etc.) que representan una versión reconstruida (p.ej., contenido de vídeo BL+EL reconstruido 166, etc.) de imágenes origen en contenido de vídeo origen que se utilizan para generar la señal de vídeo de capas múltiples. Las operaciones de decodificación basadas en PQ incluyen, pero no están limitadas a, una o más de entre: operación de decodificación BL (160), operaciones de mapeado inverso (162), operaciones de decodificación EL (154), operaciones de de-cuantización no lineal (156), operaciones de adición (158), etc.

En las operaciones de decodificación BL (160), el decodificador multicapa (152) está configurado para decodificar (p.ej., 8 bit, etc.) los datos de imagen BL (106) (p.ej., 8 bits, etc.) en los contenedores de imagen BL en palabras de código mapeadas en una o más imágenes mapeadas (p.ej., imágenes mapeadas por tonos).

En las operaciones de mapeado inverso (162), el decodificador multicapa (152) está configurado para mapear inversamente, a modo de ejemplo, sobre la base de una o más tablas de consulta (LUTs) decodificadas a partir de los metadatos (130), las palabras de código mapeadas en palabras de código VDR. En algunas formas de realización, el decodificador multicapa (152) está configurado para recibir, directamente, las tablas LUTs en los metadatos (130) en términos de claves de búsqueda y valores correspondientes a las claves de búsqueda. En algunas formas de realización, el decodificador multicapa (152) está configurado para recibir valores de parámetros tales como uno o más valores exponenciales, valores relacionados con pivote, etc., relacionados con una o más funciones, y para utilizar los valores de parámetro y las una o más funciones para generar (p.ej., de forma numérica, etc.) las una o más tablas LUTs (p.ej., claves de búsqueda y valores correspondientes a las claves de búsqueda, etc.) para aplicar en las operaciones de mapeado inverso (162).

En las operaciones de decodificación EL (154), el codificador multicapa (152) está configurado para generar (p.ej., 8 bits, etc.) valores residuales mediante la decodificación de los datos de imagen EL (108), que pueden estar en un contenedor de imagen EL (p.ej., 8 bits, etc.) en un formato de muestreo (p.ej., 4:2:0, etc.).

En las operaciones de de-cuantización no lineal (156), el decodificador multicapa (152) está configurado para la de-cuantización de los valores residuales en una representación digital de profundidad de bits relativamente baja (p.ej.,

8 bits, etc.) como decodificada a una representación digital de profundidad de bit relativamente alta (p.ej., 12+ bit, etc.) en un espacio de color (p.ej., YCbCr, etc.) utilizando uno o más parámetros NLQ decodificados a partir de los metadatos (130).

5 En las operaciones de adición (158), el decodificador multicapa (152) está configurado para generar la versión reconstruida de una o más imágenes de margen dinámico amplio, sobre la base de los valores residuales generados en las operaciones de decodificación EL (154) y las palabras de código VDR, generadas en las operaciones de mapeado inverso (162). La versión reconstruida de las una o más imágenes de margen dinámico amplio puede enviarse y/o mostrarse en un panel de visualización, a modo de ejemplo, mediante un sistema de visualización (p.ej.,  
10 HDR, VDR, etc.) que opera con o incluye, el decodificador multicapa (152).

En algunas formas de realización, el decodificador multicapa (152) está configurado para poner en práctica la predicción BL-a-EL (p.ej., predicción intra y/o inter, etc.) con el fin de reducir la cantidad de datos de imágenes EL que necesitan incluirse en las capas EL para reconstruir imágenes VDR mediante el decodificador multicapa (152).  
15 El decodificador multicapa (152) puede generar, sobre la base, al menos en parte, de las palabras de código VDR obtenidas a través de las operaciones de mapeado inverso (122), datos de imagen de predicción que comprenden las palabras de código VDR que han de utilizarse en las operaciones de adición (158).

En algunas formas de realización, los componentes o módulos usados en el mapeado inverso (162) o las operaciones de decodificación BL (160), del decodificador multicapa (152), pueden ser los mismos o prácticamente los mismos que los que se utilizan en el mapeado inverso (120), o las operaciones de decodificación BL (118) del codificador multicapa (102).

Una o más de las operaciones realizadas por el decodificador multicapa (152) pueden ponerse en práctica utilizando uno o más de una pluralidad de códecs, que incluyen, pero no se limitan a, cualquier combinación de:  
25 H.264/AVC/HEVC, MPEG-2, VP8, VC-1 y/o otros.

En algunas formas de realización, el de-cuantizador no lineal (156), el mapeado inverso (162) y el dispositivo sumador (158) pueden ser parte de una unidad de componedor (163). Sin limitación, en una forma de realización, la  
30 unidad de componedor (163) se puede poner en práctica utilizando instrucciones legibles por procesador que se ejecutan en un procesador, o utilizando hardware dedicado, tal como un FPGA y similares, o utilizando una combinación procesadores dedicados y de finalidad general.

La Figura 1C ilustra el flujo de datos para un componedor de "Perfil Base" (163) puesto en práctica utilizando una aritmética de punto fijo de conformidad con una forma de realización de esta invención. Dado los coeficientes de predicción  $a_i$ ,  $b_i$  y  $c_i$ , los parámetros del cuantizador inverso  $s$ ,  $m$ ,  $r_{max}$  y  $t$ , otros parámetros de metadatos (tal como los valores de pivote  $x_i$ ) y datos de entrada  $x_{BL}$  y  $x_{EL}$ , las operaciones del componedor se pueden expresar como:

```

BL_prediction() {
    yBL = ai + bi * xBL + ci * xBL2;
    Clip3(0, 1.0, yBL);
}

EL_inverse_quant() {
    yEL = s * ((xEL - m) - 0.5 * sign(xEL - m))
        + t * sign(xEL - m);
    Clip3(-rmax, rmax, yEL);
}

EDR_composing () {
    EDR = Round(yBL + yEL);
    Clip3(0, 2vdr_bit_depth - 1, EDR);
}

```

La Figura 1D ilustra el flujo de datos para un componedor de "Perfil Principal" (163) que se pone en práctica utilizando una aritmética de punto fijo de conformidad con otra forma de realización de esta invención. En la Figura 1D, x<sub>BL[0]</sub> indica una muestra de canal de luminancia, mientras que x<sub>BL[1]</sub> y x<sub>BL[2]</sub> indican muestras de canal cromático. Para x<sub>BL[0]</sub>, la predicción puede realizarse utilizando polinomios de segundo orden con coeficientes a<sub>i</sub>, b<sub>i</sub> y c<sub>i</sub>, tal como se representa en la Figura 1C. Para muestras cromáticas, la predicción puede basarse en polinomios de segundo orden o puede basarse en un predictor de regresión múltiple (MMR) de canales de múltiples colores, tal como se describe en la Solicitud PCT con nº de serie PCT/US2012/033605, presentada el 13 de abril de 2012.

La etapa de cuantización inversa EL, representada en la Figura 1D, puede ser la misma que la representada con más detalle en la Figura 1C, usando parámetros (s, m, t, r<sub>max</sub>). Se utiliza un muestreo ascendente espacial de coeficiente fijo si se establece un indicador correspondiente (p.ej., el\_spatial\_resampling\_filter\_flag) a 1. En una forma de realización, se utiliza un filtro de muestreo ascendente de 2x2 con un filtro horizontal de 8 tomas y un filtro vertical de 6 tomas. Conviene señalar que el muestreo ascendente espacial EL y el muestreo ascendente espacial BL son mutuamente exclusivos.

## 6. CONSERVACIÓN DE DETALLES VISUALES EN DATOS DE IMAGEN BL

Para fines de ilustración, en una o más de las operaciones de mapeado (114), las operaciones de recorte (116), o la operación de codificación BL (118), etc., el codificador multicapa (102) está configurado utilizar una función de potencia con un valor de exponente (α) como una función de mapeado para mapear palabras de código de luminancia VDR decodificadas a partir del contenido de vídeo origen (104) a palabras de código de luminancia mapeados, que pueden cuantizarse y codificarse en los datos de imagen BL (106). La función de potencia se puede representar en la siguiente expresión:

$$s_i = clip3(round((c_H - c_L) \left( \frac{v_i - v_L}{v_H - v_L} \right)^\alpha + c_L), c_L, c_H) \quad (2)$$

Cuando el valor exponente (α, que puede ser el mismo que en la expresión (1)) es uno (1), la función de potencia anterior se reduce a un mapeado de cuantización lineal. Cuando una tasa de bits disponible es alta, el valor de exponente se puede establecer, posiblemente, dentro de un margen relativamente grande ya que los artefactos en bloque son menos probables que sucedan con altas tasas de bits. Cuando una tasa de bit disponible es baja, el valor de exponente se puede establecer, posiblemente, dentro de un margen relativamente pequeño ya que los artefactos en bloque son más probables que sucedan con tasas de bits bajas. Un alto valor de exponente hace que la función de mapeado asigne, relativamente, más palabras de código a niveles de luminancia elevados que a niveles de luminancia bajos, y puede permitir que los datos de imagen BL (106) contengan más detalles en los elementos resaltados de las imágenes reconstruidas con menos artefactos de compresión. Sin embargo, un alto

valor de exponente también tiende a fusionar diferentes palabras de código origen (p.ej., VDR, etc.) en la misma o casi las mismas palabras de código comprimidas, en los elementos oscuros de las imágenes reconstruidas. En el caso más desfavorable, una parte oscura no plana (p.ej., con variaciones de luminancia perceptibles, etc.) en el contenido de vídeo origen se vuelve completamente plana (p.ej., con variaciones de luminancia próximas a 0 o inferiores a una diferencia apenas perceptible para la visión humana, etc.) en las imágenes reconstruidas, lo que tiene como resultado artefactos en bloque muy prominentes.

Con el fin de ayudar a evitar artefactos en bloque, un codificador multicapa (p.ej., 102) tal como se describe en este documento, puede estar configurado con un límite superior,  $MAX_{\alpha}$ , que el valor de exponente no debe exceder. En algunas formas de realización, este límite superior puede configurarse/establecerse/determinarse dependiendo de qué tasas de bits están disponibles para recibir una señal de vídeo multicapa, tal como aquí se describe. El codificador multicapa (p.ej., 102) está configurado para determinar un valor de exponente óptimo, limitado por el límite superior, en función del contenido de vídeo origen.

En algunas formas de realización, se puede seleccionar un valor de exponente óptimo global para la escena completa, para una pluralidad de imágenes VDR de entrada, etc. Para cada imagen VDR de entrada, se puede determinar primero un valor de exponente óptimo específico de imagen en función de esa imagen VDR de entrada. El valor de exponente óptimo global para la escena puede seleccionarse entonces (p.ej., el valor de exponente mínimo, etc.) a partir de uno o más valores de exponente óptimo específicos de imagen para una o más imágenes VDR de entrada en la escena.

En algunas formas de realización, una imagen VDR de entrada se puede dividir en una pluralidad de bloques de imagen VDR. Las desviaciones estándar entre palabras de código de bloques de imagen son indicaciones de cantidades respectivas de textura en los bloques de imagen. Una desviación estándar distinta de cero, según se describe aquí, puede representarse o determinarse, de forma alternativa, con una de las diferencias máximas-mínimas distintas de cero (p.ej., una diferencia entre los valores máximo y mínimo en un bloque de imagen, etc.), varianzas distintas de cero, valores de medición de suavidad, cada uno de los cuales corresponde a una desviación estándar distinta de cero o una varianza diferente de cero, etc. Por el contrario, una desviación estándar cero, según aquí se describe, puede representarse o determinarse, de forma alternativa, con una de las diferencias de cero máximas y mínima, varianzas cero, valores de medición de suavidad, cada una de los cuales corresponde a una desviación estándar cero o una varianza cero, etc. Un artefacto de bloques inducido por mapeado se produce cuando un bloque de imagen VDR, con desviación estándar distinta de cero, es objeto de mapeado para un bloque de imagen mapeado correspondiente que comprende palabras codificadas mapeados de una desviación estándar cero o muy pequeña (p.ej., una pequeña fracción de JND, etc.). En algunas formas de realización, una determinación de una desviación estándar distinta de cero en un bloque de imagen (p.ej., un bloque de imagen VDR, un bloque de imagen mapeado, etc.), es equivalente a, o se reduce para, una determinación de una diferencia distinta de cero entre los valores máximos y mínimos en ese bloque de imagen. De modo similar, una determinación de una desviación estándar cero en un bloque de imagen (p.ej., un bloque de imagen VDR, un bloque de imagen mapeado, etc.) es equivalente a, o se reduce para, una determinación de una diferencia cero entre los valores máximo y mínimo. En este caso, un bloque de imagen mapeado captura datos de imagen comprimida que pueden no estar optimizados para su visualización en una pantalla SDR.

## 7. DETERMINACIÓN EJEMPLO DEL VALOR DEL EXPONENTE EN LA FUNCIÓN DE POTENCIA

En algunas formas de realización, en una o más de las operaciones de mapeado (114), las operaciones de recorte (116), o la operación de decodificación BL (118), etc., el codificador multicapa (102) está configurado para utilizar una función de potencia para mapear palabras de código VDR (p.ej., palabras de código de luminancia VDR, etc.) decodificadas desde el contenido de vídeo de origen (104) a palabras de código mapeadas (p.ej., palabras de código de luminancia mapeados, etc.) que pueden cuantizarse y codificarse como una parte de los datos de imagen BL (106). En algunas formas de realización, un codificador multicapa (p.ej., 102) pone en práctica un algoritmo de búsqueda rápida o un flujo de proceso para localizar un valor de exponente óptimo  $\alpha$  de una función de mapeado de potencia, tal como se ilustra en la Figura 2A.

En el bloque 202, el codificador multicapa (102) realiza las operaciones de inicialización como sigue. Para cada imagen VDR de entrada  $j$  en un conjunto de  $F$  imágenes VDR de entrada en una escena, el codificador multicapa (102) inicializa un valor de exponente óptimo,  $\alpha_{opt}[j]$ , a 1.0. El codificador multicapa (102) divide, además, cada imagen VDR de entrada en el conjunto de  $F$  imágenes VDR de entrada en la escena en  $N$  bloques de imágenes. El codificador multicapa (102) ajusta, además,  $j$  a un valor inicial, a modo de ejemplo, 0.

En el bloque 204, el codificador multicapa (102) determina si  $j$  es menor que  $F$ . Si así, el flujo del proceso va al bloque 206. Si es falso, el flujo del proceso pasa al bloque 220.

