

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 014 376**

51 Int. Cl.:

G09G 5/10 (2006.01)

G09G 5/02 (2006.01)

G06T 5/92 (2014.01)

H04N 1/60 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **16.04.2020** **PCT/US2020/028552**

87 Fecha y número de publicación internacional: **29.10.2020** **WO20219341**

96 Fecha de presentación y número de la solicitud europea: **16.04.2020** **E 20724336 (1)**

97 Fecha y número de publicación de la concesión europea: **19.02.2025** **EP 3959709**

54 Título: **Gestión de visualización para imágenes de alto rango dinámico**

30 Prioridad:

23.04.2019 US 201962837288 P

20.11.2019 US 201962938027 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
22.04.2025

73 Titular/es:

**DOLBY LABORATORIES LICENSING
CORPORATION (100.00%)**

**1275 Market Street
San Francisco, CA 94103, US**

72 Inventor/es:

**ATKINS, ROBIN;
PYTLARZ, JACLYN ANNE y
PIERI, ELIZABETH G.**

74 Agente/Representante:

ELZABURU, S.L.P

ES 3 014 376 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

Gestión de visualización para imágenes de alto rango dinámico

Referencia cruzada a solicitudes relacionadas

Esta solicitud reivindica la prioridad del documento de Solicitud de Patente Provisional de No. de serie 62/837.288, presentada el 23 de abril de 2019 y del documento de Solicitud de Patente Provisional de No. de serie 62/938.027, presentada el 20 noviembre de 2019.

Tecnología

La presente invención se refiere en general a las imágenes. Más concretamente, una realización de la presente invención se refiere a la gestión de visualización de imágenes y señales de vídeo de alto rango dinámico (High Dynamic Range, HDR).

Antecedentes

Tal como se utiliza en el presente documento, el término “rango dinámico” (Dynamic Range, DR) puede estar relacionado con una capacidad del sistema visual humano (Human Visual System, HVS) para percibir un rango de intensidad (por ejemplo, luminancia, luma) en una imagen, por ejemplo, desde los grises más oscuros (negros) hasta los blancos más brillantes (lucos). En este sentido, el DR está relacionado con una intensidad “referida a la escena”. El DR también puede estar relacionado con la capacidad de un dispositivo de visualización para reproducir de manera adecuada o aproximada un rango de intensidad de una amplitud particular. En este sentido, el DR está relacionado con una intensidad “referida a la pantalla”. A menos que se especifique explícitamente que un sentido particular tiene un significado particular en cualquier punto de la descripción del presente documento, se debe inferir que el término puede usarse en cualquier sentido, por ejemplo, de manera intercambiable.

Tal como se utiliza en el presente documento, el término alto rango dinámico (HDR) se refiere a una amplitud de rango dinámico (DR) que abarca los 14-15 órdenes de magnitud del sistema visual humano (HVS). En la práctica, el DR en donde un ser humano puede percibir simultáneamente una amplitud extensa en el rango de intensidad puede verse algo truncado en relación con el HDR. Tal como se utiliza en el presente documento, los términos rango dinámico mejorado (Enhanced Dynamic Range, EDR) o rango dinámico visual (Visual Dynamic Range, VDR) pueden estar relacionados de manera individual o indistinta con el DR que es perceptible dentro de una escena o imagen, por un sistema visual humano (HVS) que incluye movimientos oculares, lo que permite algunos cambios de adaptación a la luz en la escena o imagen.

En la práctica, las imágenes comprenden uno o más componentes de color (por ejemplo, luma Y y croma Cb y Cr), donde cada componente de color está representada por una precisión de n bits por píxel (por ejemplo, $n = 8$). Por ejemplo, utilizando la codificación de luminancia gamma, las imágenes donde $n \leq 8$ (por ejemplo, imágenes JPEG en color de 24 bits) se consideran imágenes de rango dinámico estándar, mientras que las imágenes donde $n \geq 10$ pueden considerarse imágenes de rango dinámico mejorado. Las imágenes EDR y HDR también pueden almacenarse y distribuirse utilizando formatos de punto flotante de alta precisión (por ejemplo, 16 bits), tal como el formato de archivo OpenEXR desarrollado por Industrial Light and Magic.

Tal como se utiliza en el presente documento, el término “metadatos” se refiere a cualquier información auxiliar que se transmite como parte del flujo de bits codificado y que ayuda a un decodificador a reproducir una imagen decodificada. Dichos metadatos pueden incluir, entre otros, información sobre el espacio de color o la gama de colores, parámetros de visualización de referencia y parámetros de señal auxiliar, tales como los que se describen en el presente documento.

La mayoría de las pantallas de escritorio para consumidores actualmente admiten una luminancia de entre 200 y 300 cd/m^2 o nits. La mayoría de los televisores de alta definición para consumidores tienen una resolución de entre 300 y 500 nits, y los modelos nuevos alcanzan los 1000 nits (cd/m^2). Por lo tanto, dichas pantallas convencionales tipifican un rango dinámico inferior (Lower Dynamic Range, LDR), también denominado rango dinámico estándar (Standard Dynamic Range, SDR), en relación con HDR o EDR. A medida que aumenta la disponibilidad de contenido HDR debido a los avances tanto en equipos de captura (por ejemplo, cámaras) como en pantallas HDR (por ejemplo, el monitor de referencia profesional PRM-4200 comercializado por la firma Dolby Laboratories), el contenido HDR puede ser graduado en color y mostrado en pantallas HDR que admitan rangos dinámicos más altos (por ejemplo, de entre 1000 nits y 5000 nits o más). En general, sin limitación, los métodos de la presente divulgación están relacionados con cualquier rango dinámico superior a SDR. Tal como se utiliza en el presente documento, el término “gestión de pantalla” se refiere a los procesos que se llevan a cabo en un receptor para reproducir una imagen para una pantalla de destino. Por ejemplo, y sin limitación, dichos procesos pueden incluir mapeo de tonos, mapeo de gama, gestión de color, conversión de velocidad de cuadros y similares.

La creación de contenido de alto rango dinámico (HDR) se está generalizando actualmente, ya que esta tecnología ofrece imágenes más realistas y reales que los formatos anteriores. Sin embargo, las características de una pantalla de destino pueden no coincidir con las características de la pantalla de referencia que se utilizó para realizar la

gradación de color del contenido original. Para mejorar los esquemas de visualización existentes, tal como aprecian los inventores en el presente documento, se han desarrollado técnicas mejoradas para la gestión de visualización de imágenes HDR.

- 5 El documento EP 3 248 367 describe un método para la gestión de visualización de imágenes con un rango dinámico alto o mejorado. El método implica calcular una métrica de saturación, un escalador de intensidad de mapeo de saturación (Saturation Mapping Intensity, SMI), un escalador de saturación de mapeo de tonos (Tone Mapping Saturation, TMS) y un escalador de saturación de mapeo de saturación (Saturation Mapping Saturation, SMS).

- 10 Los enfoques descritos en esta sección son enfoques que podrían implementarse, pero no necesariamente enfoques que hayan sido concebidos o implementados previamente. Por lo tanto, a menos que se indique lo contrario, no debe suponerse que ninguno de los enfoques descritos en esta sección está calificado como técnica anterior simplemente en virtud de su inclusión en esta sección. De manera similar, no debe suponerse que las cuestiones identificadas con respecto a uno o más enfoques hayan sido reconocidas en ninguna técnica anterior sobre la base de esta sección, a menos que se indique lo contrario.

Breve descripción de los dibujos

- 15 Una realización de la presente invención se ilustra a modo de ejemplo, y no de manera limitativa, en las figuras de los dibujos adjuntos y en donde números de referencia iguales se refieren a elementos similares y en donde:
- la FIG. 1 representa un proceso de ejemplo para una canalización de entrega de vídeo;
- la FIG. 2A y la FIG. 2B representan una arquitectura de ejemplo para la gestión de visualización, según una realización de la presente invención;
- 20 la FIG. 3 representa una canalización de procesamiento de ejemplo para una unidad analizadora de cuadros, según una realización de la presente invención;
- la FIG. 4 representa con más detalle el proceso de procesamiento de conversión de color para la unidad analizadora de cuadros de la FIG. 3, según una realización de la presente invención;
- 25 la FIG. 5 representa una canalización de procesamiento de ejemplo para el mapeo de pantalla, según una realización de la presente invención;
- la FIG. 6A y la FIG. 6B representan una canalización de procesamiento de ejemplo para el mapeo de volumen de color, según una realización de la presente invención;
- la FIG. 6C representa una canalización de procesamiento de ejemplo para ajustar los recortes de color, según una realización de la presente invención;
- 30 la FIG. 7A representa un proceso de ejemplo de generación de una imagen piramidal sobremuestreada, según una realización;
- la FIG. 7B representa un ejemplo de generación de una imagen de la capa de base, según una realización; y
- la FIG. 7C representa un ejemplo de filtrado de bordes, según una realización.

Descripción de ejemplos de realización

- 35 En este documento se describen métodos para la gestión de visualización de imágenes HDR. En la siguiente descripción, a los efectos de explicación, se exponen numerosos detalles específicos para proporcionar una comprensión completa de la presente invención. Sin embargo, será evidente que la presente invención puede ponerse en práctica sin estos detalles específicos. En otros casos, no se describen con detalle exhaustivo estructuras y dispositivos bien conocidos, para evitar ocluir, oscurecer u ofuscar innecesariamente la presente invención.

40 Compendio

La invención queda definida por las reivindicaciones adjuntas.

Gestión de visualización

Codificación de vídeo de señales HDR

- 45 La FIG. 1 representa un proceso de ejemplo de una secuencia de distribución de vídeo convencional (100) que muestra varias etapas desde la captura de vídeo hasta la visualización de contenido de vídeo. Se captura o genera una secuencia de cuadros de vídeo (102) utilizando un bloque de generación de imágenes (105). Los cuadros de vídeo (102) pueden capturarse digitalmente (por ejemplo, mediante una cámara digital) o generarse mediante un ordenador (por ejemplo, utilizando animación por ordenador) para proporcionar datos de vídeo (107). Alternativamente, los cuadros de vídeo (102) pueden capturarse en película mediante una cámara de cine. La película se convierte a un

formato digital para proporcionar datos de vídeo (107). En una fase de producción (110), los datos de vídeo (107) se editan para proporcionar un flujo de producción de vídeo (112).

Los datos de vídeo del flujo de producción (112) se proporcionan a continuación a un procesador en el bloque (115) para la edición de postproducción. La edición de postproducción del bloque (115) puede incluir el ajuste o la modificación de los colores o el brillo en áreas particulares de una imagen para mejorar la calidad de la imagen o lograr una apariencia particular para la imagen según la intención creativa del creador del vídeo. Esto a veces se denomina "sincronización de color" o "gradación de color". Se pueden realizar otras ediciones (por ejemplo, selección y secuenciación de escenas, recorte de imágenes, adición de efectos visuales especiales generados por ordenador, etc.) en el bloque (115) para obtener una versión final (117) de la producción para su distribución. Durante la edición de postproducción (115), las imágenes de vídeo se visualizan en una pantalla de referencia (125).

Después de la postproducción (115), los datos de vídeo de la producción final (117) pueden enviarse al bloque de codificación (120) para su envío posterior a dispositivos de decodificación y reproducción, tales como televisores, decodificadores, salas de cine y similares. En algunas realizaciones, el bloque de codificación (120) puede incluir codificadores de audio y vídeo, tales como los definidos por ATSC, DVB, DVD, Blu-Ray y otros formatos de entrega, para generar un flujo de bits codificado (122). En un receptor, el flujo de bits codificado (122) es decodificado por la unidad de decodificación (130) para generar una señal decodificada (132) que representa una aproximación idéntica o cercana de la señal (117). El receptor puede estar conectado a una pantalla de destino (140) que puede tener características completamente diferentes a las de la pantalla de referencia (125). En ese caso, se puede utilizar un bloque de gestión de pantalla (135) para mapear el rango dinámico de la señal decodificada (132) a las características de la pantalla de destino (140) generando una señal mapeada en la pantalla (137).

Canalización de gestión de visualización

Las FIG. 2A y FIG. 2B representan una arquitectura de ejemplo para la gestión de pantallas, según una realización. Tal como se representa en la FIG. 2A, el vídeo de entrada (132) puede incluir vídeo recibido de un decodificador de vídeo (130) y/o vídeo recibido de una unidad de procesamiento gráfico (por ejemplo, de un decodificador), y/u otra entrada de vídeo (por ejemplo, de un puerto HDMI en el televisor o el decodificador, una unidad de procesamiento gráfico (GPU) y similares). El vídeo entrante (132) puede ser procesado por el analizador (205), y su salida puede almacenarse en un búfer de cuadros (220). Tal como se explicará con más detalle en una sección posterior, el analizador analiza la entrada cuadro por cuadro y puede generar metadatos, datos estadísticos u otros datos relacionados con el vídeo que se utilizarán en etapas posteriores de la arquitectura. La salida del analizador también puede ser procesada por una unidad de submuestreo de pirámide (230) (que se describirá con más detalle más adelante), que genera una pirámide de imágenes (224). El analizador también puede generar una señal de solo intensidad (207) que se alimenta al búfer de cuadros.