En el bloque 206, para cada bloque de imagen  $n$  en la imagen VDR de entrada  $j$ , el codificador multicapa (102) calcula los valores de luminancia máxima y mínima  $B(j, n)$  y  $A(j, n)$  en ese bloque de imagen. Un conjunto  $\Phi_j$  de bloques de imagen con desviaciones estándar distintas de cero, en la imagen VDR de entrada  $j$ , se construye en base a la siguiente expresión:

$$\Phi_j = \{n \mid B(j, n) - A(j, n) > 0\} \quad (3)$$

Además, el codificador multicapa (102) inicializa una variable local  $\alpha_j$  a un valor inicial, a modo de ejemplo, 1.0.

En el bloque 208, el codificador multicapa (102) determina si  $\alpha_j$  es menor que  $MAX\_a$ . Si es así, el flujo del proceso pasa al bloque 210. Si es falso, el flujo del proceso pasa al bloque 218.

En el bloque 210, para cada bloque de imagen  $n$  en el conjunto  $\Phi_j$ , el codificador multicapa (102) efectúa el mapeado, sobre la base de una función de potencia con un valor de exponente como  $\alpha_j$ , palabras de código VDR (p.ej., palabras de código de luminancia VDR, etc.) en ese bloque a palabras de código mapeadas (p.ej., palabras de código de luminancia mapeados, etc.) en un bloque de imagen mapeado correspondiente, y calcula los valores máximo y mínimo  $b(j, n)$  y  $a(j, n)$  en el bloque de imagen mapeado, en un conjunto de bloques de imagen mapeado, que se deriva del conjunto  $\Phi_j$ , según se ilustra en las siguientes expresiones:

$$b_{jn}(\alpha_j) = \text{redondeo}((c_H - c_L) \left( \frac{B(j, n) - v_L}{v_H - v_L} \right)^{\alpha_j} + c_L) \quad (4)$$

$$a_{jn}(\alpha_j) = \text{redondeo}((c_H - c_L) \left( \frac{A(j, n) - v_L}{v_H - v_L} \right)^{\alpha_j} + c_L) \quad (5)$$

Posteriormente, el codificador multicapa (102) determina un número total de bloques de imagen mapeado con desviaciones estándar mayores que un valor umbral  $T_\sigma$ , tal como se ilustra en la siguiente expresión:

$$\sum_{n \in \Phi_j} (b_{jn}(\alpha_j) - a_{jn}(\alpha_j)) > T_\sigma \quad (6)$$

en donde  $T_\sigma$  se puede establecer en uno de: cero (0), un valor próximo a cero, una fracción de un JND, un límite superior para errores de cuantización en las operaciones de mapeado, etc.

En el bloque 212, el codificador multicapa (102) determina si el número total de bloques de imagen mapeado, con desviaciones estándar distintas de cero, es igual al número total de elementos en el conjunto  $\Phi_j$ . Si es así, el flujo del proceso pasa al bloque 214. Si es falso, el proceso pasa al bloque 218.

En el bloque 214, el codificador multicapa (102) establece  $\alpha_{opt[j]}$  para  $\alpha_j$ .

En el bloque 216, el codificador multicapa (102) aumenta  $\alpha_j$  mediante un incremento positivo, a modo de ejemplo, 0.1. El flujo del proceso pasa al bloque 208.

En el bloque 218, el codificador multicapa (102) incrementa  $j$  en 1. El flujo del proceso pasa al bloque 204.

En el bloque 220, el codificador multicapa (102) selecciona uno de  $\alpha_{opt[j]}$ , en donde  $j$  es desde 0 a  $(F-1)$ , como el valor de exponente óptimo global para el conjunto de imágenes VDR de entrada  $F$  en la escena.

En algunas formas de realización, el valor de exponente óptimo global se selecciona como el valor mínimo entre  $\alpha_{opt[j]}$ , en donde  $j$  es desde 0 a  $(F-1)$ , tal como se ilustra en la siguiente expresión:

$$\alpha = \min \{ \alpha^{opt}[j] \} \quad (7)$$

La utilización de una función de mapeado de potencia con un valor de exponente óptimo global, según se ilustra en la expresión (7) evita, o reduce la probabilidad de que aparezcan bloques de imagen completamente planos inducidos por mapeado en todas las imágenes mapeadas del conjunto de imágenes VDR de entrada  $F$ .

En algunas otras formas de realización, el valor del exponente óptimo global puede ser un valor cerrado, pero no necesariamente, el valor mínimo entre  $\alpha_{opt[j]}$ , en donde  $j$  es desde 0 a  $(F-1)$ . Mediante la utilización de una función de mapeado de potencia con un valor de exponente óptimo global que no sea el mínimo, se puede causar todavía uno o más bloques de imagen completamente planos inducidos por mapeado en una o más de las imágenes mapeadas a partir del conjunto de imágenes VDR de entrada  $F$ .

Un listado de algoritmos ejemplo, para operaciones ilustradas en la Figura 2A, se muestra en la siguiente tabla:

TABLA 1

```

 $opt[j] = 1.0$  para todos  $j$ .

// para cada trama  $j$  dentro de una escena
para ( $j = 0; j < F; j++$ ) {
    // PASO 1.a: encontrar el valor máximo para cada bloque  $n$  en el dominio VDR
    Calcular  $B(j, n)$  para cada  $n$ 
    // PASO 1.b: encontrar el valor mínimo para cada bloque  $n$  en el dominio VDR
    Calcular  $A(j, n)$  para cada  $n$ 
    // PASO 1.c: encontrar bloques con  $(B(j, n) - A(j, n)) > 0$ ; es decir, índice de bloque estándar distinto de cero
    Construir  $\Phi_j = \{n \mid B(j, n) - A(j, n) > 0\}$ 
    // PASO 2: calcular la desviación estándar para cada bloque en el dominio BL con diferente
    para ( $j = 1.0; j < MAX\_j; j++ = 0.1$ ) {
        // PASO 2a: cuantizar  $B(j, n)$  en  $\Phi_j$  a BL utilizando la función de potencia con  $j$ 

$$b_{jn}(\alpha_j) = \text{redondear} \left( ((c_H - c_L) \left( \frac{B(j, n) - v_L}{v_H - v_L} \right)^{\alpha_j} + c_L) \right)$$

        // PASO 2b: cuantizar  $A(j, n)$  en  $\Phi_j$  a BL utilizando la función de potencia con  $j$ 

$$a_{jn}(\alpha_j) = \text{redondear} \left( ((c_H - c_L) \left( \frac{A(j, n) - v_L}{v_H - v_L} \right)^{\alpha_j} + c_L) \right)$$

        // PASO 2c: calcular si da  $j$  causa cero desviación
        Si

$$\left( \left| \Phi_j \right| - \sum_{n \in \Phi_j} (b_{jn}(\alpha_j) - a_{jn}(\alpha_j)) > T_\sigma \right) == 0 )$$

 $opt[j] = j$ 
        Else (si no)
            break (ruptura);
    }
}
// PASO 3: encontrar la mínima  $opt[j]$  como la final
 $= \min \{opt[j]\}$ 

```

## 8. DETERMINACIÓN EJEMPLO DE VALORES DE PARÁMETROS EN CUANTIZACIÓN LINEAL, INCLUYENDO LA CUANTIZACIÓN LINEAL POR ELEMENTOS

En algunas formas de realización, en una o más de las operaciones de mapeado (114), las operaciones de recorte (116), o la operación de decodificación BL (118), etc., el codificador multicapa (102) está configurado para utilizar una función de mapeado PWL (o un cuantizador PWL) para mapear palabras de código VDR (p.ej., palabras de código de croma VDR, etc.) decodificadas a partir del contenido de vídeo de origen (104) para palabras de código mapeadas (p.ej., palabras de código de croma mapeadas, etc.), que puede cuantizarse y codificarse como una parte de los datos de imagen BL (106). La función de mapeado PWL puede incluir uno o más segmentos, cada uno de los cuales puede representarse mediante la siguiente expresión (ignorar el redondeo aquí dado) con diferentes valores para los parámetros  $m$  y  $b$ :

$$s_i = m(v_i - v_L) + b \quad (8)$$

en donde  $m$  y  $b$  son parámetros de gradiente e intersección, e  $i$  es un índice asociado con una variable de entrada  $v_i$  (p.ej., una  $i$ -ésima palabra de código de croma VDR, etc.) y una palabra de código mapeado  $s_i$  (p.ej., una correspondiente  $i$ -ésima palabra de código de croma mapeado, etc.).

La función de mapeado PWL se puede reducir a una función de mapeado lineal (p.ej., cuantización lineal, un cuantizador lineal, etc.) cuando el número total de segmentos, en la función de mapeado PWL, es uno (1).

Un parámetro  $m$  más alto en la expresión (8) produce más compresión en las palabras de código mapeadas en el lado izquierdo de la expresión (8), haciendo que los artefactos en bloque sean más propensos a ocurrir. Con el fin de mejorar la eficiencia de codificación, los parámetros  $m$  y  $b$  finalmente seleccionados pueden ser los mismos dentro de toda la escena. En algunas formas de realización, se puede utilizar un segmento único o una función de mapeado PWL de tres segmentos para mapear palabras de código VDR (p.ej., palabras de código de croma VDR, etc.) para palabras codificadas mapeados (p.ej., palabras de código de croma mapeados, etc.) en una escena.

El uso de una función de mapeado PWL de múltiples segmentos (p.ej., una función de mapeado PWL de tres segmentos, etc.) permite que un cuantizador lineal diferente (p.ej., diferentes valores para  $m$ , etc.) se aplique a un margen diferente de datos VDR de entrada (p. ej., palabras de código de croma VDR de entrada, etc.). Se puede

mapear un primer margen cromático de datos VDR de entrada con un primer segmento de una función de mapeado PWL. Se puede mapear un segundo margen cromático de los datos VDR de entrada con un segundo segmento de la función de mapeado PWL. Se puede mapear un tercer margen cromático de los datos VDR de entrada con un tercer segmento de la función de mapeado PWL. El mejor cuantizador lineal puede buscarse y determinarse para cada uno de los múltiples segmentos. Se puede poner en práctica un algoritmo para buscar valores de parámetros óptimos específicos de imagen para cada imagen VDR de entrada en un conjunto de imágenes VDR de entrada en una escena, y a continuación, se puede encontrar una conclusión o valores de parámetros óptimos generales entre todos los valores de parámetros óptimos específicos de la imagen de las imágenes VDR de entrada dentro de la misma escena.

Una función de mapeado PWL, tal como se describe en el presente documento, puede especificarse con puntos de pivote. En algunas formas de realización, un codificador multicapa (p.ej., 102) pone en práctica un algoritmo de búsqueda rápida o un flujo de proceso para localizar un conjunto óptimo de puntos de pivote, tal como se ilustra en la Figura 2B.

En el bloque 252, el codificador multicapa (102) realiza las operaciones de inicialización del modo siguiente. Para cada imagen VDR de entrada  $j$  en un conjunto de  $F$  imágenes VDR de entrada en una escena, el codificador multicapa (102) inicializa un factor de escala de inclinación,  $w^{opt}[j]$ , a 1.0. El codificador multicapa (102) puede estar configurado para dividir cada imagen VDR de entrada en el conjunto de  $F$  imágenes VDR de entrada en  $N$  bloques de imagen, o reutilizar los  $N$  bloques de imagen como divididos en cada imagen VDR de entrada en el conjunto de  $F$  imágenes VDR de entrada en la escena, en el flujo del proceso que se ha utilizado para buscar una  $\alpha$  óptima. El codificador multicapa (102) establece, además,  $j$  para un valor inicial, a modo de ejemplo, 0.

En el bloque 254, el codificador multicapa (102) determina si  $j$  es menor que  $F$ . Si es así, el flujo del proceso pasa al bloque 256. Si es falso, el flujo del proceso pasa al bloque 270.

En el bloque 256, para cada bloque de imagen  $n$  en la imagen VDR de entrada  $j$ , el codificador multicapa (102) calcula los valores máximo, mínimo y medio  $D(j, n)$ ,  $E(j, n)$ ,  $\mu(j, n)$  (p.ej., en palabras de código de croma VDR, etc.) en ese bloque de imagen. El codificador multicapa (102) calcula, además, los valores mapeados  $d_{jn}(w_j)$ ,  $e_{jn}(w_j)$  (p.ej., palabras de código de croma mapeados, etc.) correspondientes a los valores máximo y mínimo  $D(j, n)$  y  $E(j, n)$  en ese bloque de imágenes basado en las siguientes expresiones (ilustradas aquí para una capa base de 8 bits. Si es una capa base de 10 bits, un ejemplo de función de recorte puede ser  $\text{clip}(x, 0, 1023)$ ):

$$d_{jn}(w_j) = \text{clip3}\left[w_j(C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (D(j, n) - v_L^C) + C_r^C, 0, 255\right] \quad (10)$$

$$e_{jn}(w_j) = \text{clip3}\left[w_j(C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (E(j, n) - v_L^C) + C_r^C, 0, 255\right] \quad (11)$$

En donde  $C_H^Y$ ,  $C_L^Y$ ,  $v_H^Y$ ,  $v_L^Y$ ,  $v_H^C$ ,  $v_L^C$ , y  $C_r^C$  son el valor de palabra de código de luminancia mapeado máxima, el valor de palabra de código de luminancia mapeado mínima, el valor de palabra de código de luminancia VDR de entrada máxima, el valor de palabra de código de luminancia VDR de entrada mínima, el valor de palabra de código de croma VDR de entrada máxima, el valor de palabra de código de croma VDR de entrada mínima, y un parámetro en las relaciones lineales representadas por las expresiones anteriores (10) y (11). El codificador multicapa (102) establece un conjunto  $K_j$  de bloques de imagen con desviaciones estándar distintas de cero en la imagen VDR de entrada  $j$ , pero con desviaciones estándar cero en los correspondientes valores de palabra de código de croma mapeados, basados en la siguiente expresión:

$$K_j = \{i | (D(j, n) - E(j, n) > 0) \& (d_{jn}(w_j) - e_{jn}(w_j) = 0)\} \quad (12)$$

En el bloque 258, el codificador multicapa (102) determina si el conjunto  $K_j$  está vacío. Si es así, el flujo del proceso pasa al bloque 276. Si es falso, el flujo del proceso pasa al bloque 260.

En el bloque 276, el codificador multicapa (102) está configurado para utilizar un cuantizador lineal de un solo segmento para la imagen VDR de entrada  $j$ , y establece parámetros con las siguientes expresiones:

$$\mu_{\min}(j) = v_L^C \quad (13)$$

$$\mu_{\max}(j) = v_H^C \quad (14)$$

$$w_j^I = 1 \quad (15)$$

$$w_j^M = 1 \quad (16)$$

$$w_j^H = 1 \quad (17)$$

En el bloque 260, el codificador multicapa (102) encuentra valores mínimos y máximos de los valores promediados a partir del conjunto  $K_j$  en las siguientes expresiones:

$$\mu_{\min}(j) = \min_{n \in K_j} \{\mu(j, n)\} \quad (18)$$

$$\mu_{\max}(j) = \max_{n \in K_j} \{\mu(j, n)\} \quad (19)$$

Los valores mínimo y máximo ( $\mu_{\min}(j)$  y  $\mu_{\max}(j)$ ) de los valores promediados se pueden utilizar para dividir el margen completo de valores de palabra de código de croma VDR de entrada entre  $v_L^C$  y  $v_H^C$  en tres segmentos:  $[v_L^C, \mu_{\min}(j)]$ ,  $[\mu_{\min}(j), \mu_{\max}(j)]$ , and  $[\mu_{\max}(j), v_H^C]$ .

Debe observarse que el uso de las expresiones (18) y (19) para generar particiones para valores de palabra de código de croma de VDR de entrada es para fines de ilustración solamente. En otras formas de realización, se pueden utilizar más o menos valores (p.ej., puntos medios y/o puntos fraccionarios entre cualesquiera dos de estos valores, etc.) o segmentos, a modo de ejemplo, mediante la generación de más o menos particiones a partir de valores de palabra de código de croma de entrada de media, máximo o mínimo del conjunto de imágenes de entrada en la escena.

Conviene señalar que, para los fines de la invención, un conjunto de imágenes de entrada puede comprender un número variable de imágenes de entrada, y/o puede estar relacionado, o no, con una escena. A modo de ejemplo, los flujos del proceso de la Figura 2A y Figura 2B se puede realizar con un conjunto de imágenes de entrada que pertenecen a un subconjunto de una escena. Además, los flujos del proceso de la Figura 2A y Figura 2B se puede realizar con un conjunto de imágenes de entrada que no pertenecen a una sola escena en otras formas de realización.

En el bloque 260, el codificador multicapa (102) puede estar configurado, además, para construir tres conjuntos

$K_j^L$ ,  $K_j^M$ , y  $K_j^H$  de bloques de imágenes sobre la base de las siguientes expresiones:

$$K_j^L = \{n \mid v_L^C \leq \mu(j, n) < \mu_{\min}(j), \forall n\} \quad (20)$$

$$K_j^M = \{n \mid \mu_{\min}(j) \leq \mu(j, n) < \mu_{\max}(j), \forall n \in K_j\} \quad (21)$$

$$K_j^H = \{n \mid \mu_{\max}(j) \leq \mu(j, n) \leq v_H^C, \forall n\} \quad (22)$$

Tal como se ilustra,  $K_j^M$  representa bloques de imagen VDR, en el segmento medio de la función de mapeado PWL, que tienen una desviación estándar distinta de cero en los valores cromáticos VDR, pero tienen desviaciones

estándar cero en los valores cromáticos mapeados con el factor de escala de gradiente,  $w_j$ , establecido en 1.0.  $K_j^L$  representa bloques de imagen VDR, en el segmento bajo de la función de mapeado PWL, que tienen una desviación

estándar distinta de cero en los valores cromáticos VDR.  $K_j^H$  representa bloques de imagen VDR, en el segmento superior de la función de mapeado PWL, que tiene una desviación estándar distinta a cero en valores cromáticos VDR.

En el bloque 262, el codificador multicapa (102) está configurado para establecer una matriz  $\Omega^L$  de valores candidatos para el factor de escala de gradiente,  $w_j$ , que ha de utilizarse para el conjunto  $K_j^L$ , tal como se ilustra en la siguiente expresión:

$$\Omega^L = \{1, 1-\Delta w, 1-2\Delta w, \dots, 1-(S^L-1)\Delta w\} \quad (23)$$

en donde  $\Delta w$  es un valor de decremento configurable o fijo y  $S^L$  es un número configurable o fijo de elementos de matriz en  $\Omega^L$ .

Para una variable local  $s$  en cada valor entero entre 0 y  $S^L$ , el codificador multicapa (102) utiliza las expresiones (10) y (11) para calcular o volver a calcular los valores cromáticos mapeados  $d_{jn}(w_j)$  y  $e_{jn}(w_j)$  correspondiente a los valores

cromáticos máximo y mínimo  $D(j, n)$  y  $E(j, n)$  para cada bloque de imagen con  $n$  en  $K_j^L$ , en donde el factor de

- 5 escala de gradiente,  $w_j$ , que ha de utilizarse para el conjunto  $K_j^L$ , se establece en un valor candidato en  $\Omega^L$ , según se ilustra en la siguiente expresión:

$$w_j = \Omega^L[s] \quad (24)$$

- 10 Además, para  $s$  en cada valor entero entre 0 y  $S^L$ , el codificador multicapa (102) calcula una cantidad  $\pi^L(s)$  que compara el número (indicado como  $|K_j^L|$ ) de elementos en  $K_j^L$  con el número de bloques de imagen con desviaciones estándar en los valores cromáticos mapeados por encima de un valor umbral  $T$ , tal como se ilustra en la siguiente expresión:

$$\pi^L(s) = ((\sum_{n \in K_j^L} ((D_{jn}(w_j) - E_{jn}(w_j) > 0) \& (d_{jn}(w_j) - e_{jn}(w_j) == 0))) == 0)$$

- 15 (25)

Sobre la base de  $\pi^L(s)$ , un factor de escala optimizado  $w_j^L$  de gradiente para el segmento inferior de la función de mapeado PWL, se puede derivar con las siguientes expresiones:

$$20 \quad w_j^L = \Omega^L(s_{opt}^L) \quad (26)$$

en donde  $s_{opt}^L$  se proporciona como sigue:

$$s_{opt}^L = \arg \min_{\{s\}} s \quad \text{s.t.} \quad \sum_{u=s}^{S^L-1} \pi^L(u) = 0. \quad (27)$$

- 25 El procesamiento, según se examinó con anterioridad, se puede utilizar para derivar cero, uno, dos o más factores de escala optimizados para cero, uno, dos o más segmentos de la función de mapeado PWL. A modo de ejemplo, en el bloque 264, el codificador multicapa (102) está configurado para establecer una matriz  $\Omega^M$  de valores candidato

- 30 para el factor de escala de gradiente,  $w_j$ , que ha de utilizarse para el conjunto  $K_j^M$ , según se ilustra en la siguiente expresión:

$$\Omega^M = \{1, 1+\Delta w, 1+2\Delta w, \dots, 1+(S^M-1)\Delta w\} \quad (28)$$

- 35 en donde  $\Delta w$  es un valor de decremento configurable o fijo y  $S^M$  es un número configurable o fijo de elementos de matriz en  $\Omega^M$ .

Para una variable local  $s$  en cada valor entero entre 0 y  $S^M$ , el codificador multicapa (102) utiliza las expresiones (10) y (11) para calcular, o volver a calcular, los valores cromáticos mapeados  $d_{jn}(w_j)$  y  $e_{jn}(w_j)$  correspondientes a los

valores cromáticos máximo y mínimo  $D(j, n)$  y  $E(j, n)$  para cada bloque de imagen con  $n$  en  $K_j^M$ , en donde el factor

- 40 de escala de gradiente,  $w_j$ , que ha de utilizarse para el conjunto  $K_j^M$ , se establece en un valor candidato en  $\Omega^M$ , tal como se ilustra en la siguiente expresión:

$$w_j = \Omega^M[s] \quad (29)$$

- 45 Además, para  $s$  en cada valor entero entre 0 y  $S^M$ , el codificador multicapa (102) calcula una cantidad  $\pi^M(s)$  que compara el número (indicado como  $|K_j^M|$ ) de elementos en  $K_j^M$  con el número de bloques de imagen con desviaciones estándar en los valores cromáticos mapeados por encima de un valor umbral  $T$ , según se ilustra en la siguiente expresión:

$$\pi^M(s) = ((\sum_{n \in K_j^M} ((D_{jn}(w_j) - E_{jn}(w_j)) > 0) \& \& (d_{jn}(w_j) - e_{jn}(w_j) == 0))) == 0) \quad (30)$$

Sobre la base de  $\pi^M(s)$ , un factor de escala optimizado  $w_j^M$  de la gradiente para el segmento medio de la función de mapeado PWL se puede derivar con las siguientes expresiones:

$$w_j^M = \Omega^M(s_{opt}^M) \quad (31)$$

en donde  $s_{opt}^M$  se proporciona como sigue:

$$s_{opt}^M = \arg \min_{[s]} s \quad \text{s.t.} \quad \sum_{u=s}^{S^M-1} \pi^M(u) = 0. \quad (32)$$

En el bloque 266, el codificador multicapa (102) está configurado para establecer una matriz  $\Omega^H$  de valores candidato para el factor de escala de gradiente,  $w_j$ , que ha de utilizarse para el conjunto  $K_j^H$ , según se ilustra en la siguiente expresión:

$$\Omega^H = \{1, 1-\Delta w, 1-2\Delta w, \dots, 1-(S^H-1)\Delta w\} \quad (33)$$

en donde  $\Delta w$  es un valor de decremento configurable o fijo y  $S^H$  es un número configurable o fijo de elementos de matriz en  $\Omega^H$ .

Para una variable local  $s$  en cada valor entero entre 0 y  $S^H$ , el codificador multicapa (102) utiliza las expresiones (10) y (11) para calcular, o volver a calcular, los valores cromáticos mapeados  $d_{jn}(w_j)$  y  $e_{jn}(w_j)$  correspondiente a los valores cromáticos máximo y mínimo  $D(j, n)$  y  $E(j, n)$  para cada bloque de imagen con  $n$  en  $K_j^H$ , en donde el factor de escala de gradiente,  $w_j$ , que ha de utilizarse para el conjunto  $K_j^H$ , se establece en un valor candidato en  $\Omega^H$ , tal como se ilustra en la siguiente expresión:

$$w_j = \Omega^H[s] \quad (34)$$

Además, para  $s$  en cada valor entero entre 0 y  $S^H$ , el codificador multicapa (102) calcula una cantidad  $\pi^H(s)$  que compara el número (indicado como  $|K_j^H|$ ) de elementos en  $K_j^H$  con el número de bloques de imagen con desviaciones estándar en los valores cromáticos mapeados por encima de un valor de umbral  $T$ , según se ilustra en la siguiente expresión:

$$\pi^H(s) = ((\sum_{n \in K_j^H} ((D_{jn}(w_j) - E_{jn}(w_j)) > 0) \& \& (d_{jn}(w_j) - e_{jn}(w_j) == 0))) == 0) \quad (35)$$

Sobre la base de  $\pi^H(s)$ , un factor de escala optimizado  $w_j^H$  de la gradiente para el segmento alto de la función de mapeado PWL, se puede derivar con las siguientes expresiones:

$$w_j^H = \Omega^H(s_{opt}^H) \quad (36)$$

en donde  $s_{opt}^H$  se proporciona como sigue:

$$s_{opt}^H = \arg \min_{[s]} s \quad \text{s.t.} \quad \sum_{u=s}^{S^H-1} \pi^H(u) = 0. \quad (37)$$

En el bloque 268, el codificador multicapa (102) incrementa  $j$  en 1. El flujo del proceso pasa al bloque 254.

En el bloque 270, el codificador multicapa (102) genera un conjunto de parámetros optimizados que se utilizarán para definir las coordenadas de los pivotes de la función de mapeado PWL, tal como se ilustra en las siguientes expresiones:

$$w^L = \max_j \{ w_j^L \} \quad (38)$$

$$w^M = \max_j \{ w_j^M \} \quad (39)$$

$$w^H = \max_j \{ w_j^H \} \quad (40)$$

$$\mu_{\min} = \min_j \{ \mu_{\min}(j) \} \quad (41)$$

$$\mu_{\max} = \max_j \{ \mu_{\max}(j) \} \quad (42)$$

En el bloque 272, en base al conjunto de parámetros optimizados, el codificador multicapa (102) genera las coordenadas de todos los pivotes e (p.ej., cuatro pivotes e) para la función de mapeado PWL, según se ilustra en las siguientes expresiones:

$$\text{pivot\_x}[] = \{ v_L^C, \mu_{\min}, \mu_{\max}, v_H^C \} \quad (43)$$

$$\text{pivot\_y}[0] = 0 \quad (44)$$

$$\text{pivot\_y}[1] = \text{pivot\_y}[0] + w^L (C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (\mu_{\min} - v_L^C) \quad (45)$$

$$\text{pivot\_y}[2] = \text{pivot\_y}[1] + w^M (C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (\mu_{\max} - \mu_{\min}) \quad (46)$$

$$\text{pivot\_y}[3] = \text{pivot\_y}[2] + w^H (C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (v_H^C - \mu_{\max}) \quad (47)$$

en donde  $\text{pivot\_x}[]$  y  $\text{pivot\_y}[]$  son dos matrices que representan respectivamente las coordenadas  $x$  y las coordenadas  $y$  de los cuatro pivotes.