Tal como se muestra en la FIG. 2B, el vídeo de entrada y los gráficos (226) del búfer de cuadros pueden procesarse mediante tres canalizaciones de vídeo: la canalización de vídeo principal, una canalización de imagen en imagen (Picture In Picture, PIP) opcional y una canalización de gráficos opcional. La canalización de vídeo principal incluye una unidad de sobremuestreo de pirámide (235) y puede incluir bloques de procesamiento tales como una unidad de mapeo de pantalla (210), circuitos de escalado, una unidad de eliminación de ruido y una unidad de gestión del color. De manera similar, la canalización PIP puede incluir su propio mapeo de pantalla (212) (normalmente, un subconjunto del mapeo de pantalla 210), escalado y conversión de velocidad de cuadros. Por último, la canalización de gráficos puede incluir mapeo de pantalla (212) y escalado. La salida de las tres canalizaciones se combina mediante la unidad compuesta (245) para generar la señal de vídeo compuesta 250, que, después de una gestión del color adicional y un postprocesamiento (no mostrado), se pasa al panel de la pantalla de destino (no mostrada). La arquitectura general puede controlarse mediante una capa de control de gestión de pantalla (215). A continuación se proporcionan detalles adicionales.

La unidad analizadora

Tal como se muestra en la FIG. 3, la unidad analizadora (205) analiza una fuente de entrada cuadro por cuadro y genera una variedad de salidas, que incluyen:

- Valores de metadatos mínimo, medio y máximo (310) para el marco (también pueden denominarse metadatos L1)
- Metadatos (310) relacionados con la atenuación global (también pueden denominarse metadatos L4), por ejemplo, metadatos L1 filtrados temporalmente; y
- Datos de intensidad (207) y un histograma de intensidad (320)

El analizador, en combinación con la unidad de submuestreo de pirámide (230), también puede generar metadatos relacionados con una representación piramidal del marco de entrada (132), también denominados metadatos de mapeo de precisión o de filtrado de pirámide (224).

En una realización, el analizador puede funcionar con una resolución reducida del cuadro de entrada para reducir la complejidad informática. Por ejemplo, en una realización, el analizador puede funcionar solamente con datos submuestreados a una resolución de 1024 x 576 o menor, tal como los genera un escalador 2D (260).

La FIG. 4 muestra detalles adicionales relacionados con la unidad analizadora. Tal como se muestra en la FIG. 4, la entrada de vídeo (402-a, 402-b o 402-c) se reduce de tamaño mediante la unidad de reducción de tamaño 405 (por ejemplo, utilizando el escalador 2D 260), que convierte todos los formatos de entrada a una representación 4:4:4 a una resolución más baja (por ejemplo, menor o igual que 1024 x 576). En una realización, la unidad de reducción de tamaño (405) puede aprovechar cualquier hardware de reducción de tamaño que pueda estar disponible en el decodificador. Después de la reducción de tamaño (405), una serie de etapas de conversión de color traducen el formato de color de entrada al formato de color ICtCp. Dichas etapas incluyen:

- Formato de entrada (por ejemplo, ICtCp, YCbCr, YUV y similares) a conversión RGB' no lineal (410)
- Conversión de RGB' no lineal a RGB lineal (415)
- Conversión de RGB lineal a LMS lineal (420)
- Conversión de LMS lineal a LMS' no lineal (425)
- LMS' no lineal a la componente I de ICtCp (430)
- De una primera representación RGB lineal a una segunda representación RGB lineal (435); y
- Conversión de RGB lineal a RGB no lineal (440)

Dependiendo del formato de color de entrada, se pueden omitir una o más de estas etapas. Se pueden encontrar ejemplos de transformaciones de color entre espacios de color para datos codificados tanto en PQ como en HLG en Rec. BT. 2100, "Image parameter values for high dynamic range television for use in production and international programme exchange", por la UIT.

Dados los valores RGB lineales (437) y los valores RGB' no lineales 442, el bloque de cálculo de metadatos 445 genera una variedad de valores de metadatos, tales como el valor de MinPixel (Pixel Mínimo), el valor de MaxPixel (Pixel Máximo), el valor de MeanPixel (Pixel Medio o promedio) y el StdPixel (Desviación Estándar) de píxeles en un marco, además de valores de metadatos adicionales, según sea necesario, para etapas posteriores en la gestión de visualización.

Por ejemplo, utilizando pseudocódigo, se puede utilizar el siguiente código para generar estos valores de metadatos.

Inicializar

```
MinPixel = 1
MaxPixel = 0
Scale = 1 / (Wa x Ha) //Dimensions of input image

m = 0,
m2 = 0,
```

Para cada posición de imagen [x, y]

```
if x >= ROIx1 & x <= ROIx2 & y >= ROIy1 & y <= ROIy2
//ROIx1, ROIx2, ROIy1 y ROIy2 definen la región de interés
{
    tempmin = min( R[ x, y ], G[ x, y ], B[ x, y ] ) >> 4
    tempmax = max( R[ x, y ], G[ x, y ], B[ x, y ] ) >> 4

    if tempmin < MinPixel
    {
        MinPixel = tempmin
    }
    if tempmax > MaxPixel
    {
        MaxPixel = tempmax
    }
    m = m + tempmax
    m2 = m2 + { tempmax } ^ 2
}
```

Después de procesar cada posición de imagen, calcular

```
MeanPixel    = m * Scale
StdPixel     = sqrt( ( m2 * Scale ) - ( m * Scale ) ^ 2 )
```

5 Dados estos metadatos, el filtro temporal (450) filtra los valores de MinPixel, MeanPixel, MaxPixel y StdPixel para generar valores de metadatos filtrados (452). Dicho filtrado temporal, tal como se describe en la Patente de US 10.242.627, "Backlight control and display mapping for high dynamic range images", por S. Farrell et al., tiene en cuenta los cambios de escena y limita los cambios repentinos en los valores de los metadatos, lo que puede dar como resultado una salida de pantalla deficiente, especialmente cuando la gestión de la pantalla tiene en cuenta la luz ambiental circundante. Como ejemplo, en una realización, el filtrado temporal para los metadatos de desviación estándar y media puede expresarse como

```
L4Alpha = CLAMP( 24/FPS *
                ( abs( FilteredFrameMean - MeanPixel ) * 8 + 0.1 ), 0, 1 )

FilteredFrameMean = FilteredFrameMean * ( 1 - L4Alpha ) + MeanPixel *
L4Alpha
10 FilteredFrameStd = FilteredFrameStd * ( 1 - L4Alpha ) + StdPixel * L4Alpha
```

donde FPS denota la velocidad de cuadros de la señal entrante (por ejemplo, 30 fps), L4Alpha denota un parámetro de filtrado y la función CLAMP(x , min , max) limita los valores de x entre min y max , tal como en:

```
CLAMP( $x$ ,  $min$ ,  $max$ ) = (  $x < min$  )?  $min$  : ( (  $x > max$  )?  $max$  :  $x$  )
```

Dados los datos de intensidad (I) (207), la unidad de cálculo de histograma 455 calcula su histograma 457.

15 Submuestreo de pirámide

Tal como se representa en la FIG. 2A, una realización incluye la unidad de submuestreo de pirámide 230, que genera una representación piramidal (224) del cuadro de entrada, que se utilizará más adelante para un mapeo de tonos mejorado. En algunas realizaciones, se pueden omitir capas de la pirámide para reducir el ancho de banda de la memoria. Por ejemplo, con una imagen de entrada de 4K, se podría omitir la primera capa (por ejemplo, con una resolución de 2K). Luego, durante el sobremuestreo, la imagen con un cuarto de resolución simplemente se duplicaría dos veces. Por ejemplo, dada una entrada de 4K, en una realización, la pirámide puede generar las siguientes capas: 1024 x 576, 512 x 288, 256 x 144, 128 x 72, 64 x 36, 32 x 18 y 16 x 9. De manera similar, para una imagen de entrada de 8K, se pueden omitir tanto las capas de media resolución como las de un cuarto de resolución. Esto garantiza que, independientemente del tamaño de la imagen de entrada, las capas posteriores de la pirámide tendrán las mismas dimensiones.

Aunque la pirámide se describe en términos de submuestreo utilizando los factores de submuestreo 2, 4, 8 y similares, se pueden utilizar otros factores de submuestreo sin pérdida de generalidad.

A modo de ejemplo, al crear la pirámide, la línea k -ésima en la capa n -ésima de la pirámide (por ejemplo, $n = 2$ a 7), se genera mediante el filtrado adecuado de las líneas $2*k$ y $2*k-1$ en la capa anterior. En una realización, dicho filtrado se realiza utilizando un filtro de paso bajo separable de 2×2 (por ejemplo, con coeficientes de filtro $[1 \ 1]/2$) o un filtro de paso bajo separable de 4×4 (por ejemplo, con coeficientes de filtro $[1 \ 3 \ 3 \ 1]/8$). El filtro de 4×4 da como resultado una mejor alineación entre los niveles de la pirámide, pero requiere búferes de línea adicionales. En otra realización, se pueden aplicar diferentes filtros en las direcciones horizontal y vertical, por ejemplo, un filtro horizontal de 4 tomas y un filtro vertical de 2 tomas, o viceversa.

35 Antes de calcular el primer nivel de la pirámide (por ejemplo, a 1024 x 576), la imagen ICTcp de entrada (507) se puede rellenar para:

- garantizar que todas las dimensiones espaciales sean divisibles por dos, desde los niveles más pequeños de la pirámide hasta los más grandes
- replicar píxeles de borde, teniendo en cuenta una región de interés (Region of Interest, ROI) específica
- 40 • replicar los píxeles del borde, teniendo en cuenta las imágenes de entrada de varios tamaños o relaciones de aspecto

Sobremuestreo de pirámide

El sobremuestreador de pirámide (235) recibe los datos de pirámide submuestreados (224-1, 224-2,..., 224-7) y reconstruye la imagen original en su resolución original utilizando, en cada capa, un filtro de sobremuestreo que tiene en cuenta los bordes. Primero se sobremuestra el nivel más pequeño de la pirámide y, a continuación, se sobremuestran los niveles adicionales, hasta la resolución del nivel más alto de la pirámide. El proceso general para una pirámide con siete capas se representa en la FIG. 7A.

Sea $P(i)$ la imagen piramidal en la capa i . Comenzando con el nivel de resolución más bajo (por ejemplo, $i = 7$), la imagen piramidal de resolución más baja (por ejemplo, $P(7)$) se alimenta a un filtro de conservación de bordes (715) que genera dos "imágenes" de coeficientes que se denotarán como $Ima(7)$ e $Imb(7)$. A continuación, ambas Ima e Imb son sobremuestreadas por un factor de dos para generar las imágenes de coeficientes sobremuestreadas $ImaU(7)$ e $ImbU(7)$ (720-7).

En la siguiente capa, $i = 6$, la capa $P(6)$ de la pirámide se combina con las imágenes de coeficientes sobremuestreadas $ImaU(7)$ e $ImbU(7)$ para generar la imagen

$$S(6) = ImaU(7) * P(6) + ImbU(7), \quad (1)$$

que, junto con la imagen $P(6)$ se introducen en el filtro de sobremuestreo de borde para generar las "imágenes" de coeficientes $Ima(6)$ e $Imb(6)$. A continuación, $Ima(6)$ e $Imb(6)$ son sobremuestreadas por un factor de dos para generar imágenes de coeficientes sobremuestreadas $ImaU(6)$ e $ImbU(6)$ (720-6). El mismo proceso continúa para las otras capas de la pirámide. En general, para $i = 7, 6, 5, \dots, 2$,

$$S(i-1) = ImaU(i) * P(i-1) + ImbU(i), \quad (2)$$

donde la operación de multiplicar una imagen de coeficiente por una imagen corresponde a multiplicar píxel por píxel sus píxeles correspondientes. Por ejemplo, en la ubicación del píxel (m, n) , para un nivel i de la pirámide con dimensiones $W(i) \times H(i)$,

$$S(i-1)_{m,n} = ImaU(i)_{m,n} * P(i-1)_{m,n} + ImbU(i)_{m,n}, \quad (3)$$

para $m = 1, 2, \dots, W(i-1)$ y $n = 1, 2, \dots, H(i-1)$.

Después de procesar el segundo nivel de la pirámide ($i = 2$), dados $S(1)$ y $P(1)$, el filtro de borde generará dos imágenes de parámetros $Ima(1)$ e $Imb(1)$ (710). Para generar una imagen 4K, $Ima(1)$ e $Imb(1)$ se puede ampliar por 2. Para generar una imagen de 8K, $Ima(1)$ e $Imb(1)$ se puede ampliar por 4. Tal como se muestra en la FIG. 7B, las dos imágenes de coeficientes ampliados ($ImaU(1)$ e $ImbU(1)$) (712), combinadas con la imagen de intensidad (I) del vídeo de entrada se puede utilizar para generar una imagen de la capa de base filtrada (730), como

$$RL = I_p = ImaU(1) * I + ImbU(1). \quad (4)$$

Tal como se describirá más adelante, esta imagen de la capa de base filtrada se puede utilizar para el mapeo de volumen de color para generar la imagen de salida final.