En el bloque 274, el codificador multicapa (102) está configurado para determinar si el número total de palabras de código necesarias para incluir los valores de coordenadas  $y$  de la función de mapeado PWL supera el número total (p.ej., 255, etc.) de palabras de código disponibles en un espacio de código utilizado para codificar los datos de imagen BL. A modo de ejemplo, el codificador multicapa (102) puede estar configurado para determinar si  $\text{pivot\_y}[3]$  es mayor que el número total (p.ej., 255, etc.) de palabras de código disponibles. Si ese es el caso, para una variable local  $n$  en cada valor entre 0 y 3, el codificador multicapa (102) se puede configurar para comprimir las coordenadas  $y$  de la función de mapeado PWL con la siguiente expresión:

$$\text{pivot\_y}[n] = \text{clip3}(\text{round}(\text{pivot\_y}[n] * 255 / (\text{pivot\_y}[3] - \text{pivot\_y}[0])), 0, 255)$$

(48)

Un listado de algoritmos ejemplo para las operaciones ilustradas en la Figura 2B se muestra en la siguiente tabla:

TABLA 2

```

wopt[j] = 1.0 para todos f. wopt[j] = 1.0 para todos j.

// para cada trama;
para (j = 0; j < F; j++) {
// PASO 1.a: encontrar el valor máximo para cada bloque n en el dominio VDR
Calcular D(j, n) para cada n

```

```

// PASO 1.b: encontrar el valor mínimo para cada bloque n en el dominio VDR
Calcular  $E(j, n)$  para cada n
// PASO 1.c: encontrar el valor medio para cada bloque n en el dominio VDR
Calcular  $(j, n)$  para cada n

// PASO 1.d. set  $w_j = 1$ . cuantizar  $D(j, n)$  y  $E(j, n)$  en  $K_j$  a BL


$$d_{jn}(w_j) = clip3\left(w_j(C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (D(j, n) - v_L^C) + C_r^C, 0, 255\right) \quad (\text{CNLSQ-a})$$


$$e_{jn}(w_j) = clip3\left(w_j(C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (E(j, n) - v_L^C) + C_r^C, 0, 255\right) \quad (\text{CNLSQ-b})$$


// PASO 1.e: encontrar el bloque con  $(D(j, n) - E(j, n)) > 0$  y  $(d_{jn}(w_j) - e_{jn}(w_j) == 0)$ ;
// es decir, índice de bloque estándar distinto de cero en VDR pero cero en el dominio BL.
Construir  $K_j = \{i \mid (D(j, n) - E(j, n)) > 0 \ \&\& (d_{jn}(w_j) - e_{jn}(w_j) = 0)\}$ 

Si  $(|K_j| == 0)$  {
    // usar el predictor lineal es suficiente;
     $\mu_{\min}(j) = v_L^C$ 
     $\mu_{\max}(j) = v_H^C$ 
     $w_j^L = 1$ 
     $w_j^M = 1$ 
     $w_j^H = 1$ 
} // fin de lineal
Else {
    // PASO 1.f: encontrar el valor mínimo y máximo del valor medio de  $K_j$ 
     $\mu_{\min}(j) = \min_{n \in K_j} \{\mu(j, n)\}$ 
     $\mu_{\max}(j) = \max_{n \in K_j} \{\mu(j, n)\}$ 

    // PASO 1.g.
    // encontrar tres conjuntos separados por  $[v_L^C, \mu_{\min}(j)]$ ,  $[\mu_{\min}(j), \mu_{\max}(j)]$  y  $[\mu_{\max}(j), v_H^C]$ 
     $K_j^L = \{n \mid v_L^C \leq \mu(j, n) < \mu_{\min}(j), \forall n\}$ 
     $K_j^M = \{n \mid \mu_{\min}(j) \leq \mu(j, n) < \mu_{\max}(j), \forall n \in K_j\}$ 
     $K_j^H = \{n \mid \mu_{\max}(j) \leq \mu(j, n) \leq v_H^C, \forall n\}$ 

    // PASO 2.a: calcular el estándar para cada bloque en  $K_j^L$  en el dominio BL con diferente  $w_j$ 
     $\Omega^L = \{1, 1-w, 1-2w, \dots, 1-(S^L-1)w\}$ 
    para  $(s = 0; s < S^L; s++)$  {
         $w_j = \Omega^L[s];$ 
        // PASO 2.a. 1: cuantizar  $D(j, n)$  en  $K_j^L$  a BL con gradiente  $w_j$  utilizando (CNLSQ-a)
        // PASO 2.a.2: cuantizar  $E(j, n)$  en  $K_j^L$  a BL con gradiente  $w_j$  utilizando (CNLSQ-b)
        // PASO 2.a.3: calcular si da  $w_j$  causa cero desviación
         $\pi^L(s) = ((\sum_{n \in K_j^L} ((D_{jn}(w_j) - E_{jn}(w_j)) > 0) \ \&\& (d_{jn}(w_j) - e_{jn}(w_j) == 0))) == 0)$ 
    }

     $s_{opt}^L = \arg \min_{\{s\}} s \quad \text{s.t.} \quad \sum_{u=s}^{S^L-1} \pi^L(u) = 0.$ 
     $w_j^L = \Omega^L(s_{opt}^L)$ 

    // PASO 2.b: calcular el estándar para cada bloque en  $K_j^M$  en el dominio BL con diferente  $w_j$ 
     $\Omega^M = \{1, 1+w, 1+2w, \dots, 1+(S^M-1)w\}$ 
    para  $(s = 0; s < S^M; s++)$  {
         $w_j = \Omega^M[s];$ 
        // PASO 2.b.1: cuantizar  $D(j, n)$  en  $K_j^M$  a BL con gradiente  $w_j$  utilizando (CNLSQ-a)
    }

```

```

// PASO 2.b.2: cuantizar  $E(j, n)$  en  $K_j^M$  a BL con gradiente  $w_j$  utilizando (CNLSQ-b)
// PASO 2.b.3: calcular si da  $w_j$  causa cero desviación
 $\pi^M(s) = ((\sum_{n \in K_j^M} ((D_{jn}(w_j) - E_{jn}(w_j)) > 0) \&\&(d_{jn}(w_j) - e_{jn}(w_j) == 0))) == 0)$ 
}
// encontrar el primer cero que lleva una secuencia de ceros hacia el final de la lista
 $s_{opt}^M = \arg \min_{\{s\}} s \quad \text{s.t.} \quad \sum_{u=s}^{S^M-1} \pi^M(u) = 0.$ 
 $w_j^M = \Omega^M(s_{opt}^M)$ 

// PASO 2.c: calcular el estándar para cada bloque en  $K_j^H$  en el dominio BL con diferente  $w_j$ 
 $\Omega^H = \{1, 1-w, 1-2w, \dots, 1-(S^H-1)w\}$ 
para ( $s = 0$ ;  $s < S^H$ ;  $s++$ ) {
 $w_j = \Omega^H[s]$ ;
// PASO 2.c.1: cuantizar  $D(j, n)$  en  $K_j^H$  a BL con gradiente  $w_j$  utilizando (CNLSQ-a)
// PASO 2.c.2: cuantizar  $E(j, n)$  en  $K_j^H$  a BL con gradiente  $w_j$  utilizando (CNLSQ-b)
// PASO 2.c.3: calcular si da  $w_j$  causa cero desviación
 $\pi^H(s) = ((\sum_{n \in K_j^H} ((D_{jn}(w_j) - E_{jn}(w_j)) > 0) \&\&(d_{jn}(w_j) - e_{jn}(w_j) == 0))) == 0)$ 
}

 $s_{opt}^H = \arg \min_{\{s\}} s \quad \text{s.t.} \quad \sum_{u=s}^{S^H-1} \pi^H(u) = 0.$ 
 $w_j^H = \Omega^H(s_{opt}^H)$ 
} // fin de no lineal
}

// PASO 3: construye PWL
 $w^L = \max_j \{w_j^L\}$ 
 $w^M = \max_j \{w_j^M\}$ 
 $w^H = \max_j \{w_j^H\}$ 
 $\mu_{\min} = \min_j \{\mu_{\min}(j)\}$ 
 $\mu_{\max} = \max_j \{\mu_{\max}(j)\}$ 

// entrada de puntos de partición VDR
pivot_x[] = {  $v_L^C$ ,  $\mu_{\min}$ ,  $\mu_{\max}$ ,  $v_H^C$  };
// puntos de partición BL de salida
pivot_y[0] = 0;
pivot_y[1] = pivot_y[0] +  $w^L (C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (\mu_{\min} - v_L^C)$ 
pivot_y[2] = pivot_y[1] +  $w^M (C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (\mu_{\max} - \mu_{\min})$ 
pivot_y[3] = pivot_y[2] +  $w^H (C_H^Y - C_L^Y) \frac{v_H^C - v_L^C}{v_H^Y - v_L^Y} (v_H^C - \mu_{\max})$ 

// si el número total de palabras de código supera 256, copiarlo en
si (pivot_y [3] > 255)
para (n = 0; n < 4; n++)
pivote [n] = clip3 (redondeo(pivote [n] * 255 / (pivot_y [3] - pivot_y [0])), 0, 255);
fin

```

En algunas formas de realización de ejemplo, se pueden utilizar uno o más cuantizadores lineales o no lineales para cuantizar datos de imagen de mayor profundidad de bit (p.ej., 12+ bits, etc.) a datos de imagen a una profundidad de bits más baja (p.ej., 8 bits, etc.). Se pueden seleccionar diferentes cuantizadores en diferentes espacios de color y/o en diferentes canales de color. A modo de ejemplo, para aliviar/reducir/eliminar artefactos de contorno (p.ej., en áreas lisas, etc.) y otros artefactos, las señales de vídeo se pueden cuantizar en diferentes espacios de color y/o con

diferentes métodos de cuantización. En algunas formas de realización, la cuantización tal como aquí se describe, puede comprender uno o más de entre: cuantización lineal, estiramiento lineal, cuantización optimizada basada en curva/cuantización no uniforme de función de densidad de probabilidad (Pdf), cuantización vectorial, etc. Los valores de parámetros optimizados generales se puede seleccionar sobre la base de cualquiera de: una trama, múltiples tramas, una escena, múltiples escenas, etc. En algunas formas de realización, un tipo específico de cuantización puede tener una relación correspondiente con uno o más tipos de métodos de predicción y/o mapeado inverso.

La cuantización puede realizarse en una base de canal individual o en dos o más canales al mismo tiempo. En una forma de realización ejemplo, la cuantización del vector se puede realizar a través de dos o más canales de color. A modo de ejemplo, un sistema de coordenadas (p.ej., Cartesiano 3D, etc.) se puede configurar utilizando canales de color en un espacio de color como ejes. La transformación espacial, como la rotación, se puede realizar en el sistema de coordenadas para crear nuevos ejes que se definen como combinaciones (o sumas de proyecciones) de los dos o más canales de color en el espacio de color. Palabras de código en los dos o más canales de color, proyectadas para formar uno de los nuevos ejes, pueden cuantizarse juntas mediante un cuantizador sobre uno de los nuevos ejes.

En algunas formas de realización, se puede seleccionar un método de cuantización específico en función de la eficacia de cómo puede comprimir datos de imagen VDR multicapa de salida manteniendo, al mismo tiempo, una alta calidad perceptual con los datos de imagen VDR de salida comprimida en el lado del decodificador VDR.

En algunas formas de realización, se puede seleccionar un método de cuantización específico y/o valores de parámetro óptimo para compensar las deficiencias de los códecs. A modo de ejemplo, un códec puede no funcionar bien en la compresión de áreas negras, e incluso puede generar artefactos de contorno en una imagen VDR reconstruida. La cuantización, según se describe aquí, puede utilizar una curva específica (p.ej., curva Sigmoidea, algoritmo Ley Mu, una curva basada en el sistema perceptual humano, etc.) para generar datos de imagen con menos artefactos de contorno visibles en una imagen VDR reconstruida.

Un codificador multicapa, según las técnicas que se describen en este documento, puede tomar datos de imagen VDR de entrada como la única entrada para el contenido de imagen que ha de procesarse por el codificador multicapa. Mientras que los datos de imagen VDR de entrada pueden proporcionarse para el procesamiento de datos de capa mejorada, la cuantización perceptual, que puede realizarse sobre la marcha (p.ej., a la misma velocidad de cable a la que se introduce VDR de entrada en el codificador VDR, etc.), se puede utilizar para generar datos de imagen de entrada para el procesamiento de datos de capa base, tal como aquí se describe.

La cuantización, según se describe en este documento, se puede realizar de uno o más modos operativos diferentes. La cuantización puede realizar una cuantización global en la que se cuantifica una trama completa, o una escena completa, utilizando una configuración única. La cuantización puede, además, realizar una cuantización (local) basada en partición, en la que cada trama está dividida en una pluralidad de zonas no solapadas, y cada zona no solapada es objeto de cuantización utilizando su propio ajuste. La cuantización puede realizar una cuantización (local) basada en partición en la que cada trama se divide en una pluralidad de zonas no solapadas y cada zona no solapada es objeto de cuantización utilizando su propio ajuste, pero se determinan las configuraciones del cuantizador para una zona específica no solapada, sobre la base de datos de análisis derivados de una o más zonas solapadas. La cuantización avanzada se puede aplicar en cualquiera de uno o más espacios de color diferentes. Los ejemplos de espacios de color en los que se puede aplicar la cuantización avanzada incluyen, pero no se limitan a cualquiera de: espacios de color RGB, espacios de color YCbCr, espacios de color YCoCg, espacios de color ACES u otros espacios de color.

En algunas formas de realización, un espacio de color en el que se aplica la cuantización se mantiene igual que un espacio de color en el que se realiza la predicción. Lo que antecede puede ser tanto en el proceso de codificación de imágenes VDR como en el proceso de decodificación de imágenes VDR. La transformación del espacio de color se puede realizar de forma adecuada si un espacio de color, en el que se produce la representación de la imagen es diferente de un espacio de color en el que se produce la cuantización.

## 9. EJEMPLO DE FLUJOS DE PROCESO

La Figura 3A ilustra un flujo de proceso ejemplo, de conformidad con una forma de realización ejemplo de la presente invención. En algunas formas de realización a modo de ejemplo, uno o más dispositivos o componentes informáticos pueden realizar este flujo de proceso. En el bloque 302, un codificador de vídeo VDR multicapa (p.ej., 102 de la Figura 1A) recibe una secuencia de imágenes de margen dinámico visual (VDR) de entrada.

En el bloque 304, el codificador de vídeo VDR multicapa (102) selecciona un conjunto candidato de valores de parámetro de función para una función de mapeado, a partir de una pluralidad de conjuntos candidatos de valores de parámetro de función, para la función de mapeado.

En el bloque 306, el codificador de vídeo VDR multicapa (102) construye un conjunto de bloques de imagen de desviaciones estándar distintas de cero en palabras de código VDR, en al menos una imagen VDR de entrada en la

secuencia de imágenes VDR de entrada.

En el bloque 308, el codificador de vídeo VDR multicapa (102) genera valores de código mapeados mediante la aplicación de la función de mapeado con el conjunto candidato de valores de parámetros de función a palabras de código VDR, en el conjunto de bloques de imagen, en al menos una entrada Imagen VDR.

En el bloque 310, el codificador de vídeo VDR multicapa (102) determina, en base a los valores de código mapeados, un subconjunto de bloques de imagen de desviaciones estándar por debajo de un valor umbral en palabras de código mapeadas. En este caso, el subconjunto de bloques de imágenes es un subconjunto en el conjunto de bloques de imágenes.

En el bloque 312, el codificador de vídeo VDR multicapa (102) determina, basándose, al menos en parte, en el subconjunto de bloques de imagen, si el conjunto candidato de valores de parámetros de función es óptimo, o no, para que la función de mapeado efectúe el mapeado de al menos una imagen VDR de entrada.

En una forma de realización, en respuesta a la determinación de que el conjunto candidato de valores de parámetros de función es óptimo para que la función de mapeado efectúe el mapeado de al menos una imagen VDR de entrada, el codificador de vídeo VDR multicapa (102) está configurado para determinar si el conjunto candidato de valores de parámetros de función se debe utilizar, o no, como un conjunto óptimo global de valores de parámetros de función mediante la función de mapeado para realizar el mapeado de una pluralidad de imágenes VDR de entrada que incluyen al menos una imagen VDR de entrada.

En una forma de realización, en respuesta a la determinación de que el conjunto candidato de valores de parámetros de función se debe utilizar como un conjunto óptimo global de valores de parámetros de función por la función de mapeado para realizar el mapeado de la pluralidad de imágenes VDR de entrada, que incluyen al menos una imagen VDR de entrada, el codificador de vídeo VDR multicapa (102) está configurado para realizar, además: la generación de una pluralidad de imágenes mapeadas que corresponden a la pluralidad de imágenes VDR de entrada, aplicando la función de mapeado con el conjunto óptimo global de valores de parámetros de función, para la pluralidad de imágenes VDR; la compresión de la pluralidad de imágenes mapeadas como datos de imagen de capa base (BL) en una señal de vídeo multicapa de salida.

En una forma de realización, el codificador de vídeo VDR multicapa (102) está configurado para realizar, además: la decodificación de los datos de imagen BL; la generación de datos de imagen de predicción basados, al menos en parte, en un mapeado inverso de los datos de imagen BL; la generación de valores residuales basados, al menos en parte, en los datos de imagen de predicción, y la al menos una imagen de VDR de entrada; la aplicación de una cuantización no lineal a los valores residuales con el fin de generar datos de imagen EL, comprendiendo los valores residuales valores altos de profundidad de bits, y los datos de imagen EL comprendiendo valores bajos de profundidad de bits; y la compresión de los datos de imagen EL en la señal de vídeo multicapa de salida.

En una forma de realización, el conjunto de bloques de imagen de desviaciones estándar distintas de cero, se calcula con las palabras de código VDR dentro de un margen de valor VDR específico, en una pluralidad de márgenes de valor VDR. La pluralidad de márgenes de valor VDR comprende uno o más márgenes de valor alto, márgenes de valor medio o márgenes de valor bajo. En una forma de realización, el codificador de vídeo VDR multicapa (102) está configurado para realizar, además: el cálculo de un conjunto de valores estadísticos de VDR para el conjunto de bloques de imagen de desviaciones estándar distintas de cero, en donde un valor estadístico VDR individual, en el conjunto de valores estadísticos VDR, representan uno de un promedio aritmético, una media aritmética, un promedio geométrico, una media geométrica, un máximo, o un mínimo, en palabras de código VDR de un bloque de imagen individual, en el conjunto de bloques de imagen de desviaciones estándar distintas de cero; y la generación, sobre la base del conjunto de valores estadísticos de VDR, de la pluralidad de márgenes de valor de VDR.

En una forma de realización, datos de imagen BL derivados de la secuencia de imágenes VDR de entrada, se comprimen mediante un primer codificador de 8 bits, en una señal de vídeo multicapa, y los datos de imagen EL, derivados de la secuencia de imágenes VDR de entrada, se comprimen mediante un segundo codificador de 8 bits, en el codificador multicapa, en la señal de vídeo multicapa.

La Figura 3B ilustra un flujo de proceso, a modo de ejemplo, de conformidad con una forma de realización ejemplo de la presente invención. En algunas formas de realización de ejemplo, uno o más dispositivos o componentes informáticos pueden realizar este flujo de proceso. En el bloque 352, un decodificador de vídeo VDR multicapa (p.ej., 152 de la Figura 1B) recibe datos de imagen de capa base (BL) de al menos una parte de una señal de vídeo multicapa.

En el bloque 354, el decodificador de vídeo VDR multicapa (152), o el decodificador de vídeo de capa base (172), decodifica los datos de imagen BL para generar una pluralidad de imágenes mapeadas.

En este caso, la pluralidad de imágenes mapeadas fue derivada aplicando una función de mapeado con un conjunto

óptimo global de valores de parámetros de función, a una pluralidad de imágenes de margen dinámico visible (VDR). El conjunto óptimo global de valores de parámetros de función se puede seleccionar a partir de una pluralidad de conjuntos óptimos individuales de valores de parámetros de función, para la función de mapeado. Cada conjunto óptimo individual de valores de parámetros de función, se determina basándose, al menos en parte, en la aplicación de la función de mapeado con una pluralidad de conjuntos candidatos de parámetros de función para al menos una imagen VDR, en la pluralidad de imágenes VDR.

En una forma de realización, el decodificador de vídeo VDR multicapa (152), o el decodificador de vídeo de capa base (172), está configurado para representar la pluralidad de imágenes mapeadas en un sistema de visualización.

En una forma de realización, el decodificador de vídeo VDR multicapa (152) está configurado para realizar, además: la generación de datos de imagen de predicción basados, al menos en parte, en un mapeado inverso de la pluralidad de imágenes mapeadas; la decodificación de datos de imagen EL a partir de la señal de vídeo multicapa; la aplicación de de-cuantización no lineal a los datos de imagen EL para generar valores residuales, comprendiendo los valores residuales valores altos de profundidad de bits, y comprendiendo los datos de imagen EL valores bajos de profundidad de bits; y la generación de al menos una imagen VDR sobre la base, al menos en parte, de los datos de imagen de predicción y los valores residuales. En una forma de realización, el decodificador de vídeo VDR multicapa (152) está configurado para reproducir al menos una imagen VDR en un sistema de visualización VDR.

En una forma de realización, la pluralidad de imágenes VDR de entrada, o la pluralidad de imágenes mapeadas, forman una escena.

En una forma de realización, el mapeado inverso, tal como aquí se describe, está basado en una o más tablas de consulta generadas a partir de la función de mapeado con el conjunto óptimo global de valores de parámetros de función.

En una forma de realización, una función de mapeado, según se describe aquí, representa al menos una de las funciones de potencia, funciones de cuantización lineal, o funciones de cuantización lineal por elementos.

En una forma de realización, un conjunto candidato de valores de parámetros de función, para la función de mapeado, tal como aquí se describe, representa un valor de exponente candidato para una función de potencia.

En una forma de realización, un conjunto candidato de valores de parámetros de función, para la función de mapeado, según aquí se describe, representa uno o más pivotes para una función de cuantización lineal por elementos.

En una forma de realización, un conjunto de bloques de imagen de desviaciones estándar distintas de cero, tal como se describe en este documento, se calcula con las palabras de código VDR para un canal específico, en una pluralidad de canales, de un espacio de color. La pluralidad de canales puede incluir uno o más de: un canal de luminancia, un canal cromático, un canal de color rojo, un canal de color azul, un canal de color verde u otros canales primarios. En una forma de realización, se utiliza una función de mapeado distinta para el mapeado de diferentes palabras de código VDR, para un canal diferente, en la pluralidad de canales.

En una forma de realización, al menos uno de uno o más codificadores, en un codificador de vídeo, según aquí se describe, comprende cualquiera de entre: codificadores de una profundidad de bits diferente (p.ej., codificador de 8 bits, codificador de 10 bits, codificador de 12 bits, etc.), un codificador de codificación de vídeo avanzado (AVC), un codificador de Grupo de Expertos de Imágenes en Movimiento (MPEG)-2, un codificador de Codificación de Vídeo de Alta Eficiencia (HEVC), etc.

En una forma de realización, al menos uno de uno o más codificadores, en un codificador de vídeo, según aquí se describe, comprende cualquiera de entre: codificadores de una profundidad de bits diferente (p.ej., codificador de 8 bits, codificador de 10 bits, codificador de 12 bits, etc.), un codificador de codificación de vídeo avanzado (AVC), un codificador de Grupo de Expertos de Imágenes en Movimiento (MPEG)-2, un codificador de Codificación de Vídeo de Alta Eficiencia (HEVC), etc.

En una forma de realización, imágenes VDR de entrada, o imágenes mapeadas, tal como se describe en este documento, están codificadas de forma perceptual.

En diversas formas de realización ejemplo, un codificador, un decodificador, un sistema, un aparato o uno o más de otros dispositivos informáticos, realiza cualquiera o una parte de los métodos anteriores tal como se describe.

## 10. MECANISMOS DE PUESTA EN PRÁCTICA - INFORMACIÓN GENERAL SOBRE DEL HARDWARE

De conformidad con una forma de realización, las técnicas descritas en la presente memoria se ponen en práctica por uno o más dispositivos informáticos de uso especial. Los dispositivos informáticos uso especial pueden estar cableados para realizar las técnicas, o pueden incluir dispositivos electrónicos digitales, tal como uno o más circuitos

integrados específicos de la aplicación (ASICs) o matrices de puerta programable en campo (FPGAs) que están programados continuamente para realizar la técnicas, o puede incluir uno o más procesadores de hardware de uso general programados para realizar las técnicas de conformidad con las instrucciones del programa en firmware, memoria, otro soporte de almacenamiento, o una combinación. Dichos dispositivos informáticos de uso especial pueden combinar, además, lógica de cableado personalizado, circuitos ASICs o matrices FPGA con programación personalizada para lograr las técnicas. Los dispositivos informáticos de uso especial pueden ser sistemas informáticos de escritorio, sistemas informáticos portátiles, dispositivos portátiles, dispositivos de red o cualquier otro dispositivo que incorpore lógica de programa y/o cableada para la puesta en práctica de las técnicas.

A modo de ejemplo, la Figura 4 es un diagrama de bloques que ilustra un sistema informático 400 sobre el que se puede poner en práctica una forma de realización ejemplar de la invención. El sistema informático 400 incluye un bus 402 u otro mecanismo de comunicación para la transmisión de información, y un procesador de hardware 404, acoplado con el bus 402, para procesar la información. El procesador de hardware 404 puede ser, a modo de ejemplo, un microprocesador de uso general.

El sistema informático 400 también incluye una memoria principal 406, tal como una memoria de acceso aleatorio (RAM) u otro dispositivo de almacenamiento dinámico, acoplado al bus 402 para memorizar información e instrucciones que han de ejecutarse por el procesador 404. La memoria principal 406 se puede utilizar, además, para memorizar variables temporales u otra información intermedia durante la ejecución de instrucciones que han de ponerse en práctica por el procesador 404. Dichas instrucciones, cuando se memorizan en soportes de almacenamiento no transitorios, accesibles para el procesador 404, convierten el sistema informático 400 en una máquina de uso especial que se personaliza para realiza las operaciones especificadas en las instrucciones.

El sistema informático 400 incluye, además, una memoria de solamente lectura (ROM) 408 u otro dispositivo de almacenamiento estático acoplado al bus 402, para memorizar información estática e instrucciones para el procesador 404. Un dispositivo de almacenamiento 410, tal como un disco magnético o disco óptico, es proporcionado y acoplado al bus 402 para la memorización de información e instrucciones.

El sistema informático 400 se puede acoplar a través del bus 402 a una pantalla 412, tal como una pantalla de cristal líquido, para mostrar información a un usuario informático. Un dispositivo de entrada 414, que incluye teclas alfanuméricas y otras, está acoplado al bus 402 para comunicar información y selecciones de órdenes al procesador 404. Otro tipo de dispositivo de entrada de usuario es el control de cursor 416, tal como un ratón, bola de seguimiento (Trackball), o teclas de dirección del cursor, para la comunicación de información de dirección y selecciones de órdenes al procesador 404 y para controlar el movimiento del cursor en la pantalla 412. Este dispositivo de entrada, normalmente, tiene dos grados de libertad en dos ejes, un primer eje (p.ej., x) y un segundo eje (p.ej., y), permite que el dispositivo especifique las posiciones en un plano.

El sistema informático 400 puede poner en práctica las técnicas descritas en este documento utilizando una lógica cableada personalizada, uno o más de entre un circuito ASICs o una matriz FPGAs, firmware y/o lógica de programa que, en combinación con el sistema informático, hace que el sistema informático 400 sea una máquina de uso especial. De conformidad con una forma de realización, las técnicas descritas en este documento se realizan mediante el sistema informático 400 en respuesta al procesador 404, ejecutando una o más secuencias de una o más instrucciones contenidas en la memoria principal 406. Dichas instrucciones pueden leerse en la memoria principal 406 desde otro soporte de memorización, tal como el dispositivo de almacenamiento 410. La realización de las secuencias de instrucciones, contenidas en la memoria principal 406, hace que el procesador 404 realice los pasos del proceso aquí descritas. En formas de realización alternativas, se puede utilizar circuitos cableados en lugar de, o en combinación con, instrucciones de software.

El término "soporte de almacenamiento" según aquí se utiliza, se refiere a cualquier soporte no transitorio que almacena datos y/o instrucciones que hacen que una máquina funcione de una manera específica. Dichos soportes de almacenamiento pueden incluir soportes no volátiles y/o soportes volátiles. Los soportes no volátiles incluyen, a modo de ejemplo, discos ópticos o magnéticos, tales como el dispositivo de almacenamiento 410. Los soportes volátiles incluyen una memoria dinámica, tal como la memoria principal 406. Las formas comunes de soportes de almacenamiento incluyen, a modo de ejemplo, un disquete, un disco flexible, disco duro, unidad de estado sólido, cinta magnética o cualquier otro soporte de almacenamiento de datos magnético, un CD-ROM, cualquier otro soporte de almacenamiento de datos óptico, cualquier medio físico con modelos de aberturas, una memoria RAM, una memoria PROM y una memoria EPROM, una memoria FLASH-EPROM, una memoria NVRAM, cualquier otro chip o tarjeta de memoria.

Los soportes de almacenamiento son distintos de, pero se pueden utilizar junto con, los medios de transmisión. Los medios de transmisión participan en la transferencia de información entre los soportes de almacenamiento. A modo de ejemplo, los medios de transmisión incluyen cables coaxiales, cables de cobre y fibra óptica, incluyendo los cables que componen el bus 402. Los medios de transmisión pueden tener, además, la forma de ondas acústicas o de luz, tal como las generadas durante las comunicaciones de datos por ondas de radio e infrarrojos.