Generalizando, dada una pirámide con N capas de imágenes (por ejemplo, $P(1)$ a $P(N)$), generar las imágenes de coeficientes $Ima(1)$ e $Imb(1)$ comprende:

generar $Ima(N)$ e $Imb(N)$ utilizando un filtro de borde y $P(N)$, la imagen piramidal de N capas con la resolución espacial más baja;

generar $ImaU(N)$ e $ImbU(N)$ mediante ampliación de $Ima(N)$ e $Imb(N)$ para que coincida con la resolución espacial de la capa $N-1$;

para($j = N-1$ to 2) {

$$S(i) = ImaU(i+1) * P(i) + ImbU(i+1)$$

generar $Ima(i)$ e $Imb(i)$ utilizando el filtro de borde, $S(i)$ y $P(i)$

}

calcular $S(1) = ImaU(2) * P(1) + ImbU(2)$; y

generar $Ima(1)$ e $Imb(1)$ utilizando el filtro de borde, $S(1)$ y $P(1)$.

La FIG. 7C representa con más detalle las operaciones dentro del filtro de borde (715), que, dados los datos filtrados, $S(i)$, una capa $P(i)$ de la imagen piramidal, y pesos $PW(i)$ de la pirámide (proporcionados a través de metadatos), genera dos matrices de coeficientes $Ima(i)$ e $Imb(i)$ que se utilizarán para conservar los bordes en la capa superior de la pirámide. Téngase en cuenta que para la capa N de la pirámide con la resolución espacial más baja, por ejemplo, $N = 7$, $S(N) = P(N)$.

Tal como se representa en la FIG. 7C, en primer lugar, cada una de las entradas $S(i)$, $P(i)$, $P^2(i)$, y $P(i)*S(i)$ son convolucionadas horizontal y verticalmente utilizando un filtro de paso bajo separable de 3×3 , tal como $H = [1 \ 2 \ 1]/4$. Sus salidas correspondientes se pueden denotar como $Sout$ ($Ssalida$), $Pout$ ($Psalida$), $P2out$ ($P2salida$), y $PSout$ ($PSsalida$). Si bien estas señales son específicas de cada capa, para simplificar, el índice i no se utiliza. Entonces, tal como se muestra en la FIG. 7C,

$$\begin{aligned} Pvar &= P2out - Pout^2, \\ PScov &= PSout - Sout * Pout, \\ Ima(i) &= PScov / (Pvar + PW(i)), \\ Imb(i) &= Sout - Ima(i) * Pout. \end{aligned} \quad (5)$$

La unidad de mapeo de pantalla

Dada una imagen de entrada, representada en una pantalla de referencia (por ejemplo, 125), y una pantalla de destino (por ejemplo, 140), la función de la unidad de mapeo de pantalla es generar una imagen de salida para la imagen de destino que aproveche al máximo las características de la pantalla de destino pero que también preserve la intención artística de la imagen original.

La FIG. 5 muestra un diagrama de bloques de ejemplo de las componentes principales de una unidad de mapeo de pantalla, según una realización. Las entradas a la unidad de mapeo de pantalla incluyen: datos de imagen de vídeo y metadatos (502), información sobre las características de una pantalla de destino y preferencias de usuario (504) opcionales, y datos de mapeo de precisión opcionales relacionados con una representación piramidal de la imagen de entrada (por ejemplo, 224) y los coeficientes de filtro de borde de la capa uno de la pirámide (710), que se pueden utilizar para mejorar las operaciones de mapeo de tonos y mapeo de volumen de color generando en la unidad de generación de capa de base (750) una imagen de la capa de base (730) tal como se describió anteriormente y representada en la FIG. 7B. Dadas estas entradas, la unidad de mapeo de pantalla (500) genera una imagen mapeada (522) mediante un mapeo de tonos y/o un mapeo de color (510) adecuados.

En una realización, sin limitación, las operaciones relacionadas con el mapeo de tonos y el mapeo de color (510) se pueden realizar en el espacio de color ICtCp o en el espacio de color ITP. Dependiendo de los formatos de color de la entrada (502) y de la salida (522) deseados, operar en el espacio de color ICtCp puede requerir una conversión opcional del espacio de entrada (505) y una conversión del color de salida (520).

Por ejemplo, para la conversión de color de entrada (505), se pueden seguir los mismos pasos de conversión de color que los que se muestran en la FIG. 4. Por ejemplo, la conversión de color de YCbCr a ICtCp puede aplicar las siguientes etapas:

- YUV a RGB' no lineal (410)
- RGB' no lineal a RGB lineal (415)
- RGB lineal a LMS lineal (420)
- LMS lineal a LMS' no lineal (425); y
- LMS no lineal a ICtCp (507)

Para la conversión de color de salida, a modo de ejemplo, se pueden seguir las siguientes etapas de procesamiento:

- ICtCp a LMS' no lineal
- LMS' no lineal a LMS lineal
- LMS lineal a RGB lineal (522)
- RGB lineal a RGB' no lineal (el RGB' no lineal se puede utilizar para dispositivos que requieren una entrada RGB' no lineal)
- RGB' no lineal a YUV (o YCbCr) para dispositivos que requieren una entrada YCbCr

A modo de ejemplo, la Rec. BT. 2100 proporciona ejemplos de transformaciones de color recomendadas entre formatos de color para datos codificados tanto PQ como HLG.

- 5 En algunas realizaciones, es preferible que la componente I en $ICtCp$ esté dentro de $[0, 1]$ y que Ct y Cp estén dentro de $[-0,5, 0,5]$. Si los valores de entrada de $ICtCp$ están fuera de esos rangos, entonces pueden traducirse a estos rangos utilizando una operación simple de desplazamiento y escalado. Por ejemplo, en una realización, las operaciones pueden realizarse en el espacio de color "ITP". El espacio ITP es muy similar a la escala de color de $ICtCp$, excepto por algunas operaciones de desplazamiento y escalado, por ejemplo, la componente Ct se escala por dos para una representación interna más eficiente.

Filtrado de pirámide

- 10 En algunas realizaciones, que se describirán con más detalle más adelante, el mapeo de tonos de visualización dentro del bloque de mapeo de volumen de color (por ejemplo, 510) se puede mejorar aún más tomando en consideración el contraste local y la información de detalles de la imagen de entrada. Por ejemplo, se utiliza una imagen de intensidad filtrada para dividir la imagen en dos capas: una capa de base filtrada y una capa de detalle. Aplicando la curva de mapeo de tonos a la capa de base filtrada y luego agregando nuevamente la capa de detalle al resultado, se puede
15 conservar el contraste original de la imagen tanto global como localmente.

En una realización, esto se puede realizar como una operación de dos etapas:

- a) Generar una capa de base (Base Layer, BL) para guiar el mapeo
- b) Realizar el mapeo de tonos

- 20 En una realización, la capa de base generada representa una versión de la imagen original con bordes conservados y desenfocada espacialmente. Es decir, mantiene los bordes importantes pero desenfoca los detalles más finos. Más específicamente, la generación de la imagen de BL puede incluir:

- Crear una pirámide de imágenes con capas de menor resolución y guardar cada capa
- Comenzando con la capa de resolución más baja, realizar un sobremuestreo hacia las capas superiores para generar la capa de base.

- 25 Ya se han analizado ejemplos de realización para crear la pirámide de imágenes y la imagen de la capa de base como parte de la unidad de submuestreo de pirámide (230) y de sobremuestreo de pirámide (235).

Mapeo de volumen de color

- 30 Las FIG. 6A y FIG. 6B representan un ejemplo de canalización de procesamiento para mapeo de volumen de color (510), según una realización. El objetivo del bloque de mapeo de volumen de color es ajustar el rango de intensidades y colores para la presentación en una pantalla de destino y un entorno de visualización específicos. Las realizaciones de esta invención se basan en técnicas de gestión de pantalla anteriores con la adición de la operación de filtrado de pirámide opcional. Se realiza un nuevo algoritmo de conservación de detalles, que dio como resultado la nueva operación de "filtrado de pirámide" analizada anteriormente.

- 35 El mapeo del volumen de color es principalmente una función de cuatro operaciones que ajustan la intensidad y la saturación manteniendo el tono. *ToneMapI* (Mapeo de Tonos I) (615) y *SaturationMapI* (Mapeo de Saturación I) (660) realizan ajustes basados en la intensidad, y *ToneMapS* (642) y *SaturationMapS* (645) realizan ajustes basándose en la saturación. El tono, definido como el ángulo entre P (Cp) y T (Ct), permanece constante. Sin embargo, para permitir el mapeo local, *ToneMapI* ahora es una función tanto de la intensidad de entrada I como de la intensidad filtrada de la pirámide I_f . Además, para garantizar la precisión del color con el mapeo local, el *SaturationMapI* es ahora una función
40 tanto de la intensidad original I como de la intensidad con tonos mapeados I' .

$$\begin{aligned} S &= \sqrt{T^2 + P^2}, \\ I' &= \text{ ToneMapI}(I, I_f), \\ I'' &= I' * \text{ ToneMapS}(S), \\ P' &= P * \text{ SaturationMapI}(I, I') * \text{ SaturationMapS}(S), \\ T' &= T * \text{ SaturationMapI}(I, I') * \text{ SaturationMapS}(S), \end{aligned} \quad (6)$$

Una realización incluye un nuevo módulo de recorte de 6 vectores (655) que se describirá con más detalle más adelante.

- 45 Tal como se representa en la FIG. 6A, si según los metadatos de entrada parece que las características de la pantalla de destino coinciden con las características de la pantalla de referencia o la entrada (por ejemplo, Bypass 605 está configurado en Sí), entonces se puede omitir todo el mapeo de volumen de color, y la entrada (507) se pasa

directamente a la salida (652). De lo contrario, las componentes de intensidad (507-I) y croma (507-C) de la entrada se pasan a sus canalizaciones de mapeo de perspectiva. En algunas realizaciones, se puede realizar un procesamiento especial para una determinada región de interés (ROI) dentro de cada cuadro que se va a mostrar. Por ejemplo, una unidad de "determinación de ROI" (620) puede determinar si el píxel (x, y) de la imagen de entrada está dentro de una región de interés y si los parámetros 607 están configurados en un conjunto predeterminado si las operaciones están fuera de una ROI, o, si están dentro de la ROI, estos parámetros de control (607) se ajustan mediante metadatos de entrada. De manera similar, si está fuera de la ROI, entonces, tal como se muestra en la FIG. 6B, ciertas etapas de procesamiento pueden omitirse por completo para las regiones de píxeles fuera de la ROI. Por ejemplo, la unidad de ROI puede detectar el área de imagen dentro de las áreas de buzón que rodean un marco de entrada o una ventana superpuesta sobre la imagen activa. En una realización, sin limitación, la unidad de ROI puede marcar una región de píxeles como:

- Región primaria con recortes a mapear
- Región primaria sin recortes a mapear
- Región secundaria (por ejemplo, "buzón" o una superposición sobre un buzón) que se va a mapear, configurar en negro o ignorar

ToneMapI

El bloque ToneMapI (615) aplica un mapeo de tonos al canal de intensidad, en función de la intensidad I del píxel, así como una función del píxel filtrado de la pirámide I_F . La función *ToneMapI* puede comprender una variedad de etapas de procesamiento, incluida la reducción del rango dinámico y los ajustes de la luz ambiental.

Reducción del rango dinámico realiza la mayor parte de la operación de mapeo de tonos. Tal como se muestra en la FIG. 6C, la entrada I_B al mapeador de tonos se calcula primero interpolando entre la imagen original I (705) y la imagen filtrada de la pirámide I_F (730) para permitir cierto control sobre la fuerza del mapeo local:

$$I_B = \alpha * I_F + (1 - \alpha) * I. \quad (7)$$

Cuando $\alpha = 0$, el mapeo de tonos es equivalente a un mapeo de tonos global tradicional.

En una realización, la imagen I_B luego se asigna a intensidades de salida (I'_B) a través de una curva de mapeo de tonos que se calcula dinámicamente en función de los metadatos de origen y de las configuraciones de destino. Se describen ejemplos de dichas funciones de mapeo de tonos en la Patente de U.S. 8.593.480, "*Method and apparatus for image data transformation*", por A. Ballestad y A. Kostin, y en la Publicación de la WIPO WO 2018/152063, "*Tone curve mapping for high dynamic range images*", por J.A. Pytlarz y R. Atkins.

Dada la diferencia de imagen D (625), donde $D = (I - I_B)$, un escalador opcional β en la imagen D se puede utilizar para ajustar la nitidez de la salida con tonos mapeados

$$I' = I'_B + (I - I_B) * \beta, \quad (8)$$

donde I'_B denota una versión con tonos mapeados de I_B .

En teoría, este algoritmo aplica la operación de mapeo de tonos solo a grandes regiones espaciales de la imagen, dejando las regiones espaciales pequeñas sin modificar. El resultado es que se puede conservar parte del contraste local original de la imagen en las regiones de luces y sombras. Cuando no ha habido mapeo de tonos, la operación no tiene efecto, ya que no hay razón para restaurar el contraste. Para regiones planas grandes, la operación tampoco tiene efecto. Solo se trata de pequeñas regiones de la imagen con mapeo de tonos en donde la operación tiene como objetivo preservar el contraste. Para evitar halos no deseados, el algoritmo de mapeo local tiene un efecto mínimo cerca de bordes de gran magnitud, donde se pueden realizar grandes cambios de luminancia sin causar halos visibles. El creador de contenido tiene cierto grado de control sobre la cantidad de algoritmo de mapeo local que se aplica, que varía entre ninguna y máxima intensidad.