Varias formas de soportes pueden estar implicadas en transmitir una o más secuencias, de una o más instrucciones,

al procesador 404 para su ejecución. A modo de ejemplo, las instrucciones pueden transmitirse, inicialmente, en un disco magnético o unidad de estado sólido de un ordenador distante. El ordenador distante puede cargar las instrucciones en su memoria dinámica y enviar las instrucciones a través de una línea telefónica utilizando un módem. Un módem local para el sistema informático 400 puede recibir los datos en la línea telefónica y utilizar un transmisor de infrarrojos para convertir los datos en una señal de infrarrojos. Un detector de infrarrojos puede recibir los datos contenidos en la señal de infrarrojos y los circuitos adecuados pueden colocar los datos en el bus 402. El bus 402 transmite los datos a la memoria principal 406, desde la cual el procesador 404 recupera y ejecuta las instrucciones. Las instrucciones recibidas por la memoria principal 406 pueden almacenarse, de forma opcional, en el dispositivo de almacenamiento 410, antes o después de la ejecución por el procesador 404.

El sistema informático 400 incluye, además, una interfaz de comunicación 418 acoplada al bus 402. La interfaz de comunicación 418 proporciona un acoplamiento de comunicación de datos bidireccional para un enlace de red 420 que está conectado a una red local 422. A modo de ejemplo, la interfaz de comunicación 418 puede ser una tarjeta de red digital de servicios integrados (RDSI), un módem de cable, módem satélite o un módem para proporcionar una conexión de comunicación de datos para un tipo de línea telefónica correspondiente. A modo de otro ejemplo, la interfaz de comunicación 418 puede ser una tarjeta de red de área local (LAN) para proporcionar una conexión de comunicación de datos a una red LAN compatible. Los enlaces inalámbricos también pueden ponerse en práctica. En cualquiera de dichas puestas en práctica, la interfaz de comunicación 418 envía y recibe señales eléctricas, electromagnéticas u ópticas que incluyen flujos de datos digitales, que representan diversos tipos de información.

El enlace de red 420 proporciona, normalmente, una comunicación de datos a través de una o más redes a otros dispositivos de datos. A modo de ejemplo, el enlace de red 420 puede proporcionar una conexión a través de la red local 422 a un host informático 424 o a un equipo de datos que se hace funcionar por un Proveedor de Servicios de Internet (ISP) 426. El ISP 426, a su vez, proporciona servicios de comunicación de datos a través de la red de comunicación de datos en paquetes a nivel mundial conocida comúnmente como la red "Internet" 428. La red local 422 e Internet 428 usan ambas señales eléctricas, electromagnéticas u ópticas que transmiten flujos de datos digitales. Las señales a través de las diversas redes, y las señales en el enlace de red 420, y a través de la interfaz de comunicación 418, que incluyen los datos digitales hacia y desde el sistema informático 400, son formas, a modo de ejemplo, de medios de transmisión.

El sistema informático 400 puede enviar mensajes y recibir datos, incluyendo código de programa, a través de la red o redes, el enlace de red 420 y la interfaz de comunicación 418. Tomando Internet a modo de ejemplo, un servidor 430 podría transmitir un código solicitado para un programa de aplicación a través de Internet 428, ISP 426, red local 422 e interfaz de comunicación 418.

El código recibido puede ser ejecutado por el procesador 404 a medida que se recibe, y/o memorizado en el dispositivo de almacenamiento 410, u otro almacenamiento no volátil para su posterior ejecución.

## 11. EQUIVALENTES, EXTENSIONES, ALTERNATIVAS Y DISPOSICIONES VARIAS

En la memoria descriptiva anterior, las formas de realización, a modo de ejemplo de la invención, se han descrito con referencia a numerosos detalles específicos que pueden variar de una puesta en práctica a otra. Por lo tanto, el único y exclusivo indicador de la idea inventiva de la invención, y el objetivo de los solicitantes es la invención y el conjunto de reivindicaciones que se emiten en esta solicitud en la forma específica en que se emiten dichas reivindicaciones, incluida cualquier corrección posterior. Cualquier definición expresamente establecida aquí para los términos contenidos en dichas reivindicaciones deberá regirse por el significado de tales términos según se utilicen en las reivindicaciones. Por lo tanto, ninguna limitación, elemento, propiedad, característica, ventaja o atributo que no se enumere expresamente en una reivindicación, debe limitar el alcance de dicha reivindicación en modo alguno. La especificación y los dibujos adjuntos son, en consecuencia, para ser considerados para fines ilustrativos y no en un sentido restrictivo.

## REIVINDICACIONES

1. Un método, que comprende:

5 la recepción de una secuencia de imágenes de entrada;

la selección de un conjunto candidato de valores de parámetros de función para una función de mapeado de correspondencia a partir de una pluralidad de conjuntos candidatos de valores de parámetros de función para la función de mapeado;

10 la división de cada imagen de entrada en bloques de imagen;

la determinación, para cada bloque de imagen, de una desviación estándar de los valores de píxel en ese bloque; la construcción de un conjunto de bloques de imagen, entre dichos bloques de imagen, que tienen una desviación estándar distinta de cero en los valores de píxel para al menos una imagen de entrada en la secuencia de imágenes de entrada;

15 la generación de bloques de imágenes mapeadas aplicando la función de mapeado con el conjunto candidato de valores de parámetros de función, para los valores de píxel en el conjunto de bloques de imagen en la al menos una imagen de entrada;

la determinación, para cada bloque de imagen mapeado, de una desviación estándar de los valores de píxel en ese bloque;

25 la construcción de un subconjunto de bloques de imágenes mapeadas entre los bloques de imágenes mapeadas, teniendo cada bloque de imágenes mapeadas, de dicho subconjunto, una desviación estándar mayor que un valor umbral predefinido;

30 la determinación de si un número total de bloques de imágenes mapeadas, con desviaciones estándar distintas de cero, en dicho subconjunto, es igual al número de bloques de imágenes en dicho conjunto, de modo que el conjunto candidato de valores de parámetros de función sea óptimo para que la función de mapeado para efectuar el mapeado de al menos una imagen de entrada;

35 en respuesta a la determinación de que el conjunto candidato de valores de parámetros de función es óptimo para que la función de mapeado realice el mapeado de al menos una imagen de entrada, la determinación de si el conjunto candidato de valores de parámetros de función debe utilizarse como un conjunto óptimo global de valores de parámetros de función, por la función de mapeado, para realizar el mapeado de una pluralidad de imágenes de entrada que incluyen la al menos una imagen de entrada; y

40 en respuesta a la determinación de que el conjunto candidato de valores de parámetros de función debe utilizarse como un conjunto óptimo global de valores de parámetros de función, por la función de mapeado, con el fin de realizar el mapeado de la pluralidad de imágenes de entrada que incluyen la al menos una imagen de entrada, la generación de una pluralidad de imágenes mapeadas que corresponden a la pluralidad de imágenes de entrada aplicando la función de mapeado, con el conjunto óptimo global de valores de parámetros de función, para la pluralidad de imágenes.

2. El método según la reivindicación 1, que comprende, además:

50 la compresión de la pluralidad de imágenes mapeadas como datos de imagen de capa base (BL) en una señal de vídeo multicapa de salida.

3. El método según la reivindicación 2, que comprende, además:

55 la decodificación de los datos de imagen BL;

la generación de datos de imagen de predicción basados, al menos en parte, en un mapeado inverso de los datos de imagen BL;

60 la generación de valores residuales sobre la base, al menos en parte, de los datos de imagen de predicción y la al menos una imagen de entrada;

la aplicación de cuantización no lineal a los valores residuales con el fin de generar datos de imagen de capa mejorada (EL), comprendiendo los valores residuales valores altos de profundidad de bits, y comprendiendo los datos de imagen EL valores bajos de profundidad de bits; y

65 la compresión de los datos de imagen EL en la señal de vídeo multicapa de salida.

**4.** El método según la reivindicación 3, que comprende, además, la salida de metadatos que incluyen el conjunto óptimo global determinado de valores de parámetros de función como una parte de la señal de vídeo multicapa de salida para un dispositivo de flujo descendente.

**5.** El método según la reivindicación 3 o la reivindicación 4, en donde el mapeado inverso de los datos de imagen BL se basa en una o más tablas de consulta generadas a partir de la función de mapeado con el conjunto óptimo global de valores de parámetros de función.

**6.** El método según cualquiera de las reivindicaciones 1-5, en donde una profundidad de bits, de las imágenes de entrada, es igual o mayor que 12 bits.

**7.** El método según cualquiera de las reivindicaciones 1-6, en donde la función de mapeado representa al menos una de las funciones de potencia, funciones de cuantización lineal, o funciones de cuantización lineal por elementos; y/o

en donde el conjunto candidato de valores de parámetros de función, para la función de mapeado, representa un valor de exponente candidato para una función de potencia; y/o

en donde el conjunto candidato de valores de parámetros de función, para la función de mapeado, representa uno o más pivotes para una función de cuantización lineal por elementos; y/o

en donde la secuencia de imágenes de entrada ha sido perceptualmente codificada; y/o

en donde al menos una de las desviaciones estándar distintas de cero se representa con una de entre las diferencias máximas-mínimas distintas de cero, las varianzas distintas de cero, o los valores de medición de uniformidad, cada uno de los cuales corresponde a una desviación estándar distinta de cero.

**8.** El método según cualquiera de las reivindicaciones 1-7, en donde el conjunto de bloques de imagen de desviaciones estándar distintas de cero se calcula con las palabras de código dentro de un margen de valores específico en una pluralidad de márgenes de valores.

**9.** El método según la reivindicación 8, en donde la pluralidad de márgenes de valores comprende uno o más márgenes de valores altos, márgenes de valores medios o márgenes de valores bajos, en donde los valores de píxeles en los márgenes de valores altos tienen valores más altos que los valores de píxel en los márgenes de valores medios y márgenes de valores bajos, y en donde los valores de píxel, en los márgenes de valores medios tienen valores más altos que los valores de píxel en los márgenes de valores bajos y valores más bajos que los valores de píxel en los márgenes de valores altos.

**10.** El método según la reivindicación 9, que comprende, además:

el cálculo de un conjunto de valores estadísticos para el conjunto de bloques de imágenes de desviaciones estándar distintas de cero, en donde un valor estadístico individual en el conjunto de valores estadísticos representa al menos uno de entre: un promedio aritmético, una media aritmética, un promedio geométrico, una media geométrica, un máximo, o un mínimo, en valores de píxel de un bloque de imagen individual en el conjunto de bloques de imagen de desviaciones estándar distintas de cero;

la generación, sobre la base del conjunto de valores estadísticos, de la pluralidad de márgenes de valores de píxel.

**11.** El método según cualquiera de las reivindicaciones 1-10,

en donde el conjunto de bloques de imagen de desviaciones estándar distintas de cero, se calcula con los valores de píxel para un canal específico en una pluralidad de canales de un espacio de color; y

en donde, en algunas formas de realización, la pluralidad de canales comprende uno o más de un canal de luminancia, un canal cromático, un canal de color rojo, un canal de color azul, un canal de color verde, u otros canales primarios, y/o

en donde, en algunas formas de realización, se utiliza una función de mapeado diferente para realizar el mapeado de diferentes valores de píxeles para un canal diferente en la pluralidad de canales.

**12.** El método según cualquiera de las reivindicaciones 1-11,

en donde los datos de imagen de capa base (BL), derivados de la secuencia de imágenes de entrada, se comprimen

mediante un primer codificador de 8 bits en una señal de vídeo multicapa, y en donde datos de imagen de capa mejorada (EL), derivados de la secuencia de imágenes de entrada, se comprimen por un segundo codificador de 8 bits en el codificador multicapa en la señal de vídeo multicapa; y

5 en donde, en algunas formas de realización, al menos uno de entre el primer codificador de 8 bits y el segundo codificador de 8 bits comprende al menos uno de: un codificador de codificación de vídeo avanzada (AVC), un codificador del Grupo de Expertos de Imagen en Movimiento (MPEG)-2, o un codificador de Codificación de Vídeo de Alta Eficiencia (HEVC).

10 **13.** Un método según la reivindicación 4 o cualquiera de las reivindicaciones 5-12, que están subordinadas a la reivindicación 4, que comprende, además:

la recepción, en un decodificador (152) del dispositivo de flujo descendente de la señal de vídeo de multicapa de salida, comprendiendo dicha señal de vídeo multicapa recibida, datos de imagen BL, datos de imagen EL y metadatos;

15 la decodificación de los datos de imagen BL comprendidos en la señal de vídeo multicapa recibida para generar una pluralidad de imágenes mapeadas;

20 caracterizado por

la generación de datos de imágenes de predicción utilizando un mapeado inverso de la pluralidad de imágenes mapeadas, basándose el mapeado inverso en el conjunto de valores de parámetros de función que se incluye en los metadatos (130);

25 la aplicación de la de-cuantización no lineal a los datos de imagen EL para generar valores residuales, comprendiendo los valores residuales valores más altos de profundidad de bits, y comprendiendo los datos de imagen EL, valores más bajos de profundidad de bits;