La curva de tono también se puede modificar en función de la intensidad del mapeo local. La pendiente de las partes superior e inferior de la curva de mapeo se puede ajustar para que se acerque más a una pendiente = 1 con mapeo local y combinación en función del valor de α . Esto garantiza que haya suficiente espacio para agregar detalles sin cortes ni aplastamientos.

La **corrección de la luz ambiental** ajusta la luminancia de salida en función del entorno de visualización para mantener la visibilidad de los detalles en las regiones oscuras. Esto normalmente implica aumentar el contraste en las regiones oscuras, con una reducción correspondiente del contraste en otras partes de la imagen, si es necesario. El algoritmo detrás del módulo de compensación de luz ambiental se basa en el cálculo de una curva PQ diferente en función del entorno ambiental. En niveles más altos de adaptación, las diferencias apenas perceptibles (Just Noticeable Differences, JND) se vuelven más grandes en las regiones oscuras. Al calcular una nueva curva PQ que modeliza estas JND más grandes, el algoritmo de luz ambiental puede preservar la cantidad de escalones visibles

entre los distintos grados de adaptación al entorno circundante. Además de compensar la iluminación circundante, este módulo también compensa los reflejos estimados de la pantalla. La luz ambiental se puede determinar a partir de uno o más sensores de luz ambiental instalados en la pantalla.

5 En la publicación de la WIPO WO 2018/119161, "Ambient light-adaptive display management", de J.A. Pytlarz et al., y la Solicitud de Patente Provisional de No. de serie 62/790.058, presentada el 9 de enero de 2019, "Display management with ambient light compensation" de R. Atkins et al, se describen ejemplos de este tipo de mapeo. Esta etapa también se puede utilizar para ajustar las preferencias del usuario, tales como el ajuste de contraste, el ajuste de brillo y similares.

10 Volviendo a la FIG. 2B, el mapeo de pantalla (210) puede incluir tanto el mapeo de tonos global como el mapeo de precisión, mientras que el mapeo de pantalla (212) puede incluir solo el mapeo de tonos global.

Todas estas etapas de procesamiento están controladas por indicadores y parámetros (607), proporcionados ya sea por la capa de control de gestión de pantalla (215) o por metadatos incluidos en el flujo de bits.

SaturationMapI

15 El bloque SaturationMapI (660) aplica un mapeo de saturación a los canales de croma (por ejemplo, T y P), en función de la intensidad del píxel y de la salida de ToneMapI (615). El mapeo se aplica en tres etapas:

$$\begin{aligned} \text{SaturationMapI} \\ &= \text{SaturationScale} * \text{MesopicPreservation} \\ &\quad * \text{SaturationTrim} \end{aligned}$$

SaturationScale (Escala de Saturación)

MesopicPreservation (Preservación Mesópica)

SaturationTrim (Recorte de Saturación)

20 **Etapas 1: Escala de preservación de la saturación**, ajusta la saturación para tener en cuenta el cambio de intensidad. A medida que se reduce la intensidad de los colores, la saturación también disminuye en consecuencia para mantener la apariencia o el equilibrio del color. La cantidad para ajustar la saturación se calcula como una relación que depende de la intensidad antes y después del mapeo de tonos. Téngase en cuenta que una aplicación adicional de la función CLAMP y los límites se han omitido para mayor claridad.

$$25 \quad \text{SaturationScale} = \frac{(I'-1)^3 + 1}{(I-1)^3 + 1} \quad (9a)$$

En general, la ecuación (9) puede expresarse como

$$\text{SaturationScale} = \frac{f_s(I')}{f_s(I)}, \quad (9b)$$

30 donde la función $f_s(I)$ es una función de intensidad que aumenta monótonamente, preferiblemente con un mapeo lineal para los tonos medios, pero un mapeo no lineal para los tonos oscuros y brillantes, similar a una función sigmoidea tradicional.

35 **Etapas 2: Preservación mesópica y escala de aplastamiento suave**, ajusta aún más la saturación asignando un nivel de negro más alto. La preservación mesópica (c_4) modeliza la pérdida perceptiva de saturación de color en píxeles de baja luminancia en el contenido de origen y la reproduce en el nivel de luminancia más alto. Sin esta modificación, los detalles oscuros pueden aparecer más saturados cuando se asignan a una pantalla con un nivel de negro más alto. El efecto de compresión suave (c_2) es un ajuste de saturación de mapeo de gama. Reduce la saturación de los píxeles que están cerca del mínimo de la pantalla de destino para que vuelvan a estar dentro de la gama. Sin esta modificación, los píxeles oscuros pueden quedar fuera de la gama y pueden recortarse en colores altamente saturados que no son aceptables. Téngase en cuenta que una aplicación adicional de la función CLAMP y los límites se han omitido para mayor claridad.

$$\begin{aligned}
 c_1 &= \frac{I' - T_{min}}{T_{min} - S_{min}}, \\
 c_2 &= \text{sign}(c_1 - 1) * (c_1 - 1)^2 + 1, \\
 c_3 &= \frac{I * 4095 - 25}{900 - 25}, \\
 c_4 &= \text{sign}(c_3 - 1) * (c_3 - 1)^2 + 1, \\
 \text{MesopicPreservation} &= c_2 * c_4.
 \end{aligned} \tag{10}$$

Etap 3: Escala de recorte de saturación, permite que un creador de contenido modifique aún más la saturación a partir del mapeo automático para lograr mejor su intención creativa. Esto simplemente se calcula a partir del valor de *SaturationGain* proporcionado en los metadatos, que, en una realización, tiene un rango en [-0,5, 0,5] y un valor típico de aproximadamente 0,05.

$$\text{SaturationTrim} = 1 + \text{SaturationGain}. \tag{11}$$

ToneMapS

El bloque ToneMapS (640) ajusta el canal de intensidad (*I*) de la entrada con tonos mapeados en función de la saturación de los canales de croma. La cantidad de ajuste se controla mediante dos etapas determinadas por dos coeficientes c_1 y c_2 . Téngase en cuenta que una aplicación adicional de la función CLAMP se ha omitido para mayor claridad.

$$\begin{aligned}
 S &= \sqrt{T^2 + P^2}, \\
 \text{ToneMapS} &= (1 + S^3 * c_1) * (1 + S * c_2),
 \end{aligned} \tag{12}$$

Téngase en cuenta que, en otras realizaciones, *ToneMapS* puede denotar cualquier función monótonamente creciente de *S* para $S > 0$.

En una realización, los coeficientes c_1 y c_2 se calculan de la siguiente manera.

Etap 1: Saturación primaria, modifica el brillo de los colores saturados según la diferencia entre los primarios de las pantallas de origen y destino. Esto hace que los colores muy saturados se oscurezcan ligeramente, lo que ayuda a preservar la saturación del color y las texturas en las regiones de color brillantes y saturadas. La saturación del primario rojo de la pantalla de origen se calcula primero, utilizando el espacio de color ITP. Luego se calcula un coeficiente c_1 como una relación entre la intensidad del rojo primario del objetivo y la fuente, teniendo en cuenta el mapeo de tonos adicional realizado por *ToneMapI*. El coeficiente c_1 además, se escala mediante la saturación al cubo del rojo primario, de modo que cada píxel se represente como una relación normalizada de la saturación del píxel con la saturación del rojo primario. Téngase en cuenta que la implementación real incluye una aplicación adicional de la función CLAMP para limitar el rango de cada uno de los valores y evitar divisiones por cero; las ecuaciones a continuación omiten la aplicación de la función CLAMP para mayor claridad.

$$\begin{aligned}
 S_{RedS} &= \sqrt{T_{RedS}^2 + P_{RedS}^2}, \\
 c_1 &= \left(\frac{I_{RedT}}{\text{ToneMapI}(I_{RedS})} - 1 \right) * \frac{1}{S_{RedS}^3}.
 \end{aligned} \tag{13}$$

Etap 2: Recorte de ChromaWeight (Peso del croma) permite a un creador de contenido modificar aún más el brillo de los colores saturados a partir de la asignación automática para lograr mejor su intención creativa. Esto se calcula simplemente escalando el valor de *ChromaWeight* proporcionado en los metadatos.

$$c_2 = \text{ChromaWeight} * 4. \tag{14}$$

ChromaWeight varía en [-0,5, 0,5], con un valor típico de -0,05.

SaturationMapS

El bloque SaturationMapS (645) aplica un mapeo de saturación a los canales *P* y *T*, en función de la saturación de los canales de croma. La cantidad para ajustar la saturación se controla mediante dos etapas, determinadas por dos coeficientes. c_1 y c_2 . Téngase en cuenta que se ha omitido una aplicación adicional de la función CLAMP para mayor claridad.

$$S = \sqrt{T^2 + P^2}$$

$$SaturationMapS = (1 + S^3 * c_1) * (1 + S * c_2). \quad (15)$$

En una realización, los coeficientes c_1 y c_2 se calculan de la siguiente manera.

Etapla 1: Saturación primaria, modifica la saturación según la diferencia entre los colores primarios de las pantallas de origen y destino. Esto hace que los colores muy saturados se desaturen ligeramente, lo que ayuda a conservar el tono y las texturas en las regiones de color brillantes y saturadas. La saturación del rojo primario de la pantalla de origen y las pantallas de destino se calcula primero, utilizando el espacio de color ITP. A continuación, se calcula un coeficiente c_1 como una relación de la saturación del primario rojo del objetivo y la fuente, teniendo en cuenta la desaturación adicional realizada por *SaturationMapI*. El coeficiente c_1 además, se escala mediante la saturación al cubo del rojo primario, de modo que cada píxel se represente como una relación normalizada de la saturación del píxel con la saturación del rojo primario. Téngase en cuenta que la implementación real incluye una aplicación adicional de la función CLAMP para limitar el rango de cada uno de los valores y evitar divisiones por cero; las ecuaciones a continuación omiten la aplicación de la función CLAMP para mayor claridad.

$$\begin{aligned} S_{RedS} &= \sqrt{T_{RedS}^2 + P_{RedS}^2}, \\ S_{RedT} &= \sqrt{T_{RedT}^2 + P_{RedT}^2}, \\ c_1 &= \left(\frac{S_{RedT}}{S_{RedS} * SaturationMapI(I_{RedS})} - 1 \right) * \frac{1}{S_{RedS}^3}. \end{aligned} \quad (16)$$

Etapla 2: Recorte de peso del croma permite que un creador de contenido modifique aún más la asignación de colores saturados a partir de la asignación automática para lograr mejor su intención creativa. Esto se calcula simplemente escalando el valor de *ChromaWeight* proporcionado en los metadatos (por ejemplo, consulte la ecuación (14)).

$$c_2 = ChromaWeight * 3. \quad (17)$$

Recorte de seis vectores

El bloque de recorte de seis vectores (655) permite que un creador de contenido modifique tanto el tono como la saturación de colores específicos para lograr mejor su intención creativa asignándolos a pantallas con un rango dinámico más bajo. Tal como se utiliza en este documento, el término "recortes" denota controles para que un creador de contenido realice modificaciones manuales adicionales a la curva de tonos para obtener la intención creativa deseada. Los metadatos relacionados con los recortes pueden incluir parámetros tales como: el valor de brillo máximo de una pantalla de destino, parámetros de pendiente-desplazamiento-potencia para la manipulación de la curva de tonos y pesos para ajustar los efectos de la ganancia de saturación y otros parámetros relacionados con el croma.

Las etapas, también representadas en la FIG. 6B, son las siguientes:

1. En el bloque 680, calcular el ángulo de tono de cada píxel.

$$H = \arctan\left(\frac{cp}{ct}\right). \quad (18)$$

donde ct y cp denotan los valores de croma modificados del procesamiento representado en la FIG. 6A.

2. En los bloques 685 y 690, interpolar en dos LUT pequeñas (3 bits) utilizando el ángulo de tono como índice

$$\begin{aligned} SMH &= SMHLUT(H) \\ r_1 &= HMHLUT(H) \\ r_2 &= 1 - r_1^2/2 \end{aligned} \quad (19)$$

Suponiendo que H está entre 0 y 360 grados, en una realización, el ajuste de tono (r_1) y el ajuste de saturación (SMH) se puede calcular de la siguiente manera:

$$\begin{aligned} kH &= \text{floor}(H / 3) \\ aH &= (H / 3) - kH \\ SMH &= SMHLUT[kH] * (1 - aH) + SMHLUT[kH + 1] * aH \end{aligned}$$

$$r_1 = \text{HMLUT}[k_H] * (1 - a_H) + \text{HMLUT}[k_H + 1] * a_H$$

donde, SMHLUT[j] y HMLUT[j], j=0, 1,..., 7, son tablas de búsqueda con valores de recorte proporcionados por el colorista como parte de los metadatos de entrada. Floor: Función parte entera).

3. Finalmente, las componentes de croma de salida Ct' y Cp' (657) de la salida con tonos mapeados (652) se calculan como:

$$\begin{aligned} Ct' &= SMH * (ct * r_2 - cp * r_1) \\ Cp' &= SMH * (ct * r_1 + cp * r_2) \end{aligned} \quad (20)$$

Las dos LUT se proporcionan mediante metadatos y consisten en la modificación deseada para cada uno de los seis tonos. El tono de cada píxel, calculado por el arcotangente de cuatro cuadrantes de los canales de croma, se utiliza para interpolar en esta tabla y encontrar la cantidad de modificación de tono y saturación que se debe aplicar. La LUT está pensada para uno o más niveles de luminancia específicos. Antes de la interpolación, la LUT se modifica en función del nivel de luminancia real de la pantalla de destino prevista, de la misma manera que los otros recortes.

IMPLEMENTACIÓN DE SISTEMA INFORMÁTICO DE EJEMPLO

Las realizaciones de la presente invención pueden implementarse con un sistema informático, sistemas configurados en circuitos y componentes electrónicos, un dispositivo de circuito integrado (Integrated Circuit, IC) tal como un microcontrolador, una matriz de puertas programables en campo (Field Programmable Gate Array, FPGA) u otro dispositivo lógico configurable o programable (Programmable Logic Device, PLD), un procesador de señal digital o de tiempo discreto (Digital Signal Processor, DSP), un IC específico de aplicación (Application Specific Integrated Circuit, ASIC) y/o un aparato que incluye uno o más de dichos sistemas, dispositivos o componentes. El ordenador y/o el IC pueden realizar, controlar o ejecutar instrucciones relacionadas con transformaciones de imágenes para imágenes con alto rango dinámico, tales como las descritas en el presente documento. El ordenador y/o el IC pueden calcular cualquiera de una variedad de parámetros o valores que están relacionados con la gestión de visualización de los procesos de vídeo HDR descritos en el presente documento. Las realizaciones de imagen y vídeo pueden implementarse en hardware, software, firmware y varias combinaciones de los mismos.

Ciertas implementaciones de la invención comprenden procesadores de ordenador que ejecutan instrucciones de software que hacen que los procesadores realicen un método de la invención. Por ejemplo, uno o más procesadores en una pantalla, un codificador, un decodificador, un transcodificador o similar pueden implementar métodos relacionados con la gestión de pantalla de vídeo HDR tal como se describió anteriormente, ejecutando instrucciones de software en una memoria de programa accesible a los procesadores. La invención también puede proporcionarse en forma de un producto de programa. El producto de programa puede comprender cualquier medio tangible y no transitorio que transporta un conjunto de señales legibles por ordenador que comprenden instrucciones que, cuando son ejecutadas por un procesador de datos, hacen que el procesador de datos ejecute un método de la invención. Los productos de programa según la invención pueden estar en cualquiera de una amplia variedad de formas tangibles. El producto de programa puede comprender, por ejemplo, medios físicos tales como medios de almacenamiento de datos magnéticos que incluyen disquetes, unidades de disco duro, medios de almacenamiento de datos ópticos que incluyen CD ROM, DVD, medios de almacenamiento de datos electrónicos que incluyen ROM, RAM flash o similares. Las señales legibles por ordenador en el producto de programa pueden estar comprimidas o encriptadas opcionalmente.

Cuando se ha hecho referencia a un componente (por ejemplo, un módulo de software, procesador, conjunto, dispositivo, circuito, etc.) anteriormente, a menos que se indique lo contrario, la referencia a ese componente (incluida una referencia a un "medio") debe interpretarse como que incluye como equivalentes de ese componente cualquier componente que realice la función del componente descrito (por ejemplo, que sea funcionalmente equivalente), incluidos los componentes que no sean estructuralmente equivalentes a la estructura divulgada que realiza la función en las realizaciones de ejemplo ilustradas de la invención.

EQUIVALENTES, EXTENSIONES, ALTERNATIVAS Y MISCELÁNEOS

Se describen, por tanto, realizaciones de ejemplo relacionadas con la gestión de visualización de vídeo HDR. En la memoria anterior, se han descrito realizaciones de la presente invención con referencia a numerosos detalles específicos que pueden variar de una implementación a otra. La invención se define solamente mediante las reivindicaciones adjuntas.

Cualquier definición expresamente establecida en el presente documento para los términos contenidos en dichas reivindicaciones registrará el significado de dichos términos tal como se utilizan en las reivindicaciones. Por tanto, ninguna limitación, elemento, propiedad, característica, ventaja o atributo que no se mencione expresamente en una reivindicación debe limitar en modo alguno el alcance de dicha reivindicación. Por consiguiente, la memoria y los dibujos deben considerarse en un sentido ilustrativo y no limitativo.

REIVINDICACIONES

1. Un método para gestión de visualización, comprendiendo el método:

obtener una imagen de entrada (507) en un primer rango dinámico y que tiene una primera resolución espacial, comprendiendo la primera imagen píxeles de intensidad (I) y píxeles de croma (C);

5 submuestrear (230) la imagen de entrada para generar una representación piramidal de la imagen de entrada que tiene imágenes de N capas en pirámide, donde N es al menos dos, en donde la imagen de la primera capa de la pirámide tiene una resolución espacial menor que la resolución espacial de la imagen de entrada, y la imagen de la capa k -ésima de la pirámide, para $k=2, \dots, N$, tiene una resolución espacial menor que la imagen de la capa $k-1$ de la pirámide;

10 formar una imagen de la capa de base filtrada mediante un filtrado que preserva los bordes (715) y sobremuestrear las imágenes de las N capas de la pirámide;

obtención de píxeles de intensidad intermedia I_B de una primera imagen basada en los píxeles de intensidad I de la imagen de entrada y en los píxeles de intensidad I_F de la imagen de la capa de base filtrada como

$$I_B = \alpha * I_F + (1 - \alpha) * I$$

15 donde α comprende un valor entre 0 y 1, I denota el valor del píxel de intensidad de la imagen de entrada e I_F denota el valor del píxel de intensidad correspondiente de la imagen de la capa de base filtrada;

aplicar una función de mapeo de tonos (615) a los píxeles de intensidad intermedia I_B de la primera imagen para generar píxeles de intensidad I' correspondientes de una imagen con tonos mapeados en un rango dinámico objetivo, en donde el rango dinámico objetivo es diferente del primer rango dinámico;

20 calcular una métrica de saturación para los píxeles de la primera imagen basándose en los valores de los píxeles de croma en la imagen de entrada;

calcular (660) un escalador de intensidad de mapeo de saturación (SMI), basándose en una función de los píxeles de intensidad correspondientes de la imagen de entrada y en los píxeles de intensidad de la imagen con tonos mapeados;

25 calcular (640) un escalador de saturación de mapeo de tonos (TMS) basándose en una primera función de la métrica de saturación de los píxeles de croma en la primera imagen;

calcular (645) un escalador de saturación de mapeo de saturación (SMS) basándose en una segunda función de la métrica de saturación de los píxeles de croma en la primera imagen;

aplicar el escalador de TMS a los píxeles de intensidad de la imagen con tonos mapeados para generar píxeles de intensidad de salida (I'');

30 multiplicar los escaladores de SMS y SMI con los píxeles de croma de la primera imagen para generar píxeles de croma intermedios;

calcular los píxeles de croma de salida basándose en los píxeles de croma intermedios; y

generar una imagen de salida (652) que comprende los píxeles de intensidad de salida y los píxeles de croma de salida.

35 2. El método de la reivindicación 1, en donde generar el escalador de TMS *ToneMapS* comprende el cálculo:

$$S = \sqrt{T^2 + P^2},$$

$$ToneMapS = (1 + S^3 * c_1) * (1 + S * c_2)$$

40 donde S denota la métrica de saturación para los valores de píxeles de croma T y P , c_1 es una variable calculada basándose en las diferencias entre el primer rango dinámico y el rango dinámico objetivo, y c_2 se calcula basándose en metadatos.

3. El método de la reivindicación 1 o la reivindicación 2, en donde el cálculo del escalador de SMI comprende el cálculo de una *SaturationScale*

$$SaturationScale = \frac{f_s(I')}{f_s(I)}$$

45 donde I denota un valor de píxel de intensidad en la primera imagen, I' denota un valor de intensidad correspondiente en la imagen con tonos mapeados, y la función $f_s(I)$ comprende una función de intensidad monótonamente creciente.

4. El método de la reivindicación 3, en donde

$$SaturationScale = \frac{(I'-1)^3+1}{(I-1)^3+1}$$

5. El método de cualquiera de las reivindicaciones 1 a 4, en donde generar el escalador de SMS *SaturationMapS* comprende el cálculo:

$$S = \sqrt{T^2 + P^2},$$

$$ToneMapS = (1 + S^3 * c_1) * (1 + S * c_2),$$

donde S denota la métrica de saturación para los valores de píxeles de croma T y P, c_1 es una variable calculada basándose en las diferencias entre el primer rango dinámico y el rango dinámico objetivo, y c_2 se calcula basándose en metadatos.

6. El método de cualquiera de las reivindicaciones 1 a 5, en donde la generación de los píxeles de croma intermedios comprende:

$$ct = T * SMI * SMS$$

$$cp = P * SMI * SMS$$

donde T y P denotan valores de los píxeles de croma de la primera imagen, y ct y cp denotan los píxeles de croma intermedios.

7. El método de la reivindicación 6, en donde la generación de los píxeles de croma de salida basándose en los píxeles de croma intermedios comprende:

recibir metadatos de recorte como parte de la primera imagen;

recibir una región de interés en la primera imagen;

para los píxeles de la primera imagen que pertenecen a la región de interés:

calcular un valor de tono de los píxeles de croma intermedios;

calcular un escalador de tono de mapa de saturación (SMH) basado en el valor de tono y en los metadatos de recorte;

calcular un valor de mapa de tonos (r1) basado en el valor del tono y en los metadatos de recorte; y

generar los píxeles de croma de salida basándose en el valor del mapa de tonos, en el escalador de SMH y en los píxeles de croma intermedios.

8. El método de la reivindicación 7, en donde la generación de los píxeles de croma de salida comprende calcular:

$$r2 = 1 - r1^2/2;$$

$$ct = ct * SMH;$$

$$cp = cp * SMH ;$$

$$Ct' = r2 * ct - r1 * cp;$$

$$Cp' = r1 * ct + r2 * cp,$$

donde r1 denota el valor del mapa de tonos, y Ct' y Cp' denotan los píxeles de croma de salida.

9. El método de la reivindicación 1, en donde aplicar una función de mapeo de tonos (615) a los píxeles de intensidad intermedia I_B de la primera imagen para generar píxeles de intensidad correspondientes I' de una imagen con tonos mapeados en un rango dinámico objetivo comprende:

aplicar un mapeo de tonos a los píxeles de intensidad intermedia I_B para obtener la intensidad de píxeles con tonos mapeados I'_B ; y

realizar:

$$I' = I'_B + (I - I_B) * \beta,$$

donde β es un escalador entre 0 y 1, I'_B denota la intensidad de píxeles de una versión con tonos mapeados de I_B .

10. El método de la reivindicación 1, en donde la generación de los valores de píxeles de intensidad de la imagen de la capa de base filtrada comprende:

5 generar un conjunto de imágenes de coeficientes de filtro de borde $Ima(1)$ e $Imb(1)$ (710) basándose en las imágenes de las N capas de la pirámide;

sobremuestreo del conjunto de imágenes de coeficientes de filtro de borde $Ima(1)$ e $Imb(1)$ para que coincida con la resolución espacial de la imagen de entrada y generar un conjunto de imágenes con coeficientes de filtro de borde sobremuestreados $ImaU(1)$ e $ImbU(1)$ (712); y

generar los valores de píxeles de intensidad de la capa de base filtrada como:

$$10 \quad I_F = ImaU(1) * I + ImbU(1).$$

11. El método de la reivindicación 10, en donde la generación de $Ima(1)$ e $Imb(1)$ comprende:

generar $Ima(N)$ e $Imb(N)$ utilizando un filtro de borde y $P(N)$, donde $P(N)$ denota la imagen de la capa N de la pirámide con la resolución espacial más baja;

15 generar $ImaU(N)$ e $ImbU(N)$ mediante la ampliación de $Ima(N)$ e $Imb(n)$ para que coincida con la resolución espacial de la imagen de la capa (N-1) de la pirámide;

para $i = N - 1$ a 2:

generar $S(i) = ImaU(i+1) * P(i) + ImbU(i+1)$

donde $P(i)$ denota la imagen de la capa i-ésima de la pirámide;

generar $Ima(i)$ e $Imb(i)$ utilizando el filtro de borde, $S(i)$ y $P(i)$; y

20 calcular $S(1) = ImaU(2) * P(1) + ImbU(2)$; y

generar $Ima(1)$ e $Imb(1)$ utilizando el filtro de borde, $S(1)$ y $P(1)$.

12. El método de la reivindicación 11, en donde la generación de $Ima(i)$ e $Imb(i)$ utilizando el filtro de borde, y $S(i)$ y $P(i)$ comprende:

generar imágenes $P2(i) = P^2(i)$ y $PS(i) = P(i) * S(i)$;

25 convolucionar $P2(i)$ con el filtro H para generar $P2out$;

convolucionar $P(i)$ con el filtro H para generar $Pout$;

convolucionar $S(i)$ con el filtro H para generar $Sout$;

convolucionar $PS(i)$ con el filtro H para generar $PSout$;

calcular:

$$30 \quad Pvar = P2out - Pout^2,$$

$$PScov = PSout - Sout * Pout$$

$$Ima(i) = PScov / (Pvar + PW(i)),$$

$$Imb(i) = Sout - Ima(i) * Pout$$

donde $PW(i)$ denota un conjunto de pesos recibidos a través de metadatos.