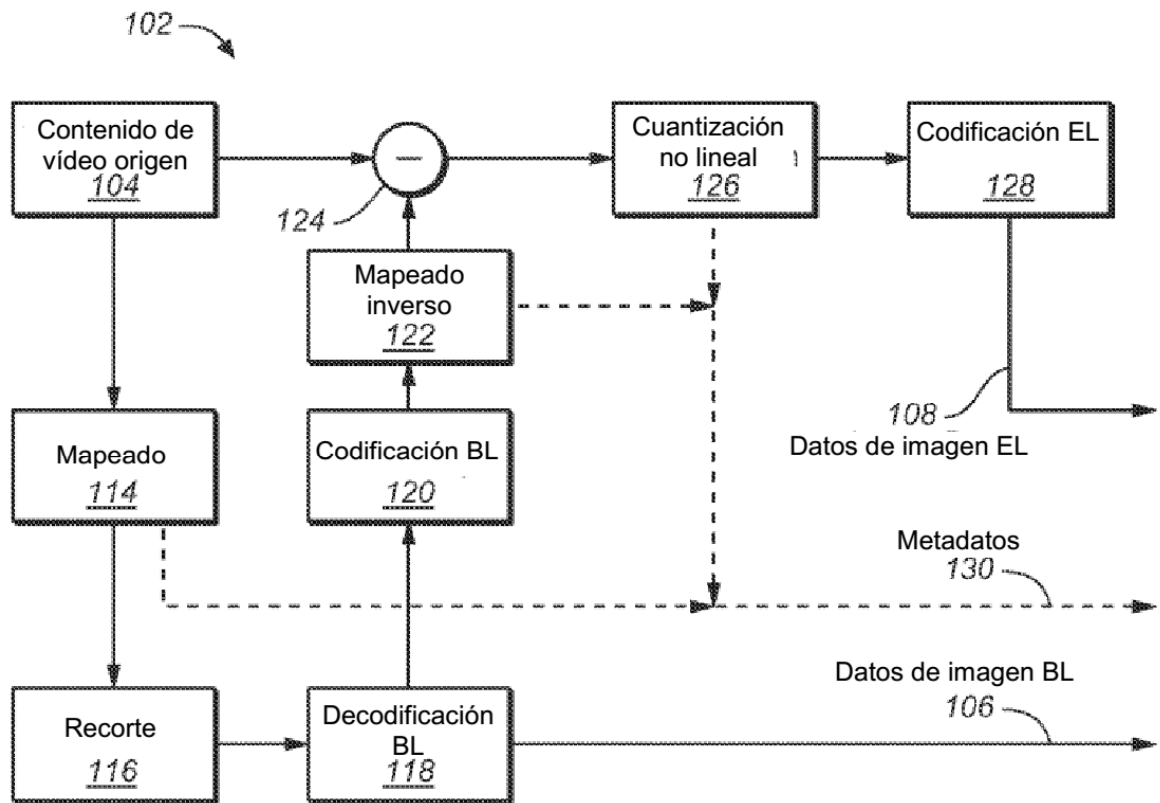
30 la generación de al menos una imagen utilizando los datos de imagen de predicción y los valores residuales;

en donde algunas formas de realización comprenden, además, la representación de la pluralidad de imágenes mapeadas en un sistema de visualización; y

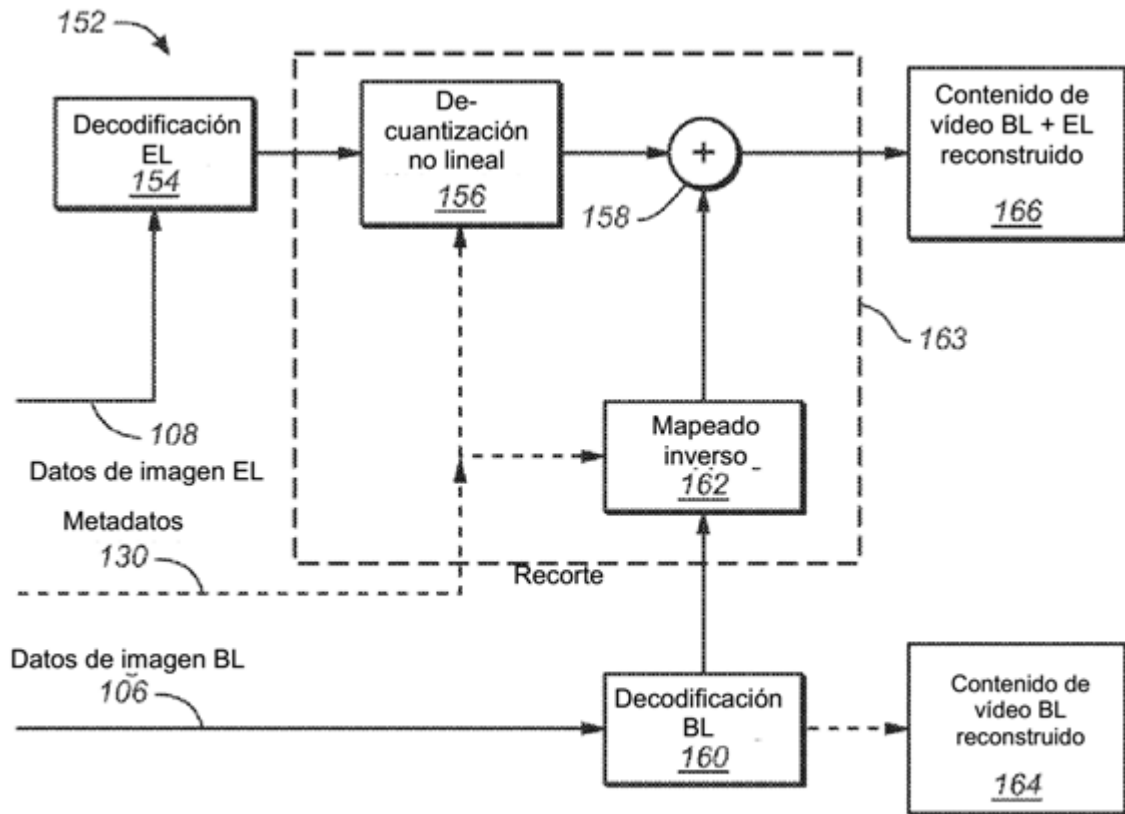
35 en donde algunas formas de realización comprenden, además, la presentación de la al menos una imagen en un sistema de visualización.

**14.** Un sistema que realiza cualquiera de los métodos descritos en las reivindicaciones 1-13.

40 **15.** Un soporte de almacenamiento legible por ordenador, que almacena instrucciones de software que, cuando se ejecutan por uno o más procesadores, causan el funcionamiento del método descrito en cualquiera de las reivindicaciones 1-13.