35 13. Un aparato que comprende un procesador y configurado para realizar cualquiera de los métodos mencionados en las reivindicaciones 1 a 12.

14. Un medio de almacenamiento no transitorio legible por ordenador que tiene almacenadas en él instrucciones ejecutables por ordenador para ejecutar un método con uno o más procesadores, según cualquiera de las reivindicaciones 1 a 12.

40

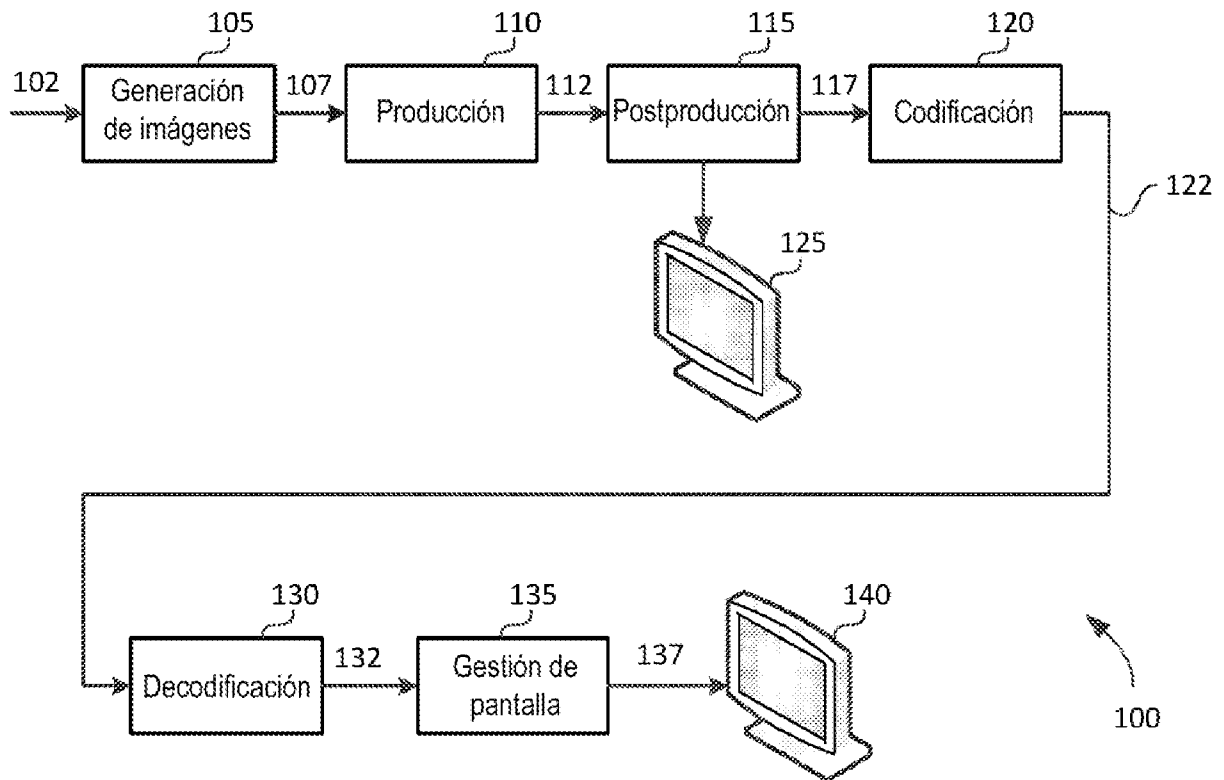


FIG. 1

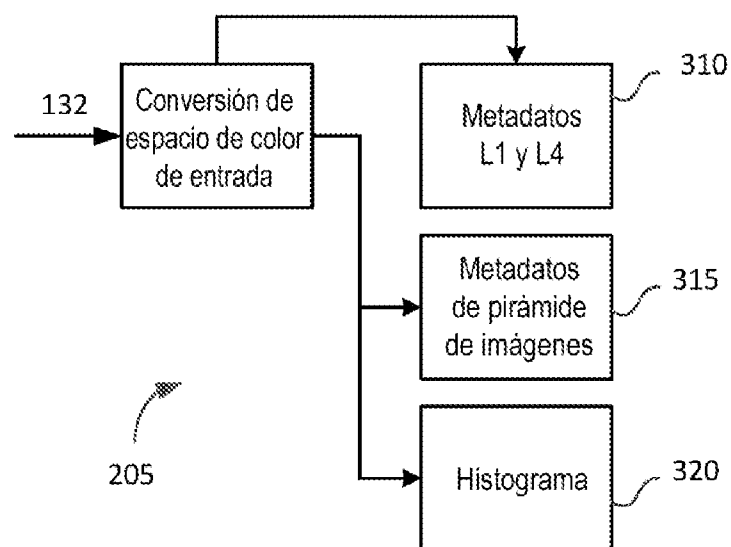


FIG. 3

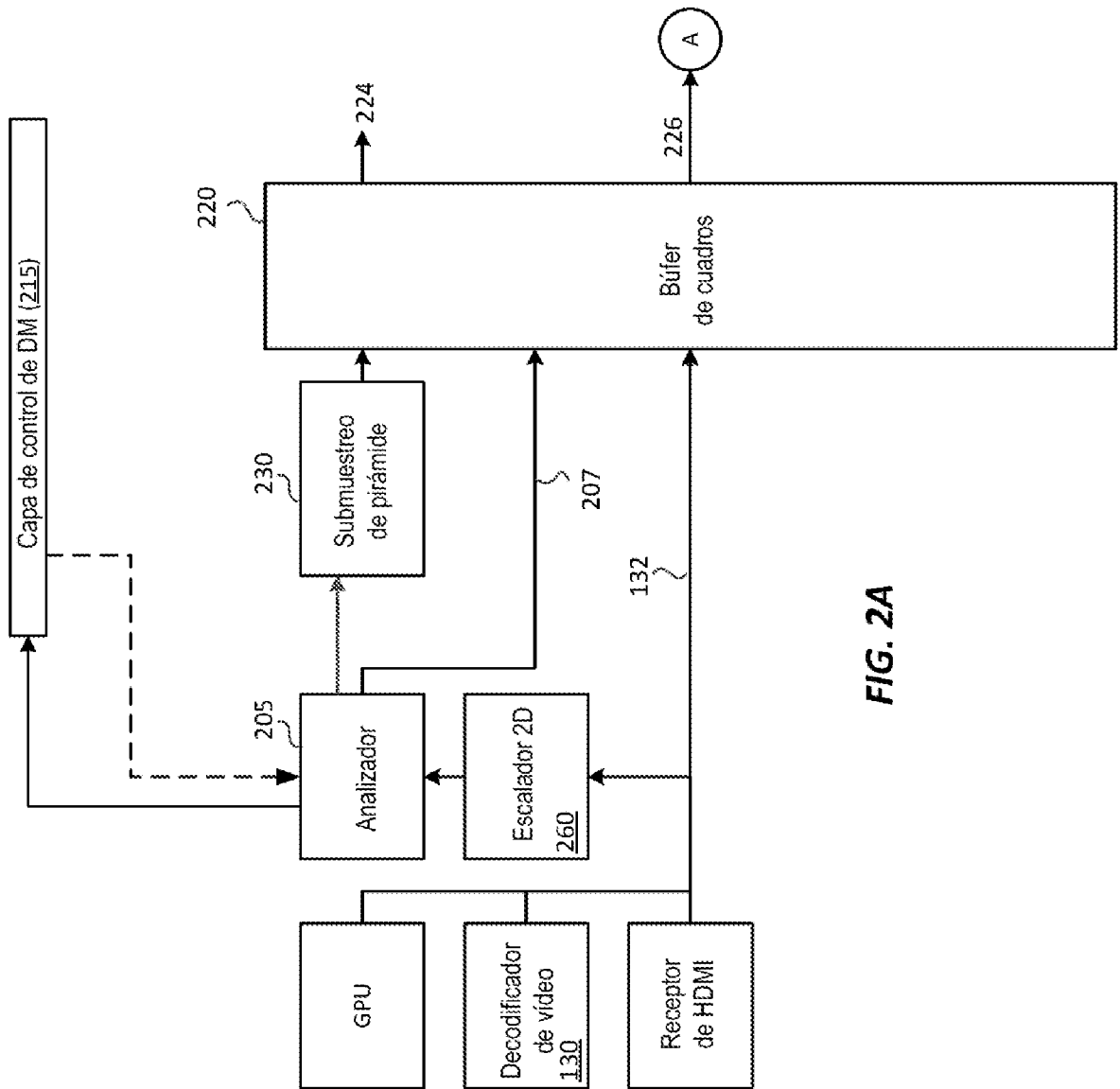


FIG. 2A

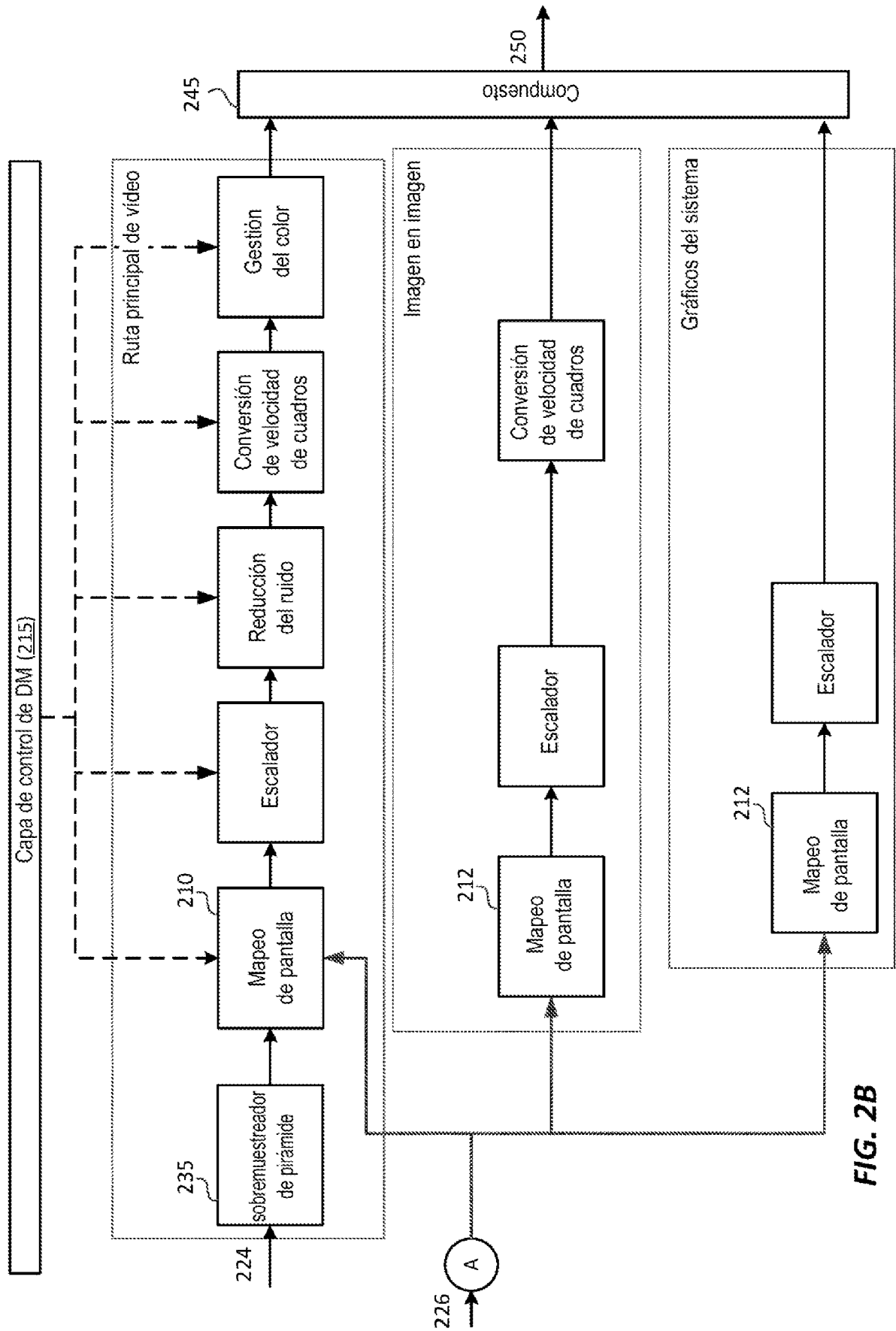
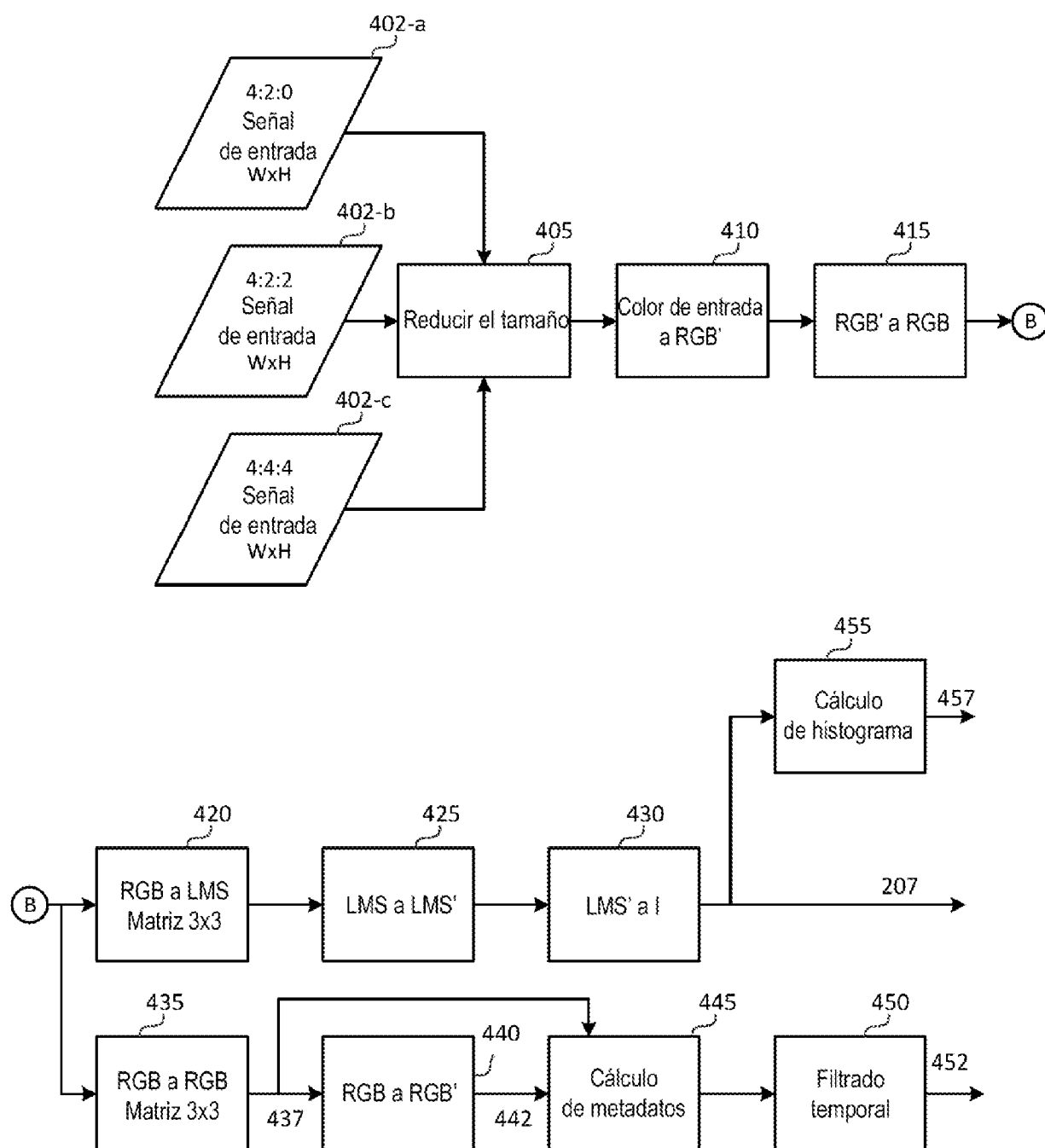


FIG. 2B

**FIG. 4**

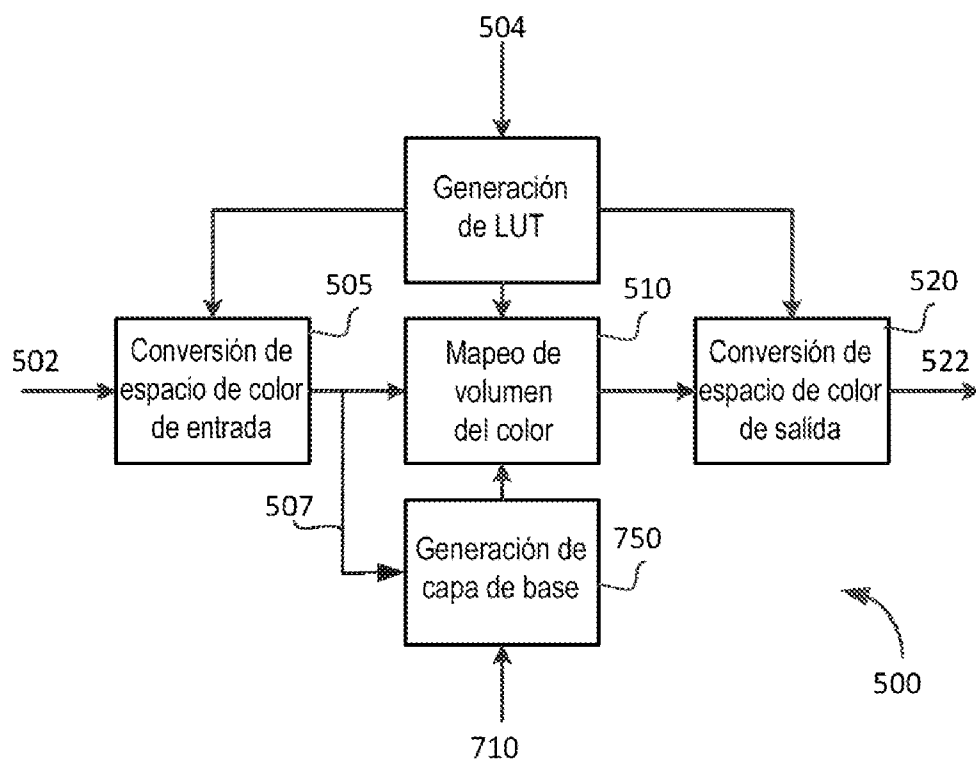


FIG. 5

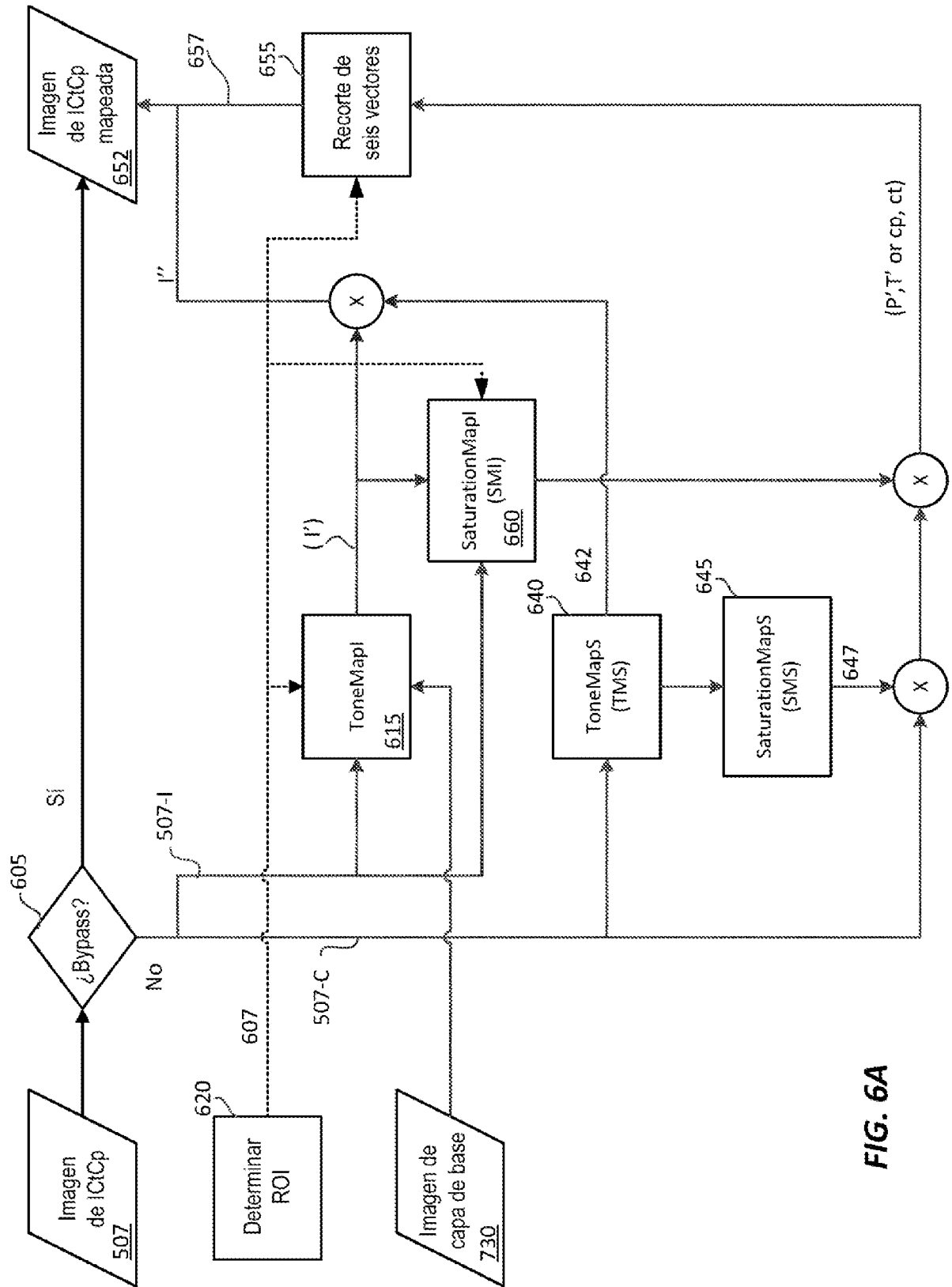


FIG. 6A

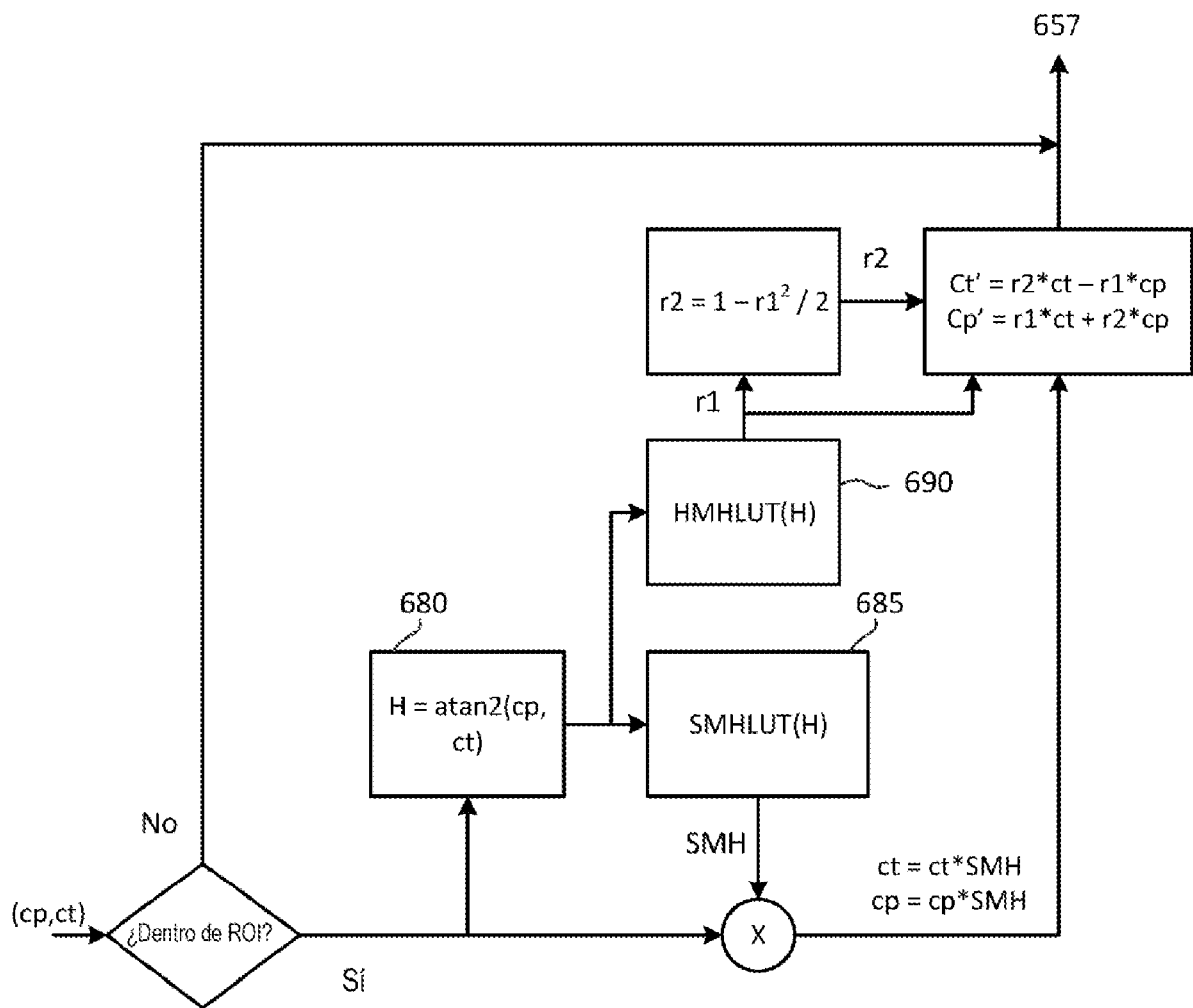


FIG. 6B