**FIG. 1A**



**FIG. 1B**

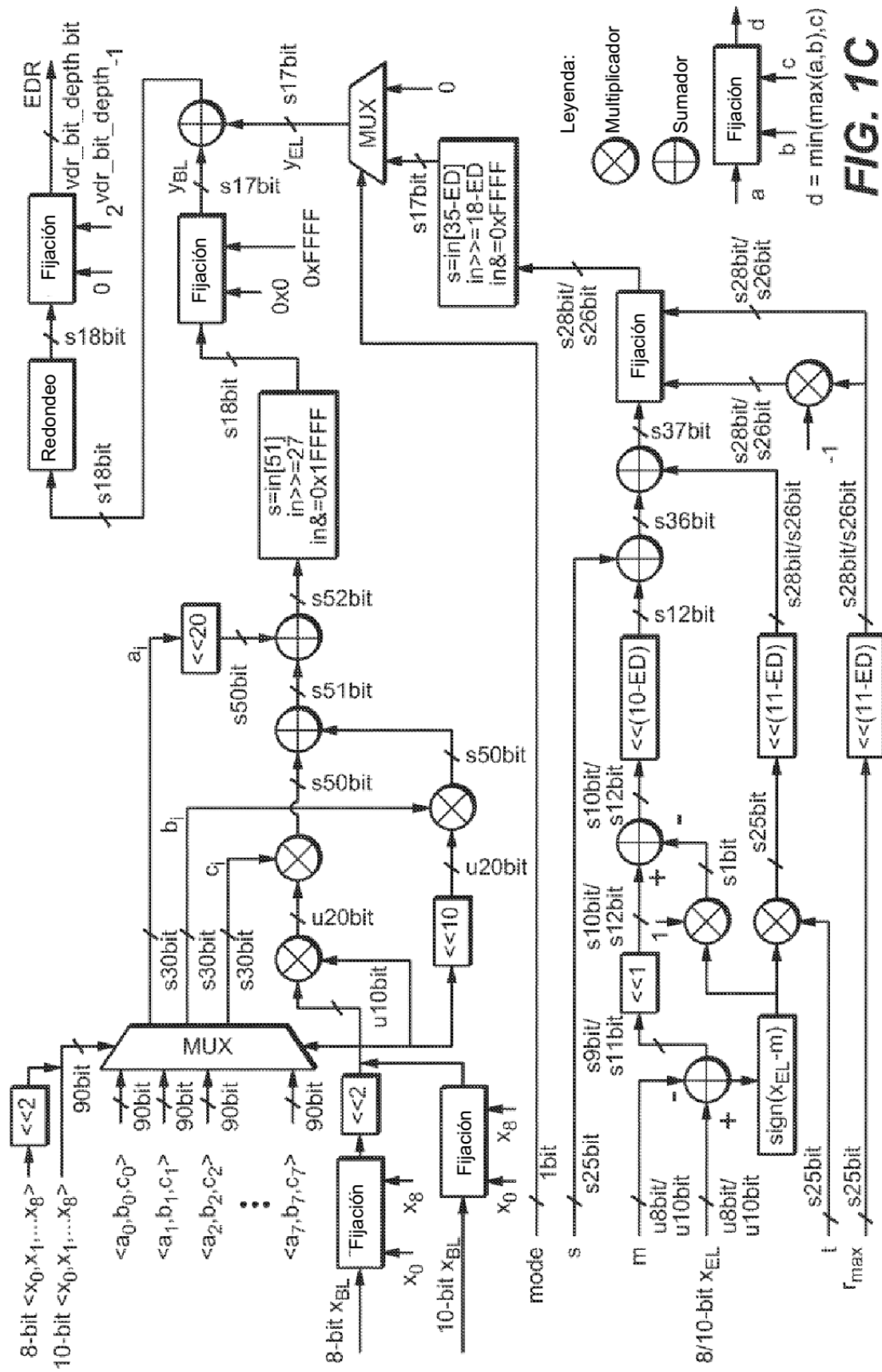


FIG. 1C

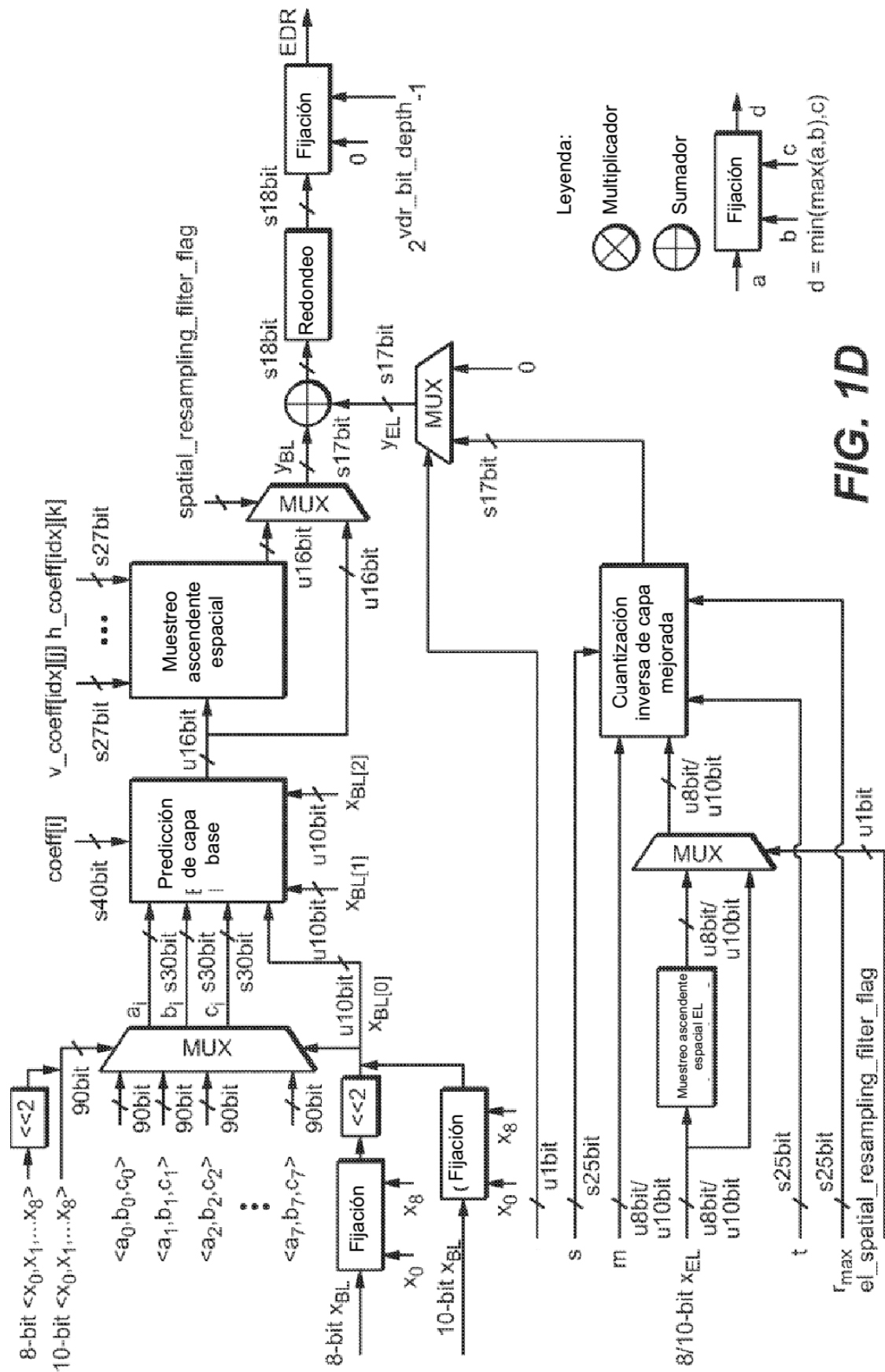


FIG. 1D

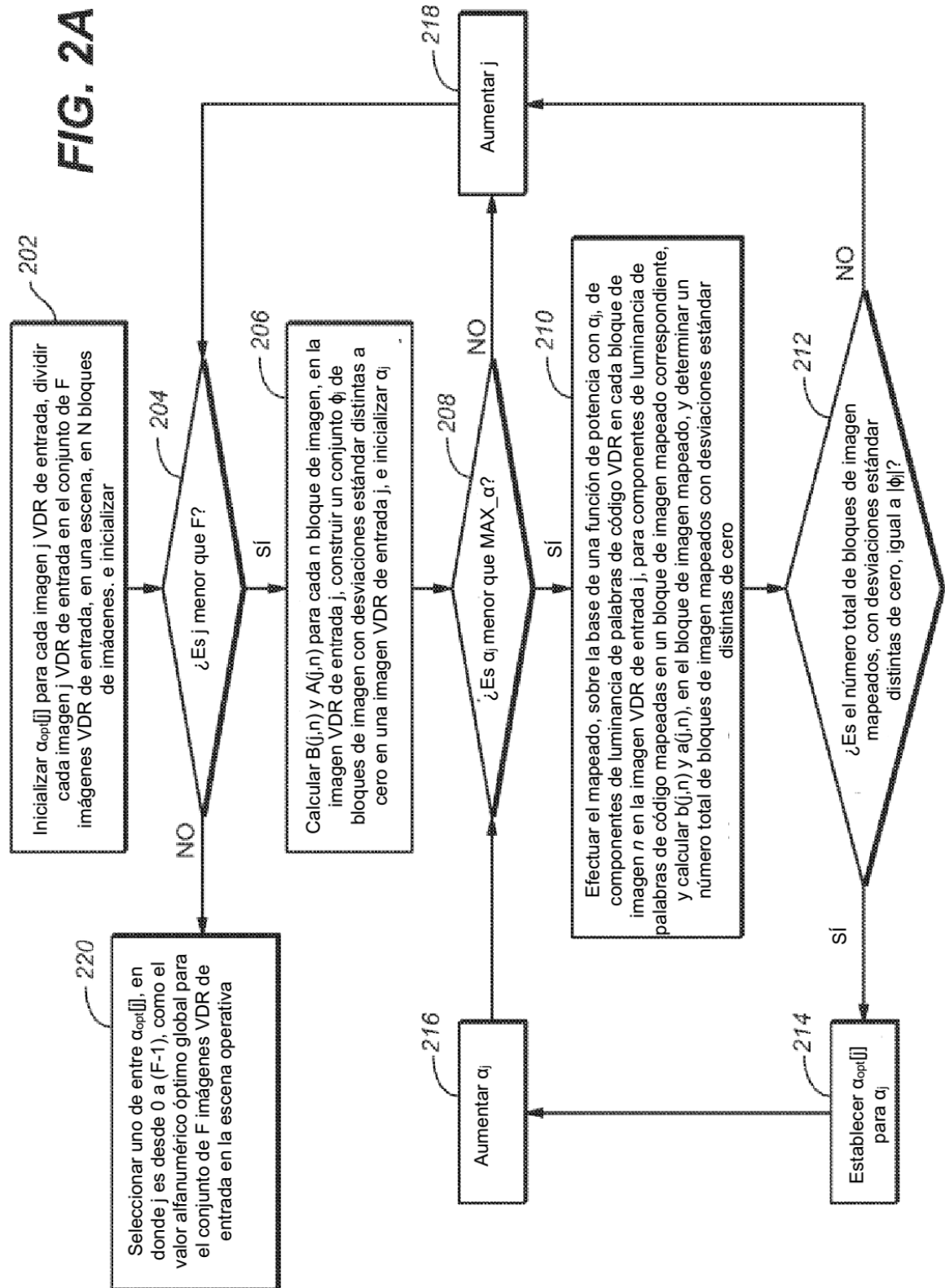
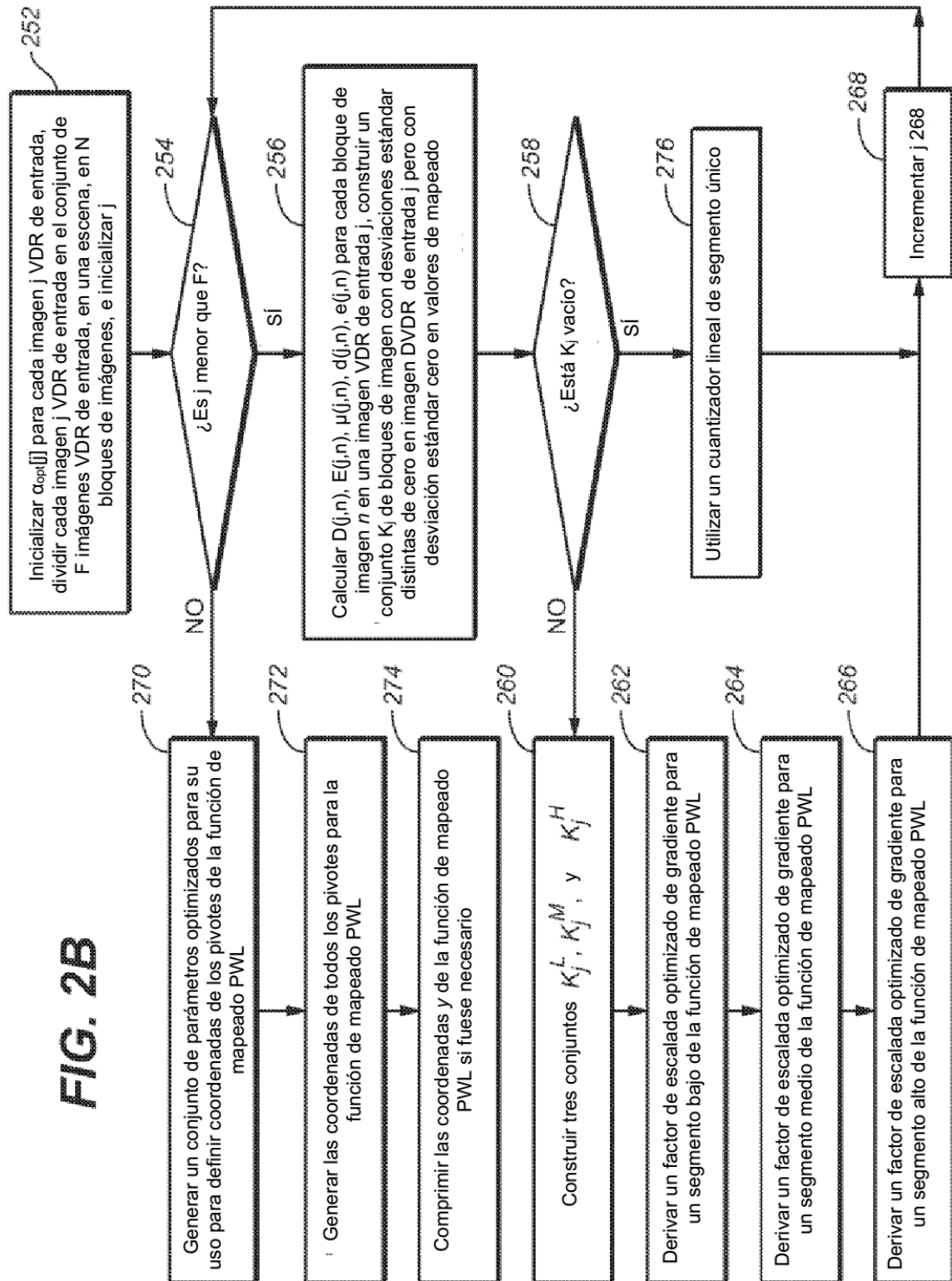
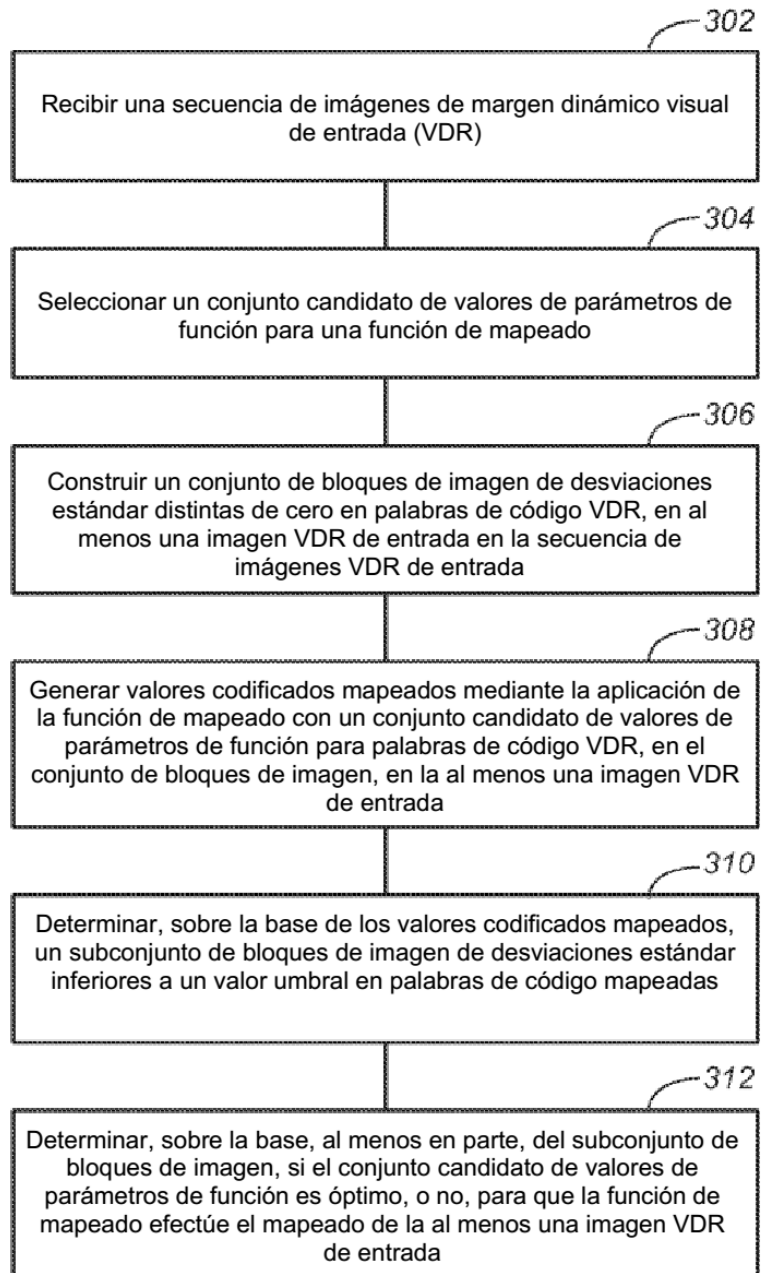


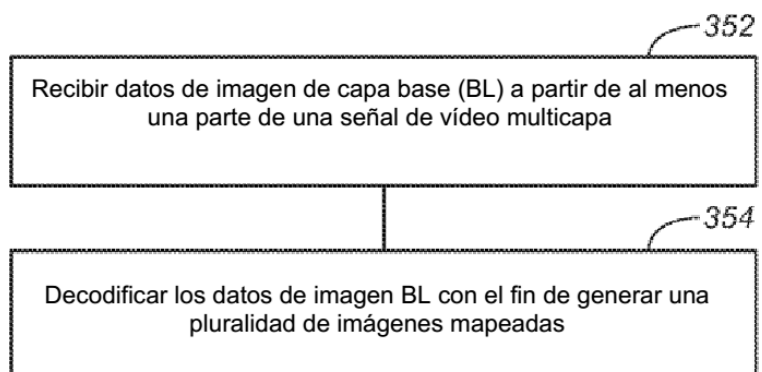
FIG. 2B

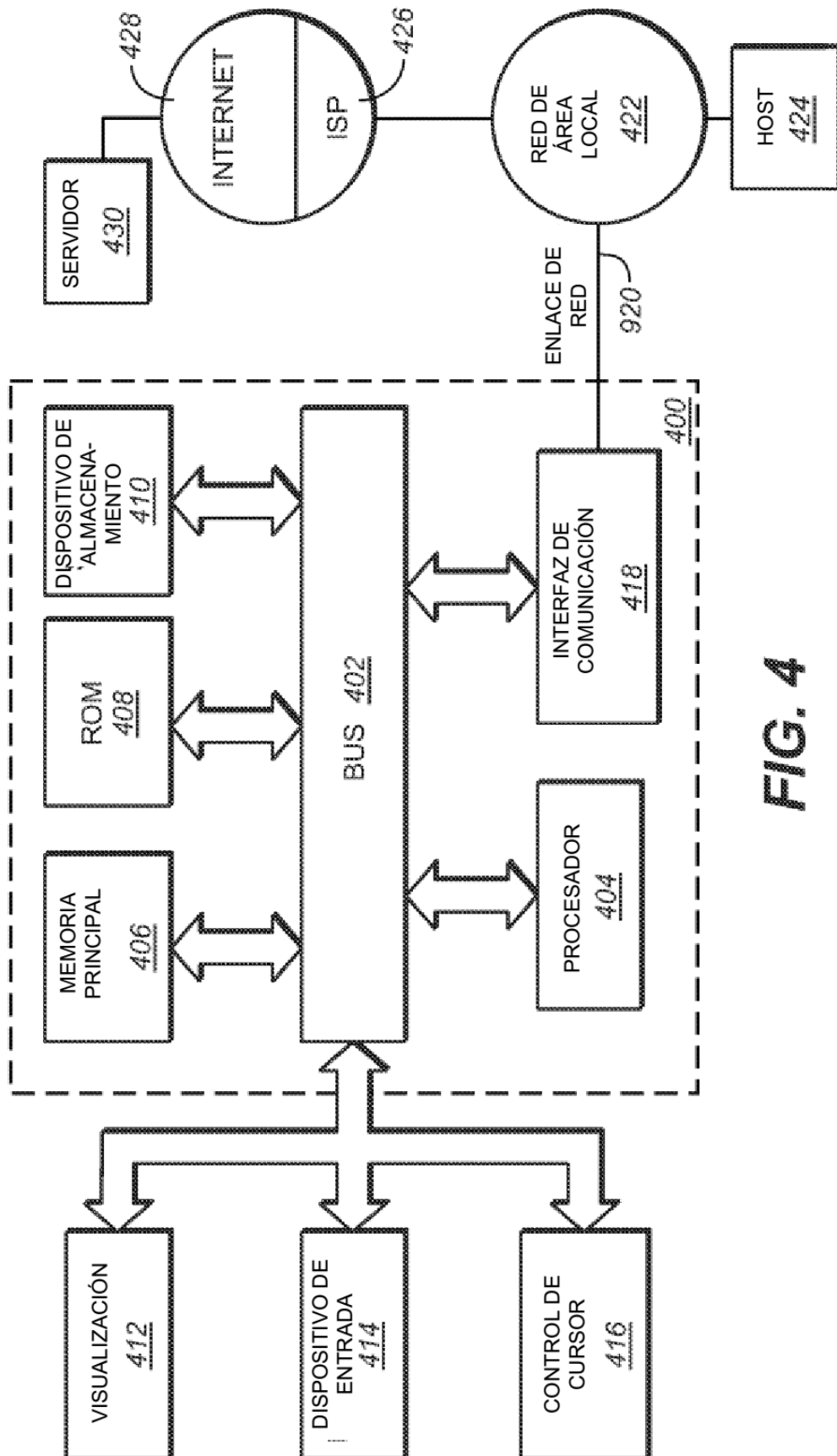


**FIG. 3A**



**FIG. 3B**





**FIG. 4**