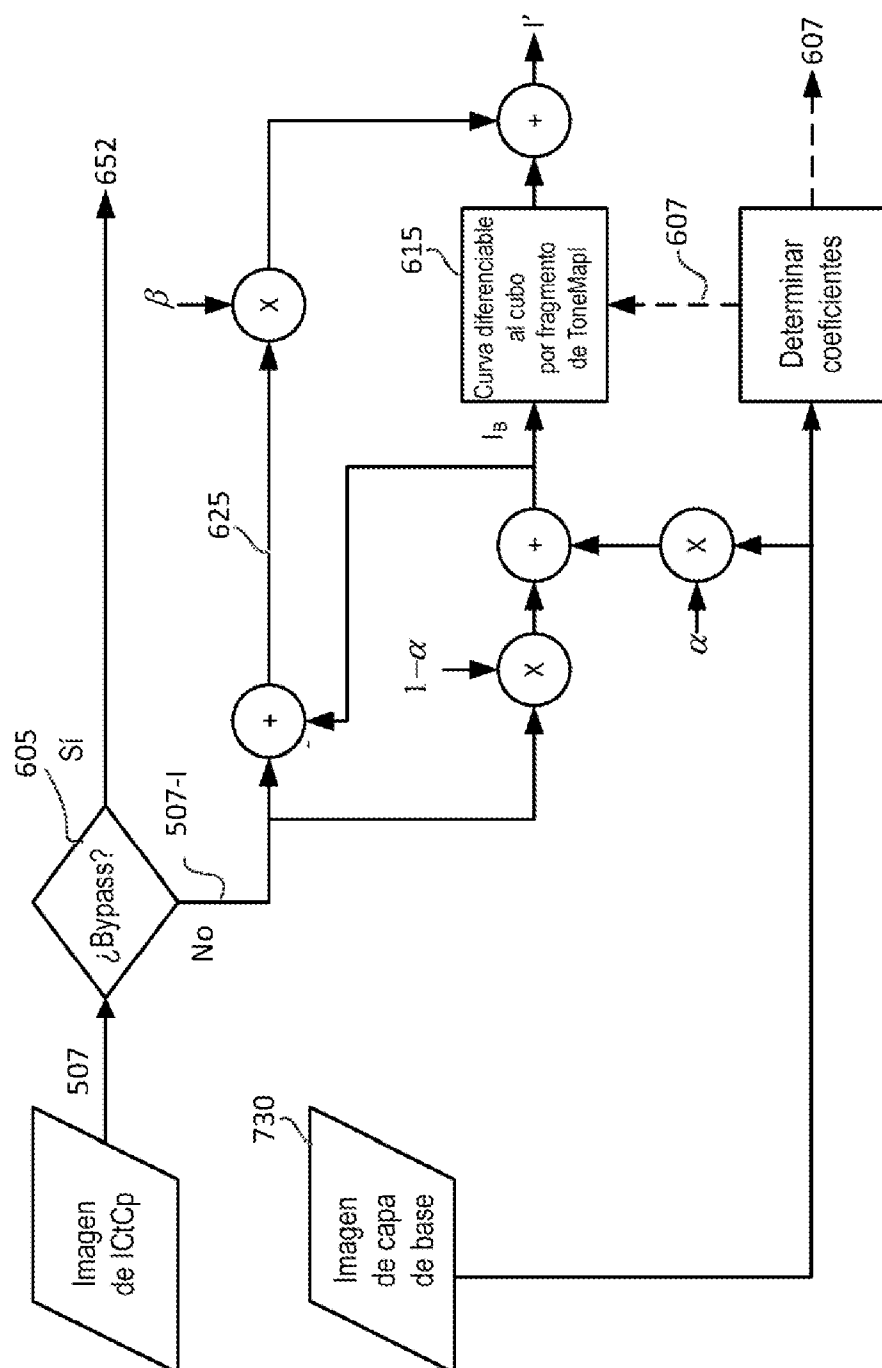


FIG. 6C

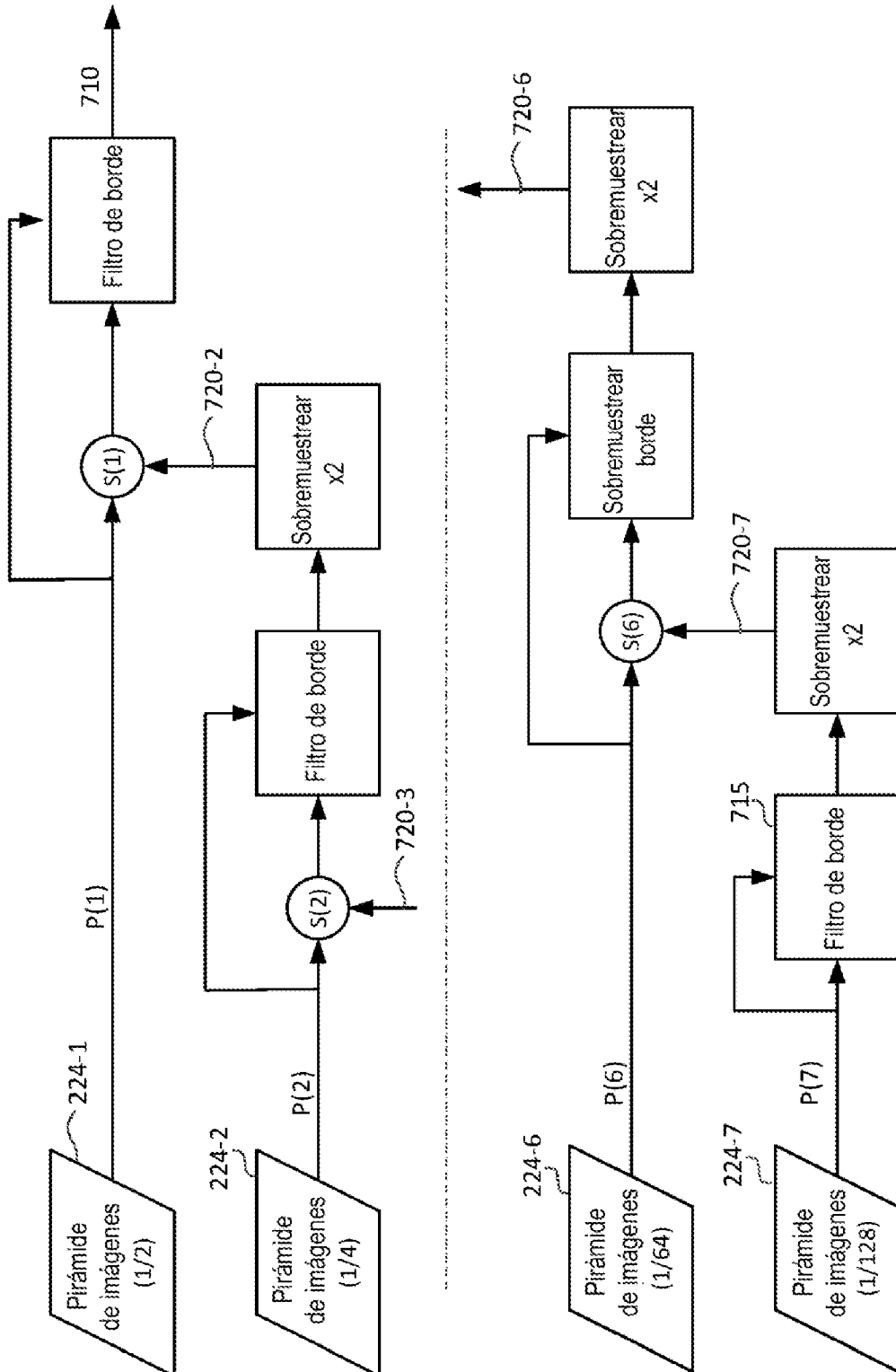


FIG. 7A

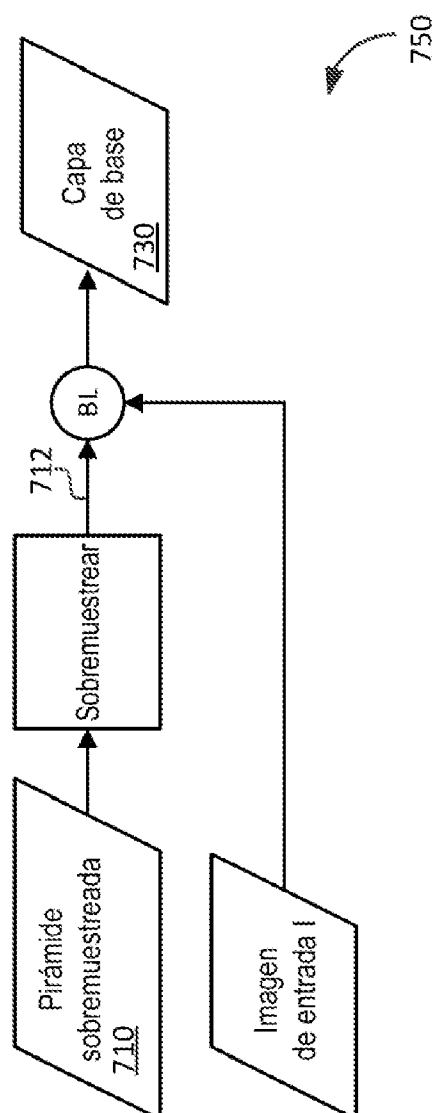


FIG. 7B

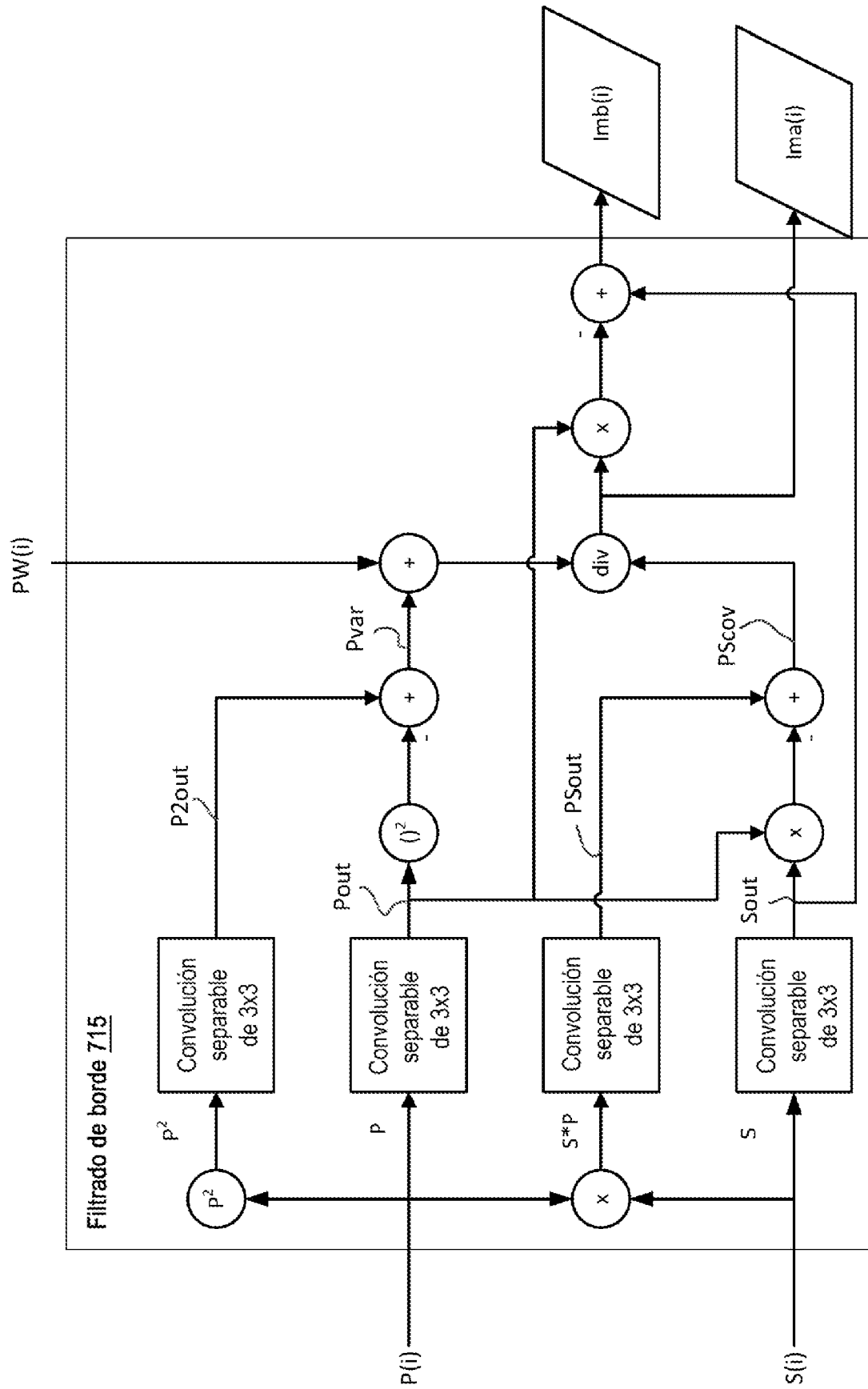


FIG. 7C